Dr.WEB
for UNIX Mail Servers

管理者マニュアル

Dr.Web for UNIX Mail Servers
バージョン11.1
管理者マニュアル
2024/02/01

Doctor Webロシア本社
2-12A, 3rd str. Yamskogo polya, Moscow, Russia, 125124
ウェブサイト: https://www.drweb.com/
電話番号：+7 (495) 789-45-87
支社および海外オフィスについては、Doctor Web公式サイトをご覧ください。

Doctor Web

Doctor Webは、悪意のあるソフトウェアやスパムからの効果的な保護を提供するDr.Web情報セキュリティソリューションの開発および販売を行っています。

世界中の個人ユーザーから政府機関、中小企業、大企業まで幅広いカスタマーに支持されています。

Dr.Webアンチウイルスソリューションは、マルウェア検出と国際情報セキュリティ基準への準拠における持続的な卓越性によって1992年よりその名を広く知られています。

Dr.Webソリューションに与えられた数々の認定や賞、そして世界中に広がるユーザーが、製品の持つ並外れた信頼性を示す何よりの証です。

Dr.Web製品をご利用いただき誠にありがとうございます。

目次

はじめに 9

表記規則と略語 10

この製品について 11

- **Dr.Web for UNIX Mail Serversのメイン機能** 11
- **Dr.Web for UNIX Mail Serversの構成** 15
- 隔離への移動 22
- ファイルのパーミッションと権限 23
- 動作モード 24

システム要件と互換性 27

ライセンス 32

インストールとアンインストール 33

- **Dr.Web for UNIX Mail Serversをインストールする** 34
  - ユニバーサルパッケージをインストールする 35
    - コマンドラインからインストールする 36
  - リポジトリからインストールする 37
- **Dr.Web for UNIX Mail Serversをアップグレードする** 40
  - パッケージとコンポーネントを更新する 40
  - 新しい製品バージョンにアップグレードする 41
  - インターネットに接続せずにデータベースを更新する 43
- **Dr.Web for UNIX Mail Serversをアンインストールする** 44
  - ユニバーサルパッケージをアンインストールする 44
    - コマンドラインからアンインストールする 45
  - リポジトリからインストールしたDr.Web for UNIX Mail Serversをアンインストールする 45
- **追加情報** 48
  - Dr.Web for UNIX Mail Serversパッケージとファイル 48
  - コンポーネントのカスタムインストールとアンインストール 51
- **セキュリティサブシステムを設定する** 57
  - SELinuxのセキュリティポリシーを設定する 58

開始する 61

- **製品の登録と有効化** 62
- **製品の動作確認** 65
- フィルターとしての**MTA**との統合 66
- **Dr.Web vxCube**との統合 74

SMTPプロキシモードでDr.Web for UNIX Mail Serversを使用する 78
透過プロキシモードでDr.Web for UNIX Mail Serversを使用する 79
簡単な説明 83
Dr.Web for UNIX Mail Serversコンポーネント 88
Dr.Web ConfigD 88
動作原理 88
コマンドライン引数 89
設定パラメータ 90
Dr.Web Ctl 94
コマンドライン呼び出しフォーマット 95
使用例 125
設定パラメータ 129
Dr.Web 管理Webインターフェース 130
コンポーネントを管理する 132
脅威の管理 132
設定を管理する 134
集中管理モードの管理 138
ローカルファイルをスキャンする 139
メールアーカイブのパスワードを復元する 142
Dr.Web MailD 143
動作原理 144
コマンドライン引数 145
設定パラメータ 146
メールシステムとの統合 158
Luaでのメール処理 158
Dr.Web Anti-Spam 216
動作原理 217
コマンドライン引数 218
設定パラメータ 219
Dr.Web Mail Quarantine 221
動作原理 221
コマンドライン引数 221
設定パラメータ 222
SpIDer Gate 224
動作原理 225
コマンドライン引数 226

設定パラメータ 226
Dr.Web Firewall for Linux 229
動作原理 229
コマンドライン引数 234
設定パラメータ 234
Luaでの接続処理 259
Dr.Web ClamD 267
動作原理 267
コマンドライン引数 267
設定パラメータ 268
外部アプリケーションとの統合 274
Dr.Web File Checker 276
動作原理 276
コマンドライン引数 277
設定パラメータ 277
Dr.Web Network Checker 280
動作原理 280
コマンドライン引数 282
設定パラメータ 283
スキャンクラスタの作成 287
Dr.Web Scanning Engine 291
動作原理 291
コマンドライン引数 292
設定パラメータ 294
Dr.Web Updater 296
動作原理 296
コマンドライン引数 297
設定パラメータ 298
Dr.Web ES Agent 304
動作原理 304
コマンドライン引数 304
設定パラメータ 305
Dr.Web HTTPD 308
動作原理 308
コマンドライン引数 309
設定パラメータ 309

HTTP APIの説明 312
Dr.Web SNMPD 335
動作原理 335
コマンドライン引数 337
設定パラメータ 337
SNMPモニタリングシステムとの統合 341
Dr.Web SNMP MIB 349
Dr.Web MeshD 376
動作原理 376
コマンドライン引数 379
設定パラメータ 380
Dr.Web URL Checker 383
動作原理 384
コマンドライン引数 384
設定パラメータ 385
Dr.Web CloudD 386
動作原理 386
コマンドライン引数 387
設定パラメータ 387
Dr.Web LookupD 389
動作原理 389
コマンドライン引数 390
設定パラメータ 391
Dr.Web StatD 403
動作原理 403
コマンドライン引数 403
設定パラメータ 404
付録 406
付録A. コンピューター脅威の種類 406
付録B. コンピューター脅威の駆除 410
付録C. テクニカルサポート 412
付録D. Dr.Web for UNIX Mail Servers設定ファイル 414
ファイル構造 414
パラメータタイプ 415
付録E. SSL証明書を生成する 419
付録F. 既知のエラー 422

付録G. 略語のリスト 474

はじめに

Dr.Web for UNIX Mail Serversをお買い上げいただきありがとうございます。本製品は最先端のウイルス検出および駆除テクノロジーを活用して、さまざまなタイプのコンピューター脅威の拡散からサーバーを確実に保護します。これにより、サーバーが提供するサービスの品質を向上させることができます。

このマニュアルの目的は、GNU/LinuxファミリーのOSやFreeBSDなど、UNIXライクなOSを実行しているサーバーの管理者が、Dr.Web for UNIX Mail Serversバージョン11.1をインストールしてご使用いただけるように支援することです。

ファイルパスの表記規則

ファイルとコンポーネントへの実際のパスは、OSによって異なります。本書では、ディレクトリに次の表記規則を使用しています。

- *<opt_dir>* - 主な製品ファイルがあるディレクトリ（実行ファイルとライブラリを含む）
- *<etc_dir>* - 設定ファイルとキーファイルがあるディレクトリ
- *<var_dir>* - サポートと製品の一時ファイルがあるディレクトリ

さまざまなOSの表記規則に対応する実際のパスは、以下の表に示されています。

OSの種類	表記規則	実際のパス
GNU/Linux	*<opt_dir>*	`/opt/drweb.com`
	<etc_dir>	`/etc/opt/drweb.com`
	<var_dir>	`/var/opt/drweb.com`
FreeBSD	*<opt_dir>*	`/usr/local/libexec/drweb.com`
	<etc_dir>	`/usr/local/etc/drweb.com`
	<var_dir>	`/var/drweb.com`

スペースを考慮して、例ではGNU/Linux OSのパスを使用しています。本書では、可能な場合においてすべてのOSの実際のパスが例に使用されています。

表記規則と略語

本マニュアルでは、以下の文字・記号を使用しています。

文字・記号	説明
	重要な事項や指示
	エラーの可能性や特に注意を必要とする重要な注意事項に関する警告
アンチウイルスネットワーク	新しい用語、または強調したい用語
<IP-address>	プレースホルダー
保存	ボタン、ウィンドウ、メニューアイテム、および他のプログラムインターフェースエレメントの名称
CTRL	キーボードのキーの名称
`/home/user`	ファイルやフォルダの名前、コード例
付録 A	マニュアル内の別の章への相互参照や外部Webページへのハイパーリンク

本マニュアルでは、(端末または端末エミュレーターで)キーボードから入力されるコマンドラインのコマンドには、コマンドプロンプト記号`$`または`#`が付いています。この記号は、該当するコマンドの実行に必要な権限を表しています(UNIX系システムの標準的表記規則に従って)。

`$` - コマンドはユーザー権限で実行できることを示します。

`#` - スーパーユーザー(通常は*root*)権限でコマンドを実行できることを示します。権限を昇格するには、`su`と`sudo`コマンドを使用してください。

略語のリストは、セクション付録G. 略語のリストにあります。

この製品について

このセクションの内容

- 機能
- Dr.Web for UNIX Mail Serversのメイン機能
- Dr.Web for UNIX Mail Serversの構成
- 隔離への移動
- ファイルのパーミッションと権限
- 動作モード

機能

Dr.Web for UNIX Mail Serversは、UNIX（GNU/Linux、FreeBSD）で実行されているメールサーバーやプロキシを、ウイルスや他のタイプの悪意のあるソフトウェアから保護し、さまざまなプラットフォーム向けに作成された脅威の拡散を防ぐためのものです。

主なコンポーネント（スキャンエンジンとウイルスデータベース）は、非常に効果的でリソースを節約するだけでなく、クロスプラットフォームでもあるため、Doctor Webスペシャリストはさまざまなプラットフォームをターゲットとした脅威から一般的なOSのコンピューターやモバイルデバイスを守る、信頼性の高いアンチウイルスソリューションを作成できます。現在、Doctor WebではDr.Web for UNIX Mail Serversとともに、UNIXベースのOS（GNU/Linux、FreeBSDなど）とIBM OS/2、Novell NetWare、macOSおよびWindowsの両方のアンチウイルスソリューションを提供しています。さらに、Android、Symbian、BlackBerry、Windows Mobileを実行するデバイスを保護するための、他のアンチウイルス製品が開発されています。

Dr.Web for UNIX Mail Serversのコンポーネントは常に更新され、ウイルスデータベース、Webリソースカテゴリーのデータベース、メールメッセージのスパムフィルタリングのルールデータベースには定期的に新しいシグネチャが追加されるため、サーバー、ワークステーション、モバイルユーザーとそのプログラムやデータに最新の保護を提供します。未知のウイルスに対する追加の保護を提供するため、スキャンエンジンとDr.Web Cloudサービスにはヒューリスティック解析が実装されており、シグネチャがデータベースにない最新の脅威に関する情報を保存します（この機能はDr.Webの一部の製品でのみ利用できます）。

Dr.Web for UNIX Mail Serversのメイン機能

1. **脅威の検出と駆除**。悪意のあるプログラム（メールファイルやブートレコードに感染するものを含むウイルス、トロイの木馬、メールワームなど）や望ましくないソフトウェア（アドウェア、ジョークプログラム、ダイアラーなど）を検索します。コンピューターの脅威の種類に関する詳細については、付録A. コンピューター脅威の種類を参照してください。

   脅威の検出方法：

   - *シグネチャ解析*。既知の脅威を検出できます。
   - *ヒューリスティック解析*。ウイルスデータベースに含まれていない脅威を検出できます。
   - *クラウドベースの脅威検出テクノロジー*。Dr.Web Cloudサービスを使用して、新しい脅威に関する最新の情報を収集し、それをDr.Web製品に送信します。

ヒューリスティックアナライザは誤検知を引き起こす可能性があります。その結果、アナライザによって検出された脅威を含むオブジェクトは「疑わしい」と見なされます。このようなファイルを隔離し、解析のためにDoctor Webアンチウイルスラボに送信することをお勧めします。脅威を駆除する方法の詳細は、付録B. コンピューター脅威の駆除を参照してください。

ユーザーのリクエストに応じてファイルシステムをスキャンする場合、ユーザーが利用できるすべてのファイルシステムオブジェクトのフルスキャン、または指定されたオブジェクトのみ（指定された基準を満たす個別のディレクトリまたはファイル）のカスタムスキャンが可能です。さらに、システム内で現在アクティブなプロセスをサポートするボリュームと実行ファイルのブートレコードを別々にチェックすることもできます。後者の場合、脅威が検出されると、悪意のある実行ファイルを駆除するだけでなく、選択的スキャンにより実行されているすべてのプロセスが強制的に終了されます。一連の異なるアクセスレベルを持つファイルへのアクセスの必須モデルを実装するシステムでは、現在のアクセスレベルでは利用できないファイルのスキャンは特別な自律コピーモードで行うことができます。

ファイルシステムで検出された脅威を含むオブジェクトはすべて、自律コピーモードで検出された脅威を除いて、永久保存された脅威レジストリに登録されます。

Dr.Web for UNIX Mail ServersにはDr.Web Ctlコマンドラインを使うと、SSHまたはTelnet経由でのリモート端末アクセスを提供するリモートネットワークホストの脅威ファイルシステムをスキャンできます。

リモートスキャンは、リモートホスト上の悪意のあるファイルや疑わしいファイルの検出にのみ使用できます。リモートホストで検出された脅威を排除するには、このホストによって直接提供される管理ツールを使用する必要があります。たとえば、ルーターやその他の「スマート」デバイスの場合は、ファームウェア更新のメカニズムを使用できます。コンピューティングマシンの場合は、コンピューティングマシンへの接続（任意でリモート端末モードを使用）とファイルシステムのそれぞれの操作（ファイルの削除または移動など）、またはコンピューティングマシンにインストールされているアンチウイルスソフトウェアの実行を介して実行できます。

2. **メールメッセージのスキャン**。Dr.Web for UNIX Mail Serversはさまざまなメールメッセージのスキャンモードをサポートしています。
   - *メールサーバー（MTA）に接続された外部フィルターモード*。Dr.Web for UNIX Mail Serversは、外部フィルター（*Milter*、*Spamd*、*Rspamd*）との接続用のインターフェースをサポートするメールサーバーに統合できます。フィルターモードでは、MTAが主導して、メールサーバーに届いたすべてのメールが有効化されたインターフェースを経由してDr.Web for UNIX Mail Serversに送信され、スキャンされます。インターフェースの機能に応じて、フィルターとして動作するDr.Web for UNIX Mail Serversは次のことが可能です。
     - *メールスキャンの結果をサーバーに通知します*。この場合、メールサーバーは受信した結果に従ってメールメッセージを*個別に*処理する必要があります（スキャン結果に脅威の存在に関する情報が含まれている場合は、配信を拒否する、ヘッダーを追加する、またはメールの内容を変更します）。
     - *メッセージを受信または拒否するコマンドをメールサーバーに送信します*。
     - ヘッダーを追加するか、検出された悪意のあるコンテンツや望ましくないコンテンツを削除して、*メールメッセージを変更します*。削除された悪意のあるコンテンツは、パスワードで保護されたアーカイブとしてメールメッセージに添付されます。メールメッセージの受信者は、保護されたアーカイブを解凍するためのパスワードをメールサーバー管理者に要求できます。推奨されませんが、管理者は必要に応じて、パスワードで保護されていないアーカイブの使用を設定できます。

メールサーバーへのコマンドの送信または変更されたメッセージの返信は、*Milter*インターフェースでのみサポートされます。*Spamd*と*Rspamd*では、Dr.Web for UNIX Mail Serversがサーバーにコマンドを送信し、変更されたメールメッセージを返すことはできません。「*メールメッセージはスパムです*」または「*メールメッセージはスパムではありません*」という2つの判定のうち1つがサーバーに返されます。このようなメッセージの処理に関するすべてのアクション(メールヘッダーの追加や変更、メッセージの拒否、受信者への送信など)は、*MTA側の設定*で定義する必要があります。

メールメッセージが拒否された理由と、場合によってはMTAがメールメッセージに適用する必要があるアクションをMTAに返すために、テキスト変数(`Spamd`には*report*、`Rspamd`には*action*)が使用されます。これはMTAで処理できる、LUAメッセージ処理手順によって返されます(たとえば、EximではACL)。

- *SMTPプロキシモード*は、1つまたは複数のMTAやMDAにさらに転送したSMTPトラフィックを迂回してスキャンします。実際には、このモードは前のモードと似ています。Dr.Web for UNIX Mail Serversが(*Milter*、*Spamd*、または*Rspamd*を使用して)MTA(たとえばPostfix)に接続され、このMTAは他のMTAにメールメッセージを送信するようにカスタマイズされています(たとえば、さまざまなドメイン宛てのメッセージルーティングの実行など)。
- *メールプロトコルの透過プロキシモード*。このモードでは、Dr.Web for UNIX Mail Serversは(SpIDer Gateコンポーネントを使用して)共有相手に対して透過的にMTAやMUAの間でデータを共有するためにチャネルに埋め込まれているプロキシサーバーの機能と、送信済みメッセージのスキャナの機能を実行します。この製品は主なメールプロトコル(SMTP、POP3、IMAP)に透過的に組み込むことができます。このモードでは、組み込まれているプロトコルにもよりますが、Dr.Web for UNIX Mail Serversは受信者にメールメッセージを送信することや(メールメッセージの変更や、変更後のヘッダーの追加、メールメッセージの再圧縮はできません)、その配信をブロックすることができます。これには、送信者または受信者への適切なプロトコルエラーの返信も含まれます。

透過プロキシモードはGNU/Linuxでのみ使用できます。

Dr.Web for UNIX Mail Serversは単体で動作可能なメールサーバーではないため、プロキシモードで動作させるには、Dr.Web for UNIX Mail Serversが動作する同じホストにメールサーバー(MTA)をインストールする必要があります。

Dr.Web for UNIX Mail Serversはディストリビューションと設定に応じて、メールメッセージのスキャンを実行します。

- 脅威を含む*悪意のある添付ファイルの検出*
- *悪意のあるWebサイトまたは望ましくないカテゴリーに属するWebサイトへのリンクの検索*
- *フィッシングとスパムの兆候の検出*(DKIMテクノロジー、スパムフィルタリングの自動的に更新されたルールデータベース、DNSxLブラックリスト内の送信者のアドレスの存在をチェックするメカニズムを使用)
- メールシステムの*管理者が独自に設定したセキュリティ基準への準拠*(正規表現を使用したメッセージ本文とヘッダーのスキャン)

メールメッセージに含まれている望ましくないWebサイトへのリンクのスキャンには、自動的に更新されるWebリソースカテゴリーのデータベースが使用されます。これはDr.Web for UNIX Mail Serversとともに配信されます。また、メールメッセージに記載されているWebソースが他のDr.Web製品によって悪意のあるものとしてマークされている場合、Dr.Web Cloudは情報の入手可能性を確認するように要求されます。

バージョン11.0以降のDr.Web for UNIX Mail Serversでは、メールメッセージに適用可能なアクションが*大幅に削減されています*。

バージョン11.0以降、Dr.Web for UNIX Mail Serversはメールメッセージに対して次のアクションのみを実行します。

- 管理者が設定した基準に準拠していることと、スパムの兆候をスキャンしていることを確認するための*メールメッセージチェック*（DNSxLブラックリストをチェックする設定がされている場合、そのリストの送信者ドメインのチェックも実施）
- 悪意のあるWebサイトまたは望ましくないカテゴリーに属するWebサイトへの*リンクの検索*
- *悪意のある添付ファイルの検出*

スキャン用のメールメッセージの受信に使用されたプロトコルとメールメッセージを送信した側（MTA/MDAまたはMUA）がスキャン用の転送済みメールメッセージの変更をサポートしている場合、Dr.Web for UNIX Mail Serversは標準的なアクション「無視」と「拒否」に加え、事前に定義された再圧縮テンプレートの1つに基づいてメールメッセージを*再圧縮*できます（再圧縮中は、すべての脅威がメールに添付された保護アーカイブに移動され、脅威や望ましくない内容に関する通知がメールの本文に追加されます）。その上、メールのヘッダーを追加、修正する基本機能がサポートされています。

*その他*のすべてのアクション（管理者への通知の送信、添付ファイルの完全な削除、名前変更など）が必要な場合は、保護されたメールサーバー（MTA/MDA）を*介して実装する必要があります。必要に応じて、他社の開発者から対応する処理用に設計された一連の特定のフィルタープラグインを入手し、接続することにより、保護されたメールサーバー経由でそれらを実装する必要があります。*

ディストリビューションによっては、Dr.Web for UNIX Mail ServersにDr.Web Anti-Spamコンポーネントが含まれない場合があります。この場合、スパムに対するメッセージのスキャンは行われません。

3. **感染したオブジェクトや疑わしいオブジェクトを確実に隔離します**。サーバーのファイルシステムで検出されたそのようなオブジェクトは、システムへの害を防ぐ特別なフォルダに移動され、隔離されます。隔離へ移動されたオブジェクトは、特別なルールに従って名前が変更され、必要に応じて、要求があった場合にのみ元の場所に復元できます。

   Dr.Web MailDコンポーネントによって検出されたメールメッセージ内の脅威は、サーバー上の隔離に移動され、変更されたメールメッセージ内でユーザー受信者に送信されます。その際、パスワードで保護されたアーカイブに圧縮されます。ユーザーは、Dr.Web for UNIX Mail Serversの管理者から受け取ったパスワードを指示するだけで、アーカイブの内容にアクセスできます。
4. マルウェアに対する高度な保護を維持するための、スキャンエンジン、ウイルスデータベース、Webリソースカテゴリーのデータベース、メールスパムフィルタリングルールのデータベースの**自動更新**。
5. ウイルスイベントに関する**統計の収集**、脅威検出イベントのロギング。SNMPを介して検出された脅威に関する通知を、外部のモニタリングシステム、集中管理サーバー（Dr.Web for UNIX Mail Serversが集中管理モードで動作している場合）、およびDr.Web Cloudに送信します。
6. **集中管理モードでの動作**（Dr.Web Enterprise Serverなどの集中管理サーバーに接続している場合、またはDr.Web AV-Deskサービスの一部として）。このモードでは、保護されたネットワーク内のコンピューターに、統合されたセキュリティポリシーを導入できます。保護されたネットワークには、企業ネットワークやプライベートネットワーク（VPN）、サービスプロバイダー（インターネットサービスプロバイダーなど）のネットワークが該当します。

Dr.Web for UNIX Mail Serversの構成

Dr.Web for UNIX Mail Serversは一連のコンポーネントで構成される製品であり、各コンポーネントにはそれぞれ独自の機能のセットがあります。コンポーネントはその目的に応じて次のカテゴリーに分けられます。

- 基本アンチウイルスコンポーネント：Dr.Web for UNIX Mail Serversの中核をなすコンポーネント。このカテゴリーにコンポーネントがない場合、本製品はファイル（およびその他のデータ）をスキャンできず、ウイルスやその他の脅威を検出できません。
- 脅威検索コンポーネント：これらのコンポーネントは、Dr.Web for UNIX Mail Serversの基本的なタスク（脅威や潜在的に危険なオブジェクトの検出）で使用されます。このカテゴリーに属するコンポーネントはその動作において、基本アンチウイルスコンポーネントを使用します。
- サービスコンポーネント：アンチウイルス保護に関する補助的な機能（アンチウイルスデータベースの更新、集中管理サーバーの接続、一般的なDr.Web for UNIX Mail Serversの動作管理など）を提供します。
- インターフェースコンポーネント：Dr.Web for UNIX Mail Serversのインターフェースを（ユーザーまたは他社製アプリケーションに）提供します。

以下は、Dr.Web for UNIX Mail Serversコンポーネントのリストです。

1. 基本アンチウイルスコンポーネント

コンポーネント	説明
Dr.Web Virus-Finding Engine	アンチウイルスエンジン。ウイルスと悪意のあるプログラムを検出するアルゴリズムを実装します（シグネチャ解析とヒューリスティック解析を使用）。 Dr.Web Scanning Engineによって管理されます。 ライブラリファイル：`drweb32.dll` ログファイルに表示される内部名：`CoreEngine`
Dr.Web Scanning Engine	スキャンエンジン。このコンポーネントはDr.Web Virus-Finding Engineとアンチウイルスデータベースを読み込みます。 • スキャンのためにファイルの内容とブートレコードをアンチウイルスエンジンに送信します。 • スキャン対象のファイルのキューを管理します。 • このアクションが適用可能な脅威を修復します。 Dr.Web ConfigDによって動作するか、自律的に動作できます。 Dr.Web File CheckerおよびDr.Web Network Checkerコンポーネントで使用されます。また、Dr.Web MeshDコンポーネント（一部の動作モード）や、Dr.Web Scanning Engine APIを特別に使用する、Dr.Web for UNIX Mail Serversに対して外部のアプリケーションによって使用されることもあります。 実行ファイル：`drweb-se` ログに表示される内部名：`ScanEngine`
ウイルスデータベース	ウイルスやその他の脅威のシグネチャ、および悪意のあるソフトウェアの検出・駆除アルゴリズムの自動的に更新されるデータベース。 Dr.Web Virus-Finding Engineによって使用され、一緒に提供されています。

コンポーネント	説明
Webリソースカテゴリーのデータベース	分類されたWebリソースのリストを含み、望ましくないWebサイトを識別するために使用される、自動的に更新されるデータベース。 SpIDer Gate、Dr.Web MailDなど、ユーザーとアプリケーションのネットワークアクティビティをスキャンするコンポーネントによって使用されます。
Dr.Web File Checker	ファイルシステムのオブジェクトをスキャンするコンポーネントと隔離マネージャー。 • Dr.Web Scanning Engineに対してローカルのファイルシステムにあるファイルをスキャンするときに、脅威スキャンコンポーネントからタスクを受け取ります。 • タスクに従ってファイルシステムのディレクトリを検索し、スキャンするファイルをDr.Web Scanning Engineに送信し、スキャンの進行状況をクライアントコンポーネントに通知します。 • 感染したファイルの削除、隔離への移動と隔離からの復元、隔離ディレクトリの管理を行います。 • キャッシュを構築し、最新の状態に保ちます。キャッシュには、以前にスキャンされたファイルに関する情報が含まれており、ファイルを再スキャンする周期を減らします。 ファイルシステムのオブジェクトをスキャンするコンポーネントによって使用されます。 実行ファイル：`drweb-filecheck` ログに表示される内部名：`FileCheck`
Dr.Web Network Checker	ネットワークデータスキャンエージェント。 • スキャンエンジンへのデータの送信に使用されます。データはネットワーク経由で製品のコンポーネント（Dr.Web ClamD、SpIDer Gate、Dr.Web MailDなど）によって送信されます。 • このコンポーネントによってDr.Web for UNIX Mail Serversは分散ファイルのスキャンを管理できるようになり、リモートホストとの間でスキャン用のファイルを送受信します。そのためには、リモートホストにはUNIX OS用にインストールされ実行されているDr.Webを備えている必要があります。それによって分散スキャンモードでは、多数のスキャンタスクのあるホスト（メールサーバー、ファイルサーバー、インターネットゲートウェイなど）の負荷を減らすことで、リモートホスト間でスキャン負荷を自動的に分散できます。 スキャン用のデータを受信できるパートナーホストがネットワーク上に存在する場合、スキャンにDr.Web Network Checkerを使用するコンポーネントはローカルのDr.Web Scanning Engineを使用しない場合があります。したがって、Dr.Web Scanning Engine、Dr.Web Virus-Finding Engine、およびアンチウイルスデータベースが存在しない可能性があります。 セキュリティ上の理由から、ファイルはネットワーク経由でSSLを使用して転送されます。 実行ファイル：`drweb-netcheck` ログに表示される内部名：`NetCheck`
Dr.Web URL Checker	URLが潜在的に危険なカテゴリーや望ましくないカテゴリーに該当するかどうかを解析するためのコンポーネント

コンポーネント	説明
	実行ファイル：drweb-urlcheck ログに表示される内部名：UrlCheck
Dr.Web MeshD	Dr.Web for UNIX Mail Serversをローカルクラウドに接続するコンポーネント。Dr.Web for UNIX製品が更新およびファイルスキャンの結果を交換し、スキャンのためにファイルを相互に送信し、スキャンエンジンサービスを直接提供できるようにします。 製品にこのコンポーネントが含まれている場合、このコンポーネントが接続されているローカルクラウドや、スキャンエンジンサービスを提供するホスト、Dr.Web Scanning Engine、Dr.Web Virus-Finding Engine、アンチウイルスデータベースが存在しない場合があります。 実行ファイル：drweb-meshd ログに表示される内部名：MeshD

2. 脅威検索コンポーネント

コンポーネント	説明
SpIDer Gate	ネットワークトラフィックとURLをモニタリングするためのコンポーネント。 ネットワークからローカルホストにダウンロードされ、そこから外部ネットワークに送信されたデータの脅威をスキャンするように設計されています。またこのコンポーネントは、Webリソースの望ましくないカテゴリーや、システム管理者が作成したブラックリストに追加されたネットワークホストとの接続を阻止します。 メールプロトコル（SMTP、POP3、IMAP）の透過プロキシとして、Dr.Web MailDによって使用されます。 Dr.Web Network Checkerを使用して、受信したデータをスキャンします。 GNU/Linux OS用ディストリビューションにのみ含まれています。 実行ファイル：drweb-gated ログに表示される内部名：GateD
Dr.Web Firewall for Linux	ネットワーク接続モニター。 SpIDer Gateによって使用され、送信されたネットワークトラフィックをスキャンするためにサーバー上で動作するアプリケーションに接続ルーティングを提供します。 GNU/Linux OS用ディストリビューションにのみ含まれています。 実行ファイル：drweb-firewall

コンポーネント	説明
	ログファイルに表示される内部名：LinuxFirewall
Dr.Web MailD	メールメッセージをスキャンするためのコンポーネント。 メールメッセージを解析し、それらに脅威がないかスキャンする準備をします。次の2つのモードで動作できます。 1) *Milter*、*Spamd*または*Rspamd*インターフェースを介して接続されたメールサーバー（Sendmail、Postfixなど）のフィルター。 2) メールプロトコル（SMTP、POP3、IMAP）の透過プロキシ。このモードではSpIDer Gateを使用します。 Dr.Web Network Checkerを使用してメールメッセージ内のデータをスキャンします。 実行コンポーネントファイル：drweb-maild ログに表示される内部名：MailD
Dr.Web Anti-Spam	スパムの特徴の有無を調べるため、メールをスキャンするコンポーネント。 Dr.Web MailDによって使用されます。ディストリビューションパッケージによっては利用できない場合があります。利用できない場合、Dr.Web MailDはスパムの特徴の有無についてメールメッセージをスキャンしません。 このコンポーネントはARM64およびE2Kのアーキテクチャではサポートされていません。 実行ファイル：drweb-ase ログに表示される内部名：Antispam

3. サービスコンポーネント

コンポーネント	説明
Dr.Web CloudD	Dr.Web Cloudとのインタラクションのためのコンポーネント。 ユーザーが閲覧したURLと、スキャンしたファイルに関する情報をDr.Web Cloudサービスに送信して、ウイルスデータベースにまだ含まれていない脅威がないかスキャンします。 実行ファイル：drweb-cloudd ログに表示される内部名：CloudD
Dr.Web ConfigD	Dr.Web for UNIX Mail Servers設定デーモン。 • 設定に応じて、他の製品のコンポーネントを起動／停止します。 • コンポーネントの動作に障害が発生した場合、該当するコンポーネントを再起動します。他のコンポーネントのリクエストに応じてコンポーネントを起動します。別のコンポーネントが起動またはシャットダウンしたときに、アクティブなコンポーネントに通知します。 • 現在のライセンスキーと設定に関する情報を保存し、その情報をすべてのコンポーネントに提供します。それらの情報を提供すると想定され

コンポーネント	説明
	るDr.Web for UNIX Mail Serversのコンポーネントから、調整済みの設定とライセンスキーを受け取ります。ライセンスキーと設定の変更について他のコンポーネントに通知します。 実行ファイル：`drweb-configd` ログファイルに表示される内部名：`ConfigD`
Dr.Web ES Agent	集中管理エージェント。集中管理モードとモバイルモードでの製品の動作を確実なものにします。 • 製品と集中管理サーバー間の接続を提供し、ライセンスキーファイル、ウイルスデータベースとアンチウイルスエンジンの更新を受信します。 • Dr.Web for UNIX Mail Serversに含まれるコンポーネントとそのステータス、ウイルスイベントの統計に関する情報をサーバーに送信します。 実行ファイル：`drweb-esagent` ログに表示される内部名：`ESAgent`
Dr.Web LookupD	外部データソースからデータを取得するためのコンポーネント。 外部データソース（ディレクトリサービス、ファイル、関連するデータベースなど）から、トラフィックモニタリングのルールで使用されるデータを取得します。 実行ファイル：`drweb-lookupd` ログに表示される内部名：`LookupD`
Dr.Web Mail Quarantine	スキャン対象のメッセージのキューを管理するメールメッセージスキャンコンポーネント。 Dr.Web MailDによって使用されます。ディストリビューションパッケージによっては利用できない場合があります。利用できない場合、Dr.Web MailDのSMTPモードとBCCモードはサポートされません。 実行ファイル：`drweb-mail-quarantine` ログに表示される内部名：`MailQuarantine`
Dr.Web StatD	Dr.Web for UNIX Mail Serversのコンポーネントの動作イベントを保存するためのコンポーネント。 製品コンポーネントのイベント（異常終了や脅威検出など）を受信して保存します。 実行ファイル：`drweb-statd` ログファイルに表示される内部名：`StatD`
Dr.Web Updater	更新コンポーネント。 ウイルスデータベース、Webリソースカテゴリーのデータベース、アンチウイルスエンジン、スパムの特徴の有無についてメールメッセージをスキャンするためのライブラリの更新をDoctor Webのサーバーからダウンロードします。 更新はスケジュールに従い、またユーザーの要求に応じて（Dr.Web Ctlまたは管理Webインターフェース経由で）自動的にダウンロードできます。

コンポーネント	説明
	実行ファイル：drweb-update ログに表示される内部名：Update

4. インターフェースコンポーネント

コンポーネント	説明
Dr.Web HTTPD	Dr.Web for UNIX Mail Servers管理Webサーバー。 Dr.Web for UNIX Mail Serversコンポーネントを管理するためのカスタムHTTP APIを提供します。 指定されたAPIは管理Webインターフェース（別途インストールが必要）によって使用されます。 セキュリティ上の理由から、このコンポーネントはWebインターフェースへの接続にHTTPSを使用します。 Dr.Web Network Checkerを使用して、スキャン用のデータをDr.Web Scanning Engineに送信します。 実行ファイル：drweb-httpd ログに表示される内部名：HTTPD
Dr.Web 管理Webインターフェース	管理Webインターフェース。 このインターフェースには、ローカルホストまたはリモートホスト上の任意のブラウザからアクセスできます。このWebインターフェースにより、製品は他社製Webサーバー（Apache HTTP Serverなど）もWebminなどのリモート管理ツールも必要としません。 このコンポーネントの機能はDr.Web HTTPDコンポーネントによって提供されます。
Dr.Web Ctl	コマンドラインからDr.Web for UNIX Mail Serversを管理するためのツール。 このツールによってユーザーは、ファイルスキャンの開始、隔離されたオブジェクトの表示と管理、ウイルスデータベースの更新手順の開始、Dr.Web for UNIX Mail Serversと集中管理サーバー間の接続と切断、製品パラメータの表示と設定を行えるようになります。 実行ファイル：drweb-ctl ログに表示される内部名：Ctl
Dr.Web SNMPD	SNMPエージェント。 Dr.Web for UNIX Mail ServersとSNMP経由の外部モニタリングシステムとの統合用に設計されています。統合することにより、製品のコンポーネントの状態を監視したり、脅威の検出と駆除に関する統計を収集したりできます。 SNMP v2cとv3をサポートします。 実行ファイル：drweb-snmpd ログに表示される内部名：SNMPD

コンポーネント	説明
Dr.Web ClamD	アンチウイルスデーモンである`clamd`（ClamAV®アンチウイルスのコンポーネント）のインターフェースをエミュレートするコンポーネント。 このコンポーネントによって、ClamAV®をサポートするすべてのアプリケーションがアンチウイルススキャンにDr.Web for UNIX Mail Serversを透過的に使用できるようになります。 モードに応じてDr.Web File CheckerまたはDr.Web Network Checkerを使用して、スキャン用のデータをDr.Web Scanning Engineに送信します。 実行ファイル：`drweb-clamd` ログに表示される内部名：`ClamD`

以下の図は、Dr.Web for UNIX Mail Serversの構造とその外部アプリケーションとの動作を示しています。

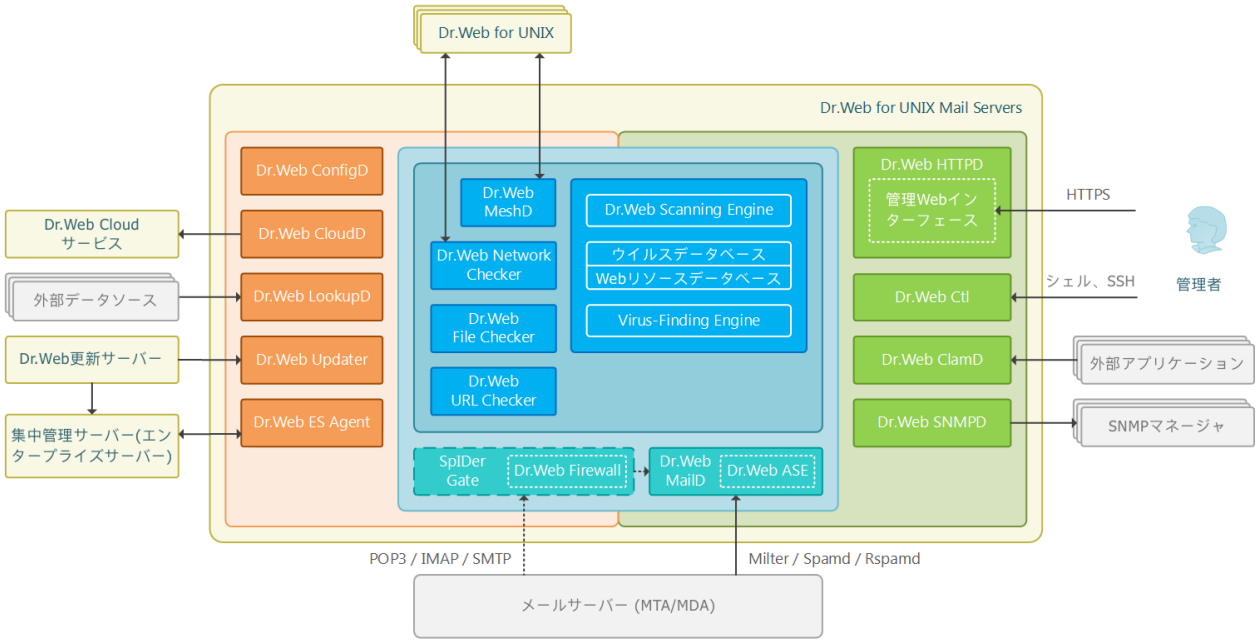

図1. Dr.Web for UNIX Mail Serversの構成

このスキームでは、次の表記法が使用されています。

	- Dr.Web for UNIX Mail Serversの全体像と、ソリューションに含まれない外部のDr.Webアプリケーション。
	- Dr.Web for UNIX Mail Serversの外部にあるプログラムと、それと統合されている製品。
	- 特定のアンチウイルス保護タスク（アンチウイルスデータベースの更新、集中管理サーバーへの接続、動作全体の調整など）を実行するサービスコンポーネント。
	- Dr.Web for UNIX Mail Serversを管理するためのインターフェースを（ユーザーまたは他社製アプリケーションに）提供するコンポーネント。
	- アンチウイルススキャンに使用されるコンポーネント。

	- Dr.Web for UNIX Mail Serversの中核をなす基本的なアンチウイルスコンポーネント。データとファイルのスキャンを実行するコンポーネントによって使用されます。

Dr.Web for UNIX Mail Serversのディストリビューションと使用状況によっては、破線でマークされたコンポーネントが含まれない場合があります。

Dr.Web for UNIX Mail Serversコンポーネントの詳細については、Dr.Web for UNIX Mail Serversコンポーネントを参照してください。

隔離への移動

Dr.Web for UNIX Mail Servers 11.1の隔離ディレクトリは、システムセキュリティに脅威を与え、現在修復できないファイルを分離するのに役立ちます。このような脅威は、Dr.Web for UNIX Mail Serversにとって未知のもの（つまり、ヒューリスティックアナライザによってウイルスが検出されたが、ウイルスのシグネチャおよび修復方法がデータベースにないもの）または修復中にエラーを引き起こしたものになります。さらに、ユーザーが検出された脅威のリストでアクションとして隔離を選択した場合や、脅威のタイプごとにアクションとして隔離を設定で指定した場合は、ユーザーの要求に応じてファイルを隔離できます。

隔離されたファイルの名前は特別なルールに従って変更されます。隔離されたファイルの名前を変更することで、ユーザーやアプリケーションによって特定されることを防ぎ、Dr.Web for UNIX Mail Serversに備わった隔離管理ツールを回避してそれらにアクセスしようとする試みを困難にします。また、ファイルが隔離に移されると、それらを起動させる試みを防ぐために実行ビットがリセットされます。

隔離ディレクトリは以下の場所にあります。

- *ユーザーのホームディレクトリ*（コンピューター上に複数のユーザーアカウントが存在する場合、各ユーザーに個別の隔離ディレクトリが作成される可能性があります）
- ファイルシステムにマウントされた*各論理ボリュームのルートディレクトリ*

Dr.Web隔離ディレクトリの名前は常に`.com.drweb.quarantine`となり、**隔離**アクションが適用されるまで作成されません。その際、オブジェクトを隔離するために必要なディレクトリのみが作成されます。ディレクトリを選択する際は、ファイル所有者の名前を使用します。検索は、悪意のあるオブジェクトのある場所から上の階層に向かって行われ、所有者のホームディレクトリに到達した場合、このディレクトリに作成された隔離ストレージが選択されます。そうでない場合、ファイルはボリュームのルートディレクトリ内に作成された隔離内に移されます（これはファイルシステムのルートディレクトリと同じではない場合があります）。したがって、隔離に移動された感染ファイルは常にボリューム上に配置されるため、複数のリムーバブルデータストレージやその他のボリュームがシステム内の異なる場所にマウントされている場合でも、隔離は正しく動作します。

ユーザーは、ユーティリティDr.Web Ctlを使用してコマンドラインから隔離コンテンツを管理できる他、管理用Webインターフェースからも管理できます（インストールされている場合）。すべてのアクションは統合された隔離に適用されます。つまり、変更はその時点で利用可能なすべての隔離ディレクトリに影響します。

	隔離されたオブジェクトに対する操作は、有効なライセンスが見つからない場合でも行うことができます。ただし、この場合、隔離されたオブジェクトは修復できません。

Dr.Web for UNIX Mail Serversのすべてのアンチウイルスコンポーネントが脅威の分離に隔離を使用できるわけではありません。たとえば、Dr.Web ClamDやDr.Web ICAPD（お使いの製品には含まれていません）、Dr.Web MailDコンポーネントでは使用されません。

ファイルのパーミッションと権限

ファイルシステムのオブジェクトをスキャンし、脅威を駆除するために、Dr.Web for UNIX Mail Servers（を動作させるユーザー）は以下のパーミッションを必要とします。

アクション	必要な権限
検出されたすべての脅威を一覧にする	制限されていません。特別なパーミッションは必要ありません。
アーカイブのコンテンツを一覧表示する （破損した、または悪意のあるエレメントのみを表示する）	制限されていません。特別なパーミッションは必要ありません。
隔離へ移動する	制限されていません。その読み込みまたは書き込み権限に関係なく、ユーザーは感染したすべてのファイルを隔離できます。
脅威を削除する	ユーザーは削除するファイルに対する書き込み権限を持っている必要があります。 コンテナ（アーカイブ、メールメッセージなど）内のファイルで脅威が検出された場合は、削除アクションの代わりにコンテナの隔離への移動が実行されます。
修復する	制限されていません。アクセス権限と修復されたファイルの所有者は修復後も変わりません。 検出された脅威を削除することによって修復が可能である場合、ファイルを削除できます。
隔離からファイルを復元する	ユーザーはファイルの読み込み権限と復元先ディレクトリへの書き込み権限を持っている必要があります。
隔離からファイルを削除する	ユーザーは隔離へ移されたファイルへの書き込み権限を持っている必要があります。

スーパーユーザー（*root*）権限でコマンドライン管理Dr.Web Ctlツールの動作を有効にするには、`su`コマンドを使用してユーザーを変更するか、`sudo`コマンドを使用して、他のユーザーとしてコマンドを実行できます。

Dr.Web Scanning Engineスキャンエンジンは、4GBを超えるサイズのファイルをスキャンできません（このようなファイルをスキャンしようとすると、ファイルサイズが大きすぎることを示すエラーメッセージ "*File is too large*" が表示されます）。

動作モード

Dr.Web for UNIX Mail Serversはスタンドアロンモードでも、*集中管理サーバー*によって管理される*アンチウイルスネットワーク*の一部としても動作できます。*集中管理モード*での動作には、追加のソフトウェアのインストールやDr.Web for UNIX Mail Serversの再インストールまたは削除は必要ありません。

- *スタンドアロンモード*では、保護するコンピューターはアンチウイルスネットワークに接続されず、その操作はローカルで管理されます。このモードでは、設定ファイルとライセンスキーファイルはローカルディスクにあり、Dr.Web for UNIX Mail Serversは保護するコンピューターから完全に制御されます。ウイルスデータベースの更新はDoctor Web更新サーバーから受信します。
- *集中管理モード(エンタープライズモード)では*、コンピューターの保護は集中管理サーバーによって管理されます。このモードでは、Dr.Web for UNIX Mail Serversの一部の機能や設定を、アンチウイルスネットワークに対して適用される一般的な(企業の)アンチウイルス保護ポリシーに応じて変更できます。集中管理モードでの動作に使用するライセンスキーファイルは、集中管理サーバーから受け取ります。ローカルコンピューター上に保存されたデモキーファイルがある場合、それは使用されません。ウイルスイベントに関する統計は、Dr.Web for UNIX Mail Serversの動作に関する情報と一緒に集中管理サーバーに送信されます。ウイルスデータベースの更新もまた、集中管理サーバーから受け取ります。
- *モバイルモードでは*、Dr.Web for UNIX Mail ServersはDoctor Web更新サーバーから更新を受信しますが、製品の動作はローカルの設定と集中管理サーバーから受信したライセンスキーファイルで管理されます。モバイルモードに切り替えることができるのは、集中管理サーバーの設定で許可されている場合のみです。

集中管理のコンセプト

Doctor Webの集中管理ソリューションはクライアント-サーバーモデルを使用します(下図参照)。

ワークステーションとサーバーは、それらにインストールされている*ローカルのアンチウイルスコンポーネント*(以下「Dr.Web for UNIX Mail Serversコンポーネント」)によって脅威から保護されます。これらコンポーネントはリモートコンピューターのアンチウイルス保護を提供し、ワークステーションと集中管理サーバーとの接続を可能にします。

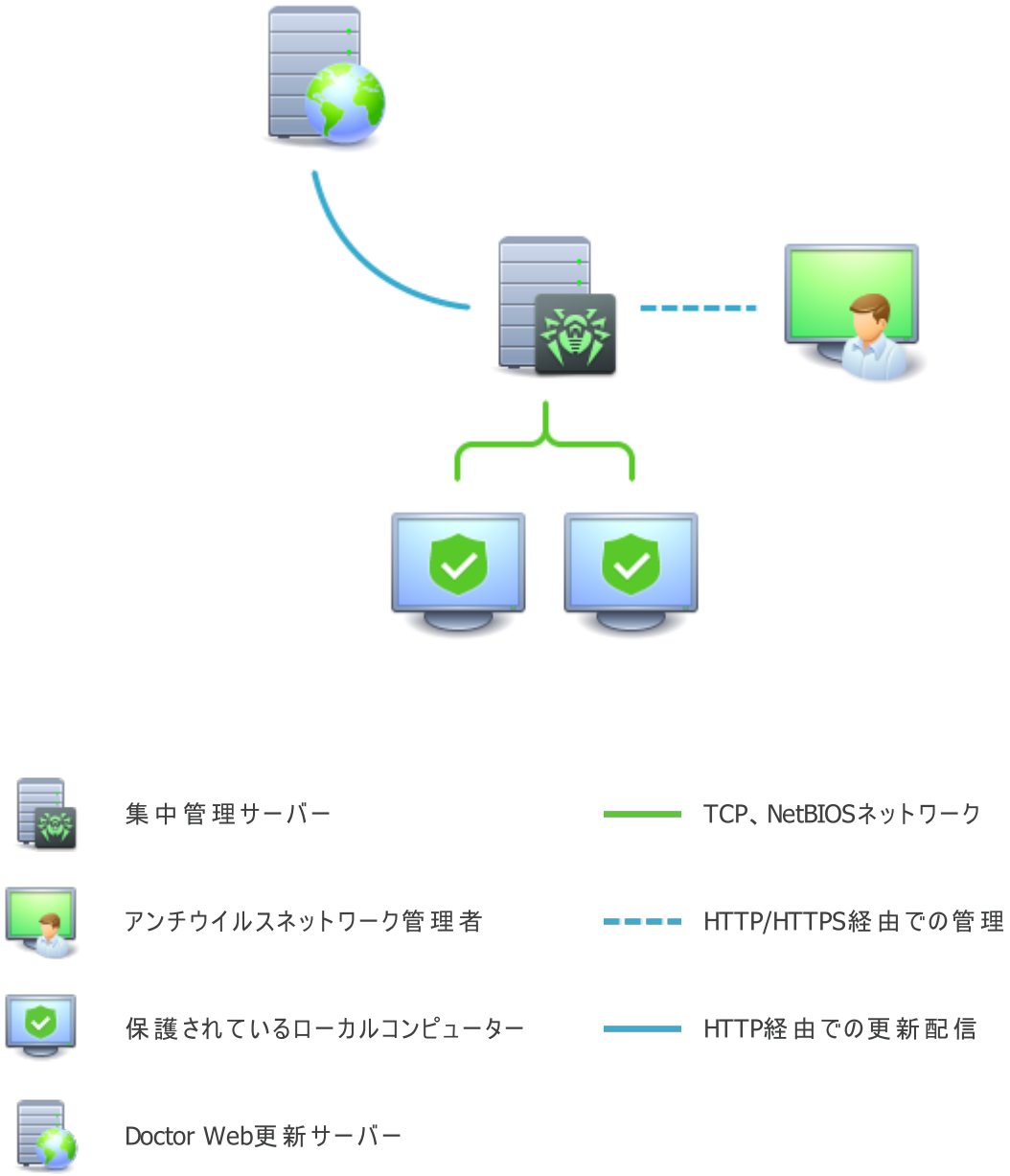

図2. アンチウイルスネットワークの論理的構造

ローカルコンピューターの更新と設定は*集中管理サーバー*から行われます。アンチウイルスネットワーク内の一連の指示やデータ、統計も集中管理サーバーを経由します。保護するコンピューターと集中管理サーバー間のトラフィック量はかなり大きくなる場合があるため、トラフィックを圧縮するオプションがソリューションに用意されています。機密データの漏洩や、保護するコンピューター上にダウンロードされたソフトウェアの置き換えを防ぐため、暗号化もサポートされています。

必要なすべての更新がDoctor Web更新サーバーから集中管理サーバーにダウンロードされます。

ローカルのアンチウイルスコンポーネントは、アンチウイルスネットワーク管理者より受け取ったコマンドに応じて集中管理サーバーから設定・管理されます。管理者は集中管理サーバーとアンチウイルスネットワークのトポロジーを管理し（リモートコンピューターから集中管理サーバーへの接続を検証するなど）、必要に応じてローカルのアンチウイルスコンポーネントの動作を設定します。

ローカルのアンチウイルスコンポーネントは、他社のアンチウイルス製品、または集中管理モードでの動作をサポートしていないDr.Webのアンチウイルスソリューション（Dr.Web Anti-virusバージョン5.0など）と互換性がありません。同一のコンピューターに2つのアンチウイルスプログラムをインストールすると、システムクラッシュや重要なデータの消失を引き起こす場合があります。

集中管理モードでは、集中管理センターを使用して動作レポートをエクスポートできます。レポートはHTML、CSV、PDF、XML形式でエクスポートできます。

集中管理サーバーとの接続

Dr.Web for UNIX Mail Serversは、コマンドラインベース管理ツールDr.Web Ctlの`esconnect`コマンドを使用して、アンチウイルスネットワークの集中管理サーバーに接続できます。

集中管理サーバーの検証には、サーバーが使用する一意のパブリックキーに対応する証明書が使用されます。証明書ファイルを指定しない限り、Dr.Web ES Agent集中管理エージェントはデフォルトでサーバーへの接続を許可しません。証明書ファイルは、Dr.Web for UNIX File Serversを接続するサーバーが提供するアンチウイルスネットワークの管理者から入手する必要があります。

Dr.Web for UNIX Mail Serversが集中管理サーバーに接続されている場合は、製品をモバイルモードに切り替えたり、集中管理モードに戻したりできます。モバイルモードのオン／オフの切り替えは、Dr.Web ES Agentコンポーネントの`MobileMode`設定パラメータを使用して行います。

Dr.Web for UNIX Mail Serversは、集中管理サーバーの設定で許可されている場合にのみ、モバイルモードに切り替えることができます。

製品をアンチウイルスネットワークから切断する

Dr.Web for UNIX Mail Serversは、コマンドラインベース管理ツールDr.Web Ctlの`esdisconnect`コマンドを使用して、アンチウイルスネットワークの集中管理サーバーから切断できます。

システム要件と互換性

このセクションの内容

- システム要件
- サポートされているOSバージョンのリスト
- 追加のパッケージとコンポーネント
- 免責事項
- サポート対象のメールサーバー（MTA）
- セキュリティサブシステムとの互換性

システム要件

Dr.Web for UNIX Mail Serversは、以下の要件を満たすコンピューターで使用できます。

コンポーネント	要件
プラットホーム	次のアーキテクチャとコマンドシステムのプロセッサがサポートされています。 • Intel/AMD: 32ビット（*IA-32*、*x86*）、64ビット（*x86-64*、*x64*、*amd64*） • ARM64 • E2K (*Elbrus*) • IBM POWER (*ppc64el*)
RAM空き容量	500 MB以上（1 GB以上を推奨）
ディスク空き容量	製品ディレクトリが配置されるボリュームに少なくとも2 GB
オペレーティングシステム	GNU/Linux（カーネルバージョン2.6.37以降、`glibc`ライブラリ2.13以降、`systemd` 初期化システムバージョン209以降）、FreeBSD。サポートされているオペレーティングシステムのバージョンは下記を参照してください オペレーティングシステムは、PAM認証メカニズムをサポートしている必要があります
その他	次の有効なネットワーク接続： • ウイルスデータベースとDr.Webコンポーネントの更新を可能にするための有効なインターネット接続 • 集中管理モードで動作している場合は、ローカルネットワーク上の集中管理サーバーへの接続

コンポーネントDr.Web Firewall for Linuxが正しく動作するためには、以下のオプションを指定してOSカーネルが構築されている必要があります。

- *CONFIG_NETLINK_DIAG, CONFIG_INET_TCP_DIAG;*
- *CONFIG_NF_CONNTRACK_IPV4, CONFIG_NF_CONNTRACK_IPV6, CONFIG_NF_CONNTRACK_EVENTS;*
- *CONFIG_NETFILTER_NETLINK_QUEUE, CONFIG_NETFILTER_NETLINK_QUEUE_CT, CONFIG_NETFILTER_XT_MARK.*

上記のリストのうち必要なオプションの組み合わせは、使用するOSバージョンキットによって異なります。

Dr.Web for UNIX Mail Serversを正しく動作させるために、以下のポートを開いてください。

目的	方向	ポート番号
更新を受け取るため	送信	80
Dr.Web Cloudサービスに接続するため	送信	2075（TCP、UDP） 3010（TCP） 3020（TCP） 3030（TCP） 3040（TCP）

サポートされているOSバージョンのリスト

- **GNU/Linux**

プラットホーム	サポートされているGNU/Linuxのバージョン
x86_64	• ALT 8 SP • ALT Server 9、10 • ALT Workstation 9、10 • Astra Linux Common Edition（Orel）2.12 • Astra Linux Special Edition 1.5（累積パッチ20201201SE15）、1.6（累積パッチ20200722SE16）、1.7 • CentOS 7、8 • Debian 9、10、11、12 • Fedora 37、38 • GosLinux IC6 • Red Hat Enterprise Linux 7、8 • RED OS 7.2 MUROM、RED OS 7.3 MUROM • SUSE Linux Enterprise Server 12 SP3 • Ubuntu 18.04、20.04、22.04
x86	• ALT 8 SP • ALT Workstation 9、10 • CentOS 7

プラットホーム	サポートされているGNU/Linuxのバージョン
	• Debian 10
ARM64	• ALT 8 SP • ALT Server 9、10 • ALT Workstation 9、10 • Astra Linux Special Edition (Novorossiysk) 4.7 • CentOS 7、8 • Debian 11、12 • Ubuntu 18.04
E2K	• ALT 8 SP • Astra Linux Special Edition（Leningrad）8.1（累積パッチ 20201201SE15） • Elbrus-D MCST 1.4 • GS CS Elbrus 8.32 TVGI.00311-28
ppc64el	• CentOS 8 • Ubuntu 20.04

ALT 8 SP、Astra Linux Special Edition（Novorossiysk）4.11、Elbrus-D MCST 1.4、GosLinux IC6では強制アクセス制御はサポートされていません。

これらの要件を満たすその他のGNU/Linuxバージョンであっても、Dr.Web for UNIX Mail Serversとの完全な互換性は保証されていません。互換性の問題が発生した場合は、テクニカルサポートにお問い合わせください。

- **FreeBSD**

プラットホーム	サポートされているFreeBSDのバージョン
x86	11, 12, 13
x86_64	11, 12, 13

FreeBSD OSの場合、Dr.Web for UNIX Mail Serversはユニバーサルパッケージからのみインストールできます。

追加のパッケージとコンポーネント

CommuniGate Proメールサーバーとの統合にはPython3.6+パッケージが必要です。

コマンドラインでDr.Web for UNIX Mail Serversを便利に操作するために、使用しているコマンドシェルでコマンド自動補完機能を有効にできます（無効になっている場合）。

追加のパッケージやコンポーネントのインストールで問題が発生した場合は、お使いのOSバージョンのドキュメントを参照してください。

免責事項

- SpIDer Gateは、OSにインストールされている他のファイアーウォール（SUSE Linux Enterprise Server OSのShorewallやSuseFirewall2、Fedora、CentOS、Red Hat Enterprise LinuxのFirewallDなど）と*競合する可能性があります*。競合の兆候は、コードx109を伴うSpIDer Gateのエラーに関するメッセージ、またはコードx102を伴うDr.Web Firewall for Linuxのエラーに関するメッセージです。競合を解決する方法は、それぞれ「既知のエラー」セクションのエラーx109とx102で説明されています。
- 使用しているOSに1.4.15より前のバージョンのNetFilterが含まれている場合、SpIDer Gateは正しく動作しない可能性があります。この問題はNetFilterの内部エラーに関連しており、SpIDer Gateを無効にすると、ネットワーク接続が切断され、再確立できなくなります。この問題が発生した場合は、バージョン1.4.15以降のNetFilterを含むOSにアップグレードすることをお勧めします。この問題を解決する方法は、「既知のエラー」セクションで説明されています。

サポート対象のメールサーバー（MTA）

Dr.Web for UNIX Mail Serversには、メールサーバー（MTA）がインストールされている必要があります。

- プラグインフィルターのモードでDr.Web for UNIX Mail ServersとMTAを統合するには、メールサーバーが外部のスパムおよびアンチウイルスフィルターと統合するためのインターフェース（*Milter*、*Spamd*、*Rspamd*）をサポートしている必要があります。たとえば、次のMTAを使うことができます：Sendmail、Postfix、Exim、CommuniGate Pro。
- *Milter*、*Spamd*、*Rspamd*の統合インターフェースをサポートしていないMTAは、アンチウイルススキャン*Clamd*のインターフェースを介して、Dr.Web ClamDアンチウイルススキャンコンポーネントに直接接続することによりDr.Web for UNIX Mail Serversと統合できます（MTAをインストールして設定するには、追加の統合モジュールが必要になる可能性があります）。この統合モードではDr.Web MailDコンポーネントが使用されないため、メールメッセージのスパムスキャンは行われず、脅威が検出された場合にメールメッセージを再圧縮することもできません。感染したメールメッセージの処理を目的としたすべてのアクションは、メールサーバーに転送されます。メールサーバーは脅威に対するメールメッセージのスキャンの結果を取得します。

統合設定は複雑なため、メールサーバーQmailを操作するには、Dr.Web for UNIX Mail Serversの前バージョン（6.0.2.x）を使用するか、透過プロキシモードを使用することをお勧めします。

- 透過プロキシモードでは、MTAとMDA間、またはMDAとMUA間で透過的にメールメッセージのアンチウイルスおよびアンチスパムスキャンを行うDr.Web for UNIX Mail Serversを統合できます（プロトコルSMTP、POP3、IMAPを介したデータ交換チャネルへの統合が実行されます）。このモードでは、MTAとDr.Web for UNIX Mail Serversを同じホストにインストールする必要があります。

透過プロキシモードでは、Dr.Web for UNIX Mail ServersにSpIDer Gateが含まれている必要があります。SpIDer GateはGNU/Linuxでのみ動作します。

- SMTPプロキシモードは、MTAが受信メッセージのルーティングに設定されている（メールリレーのように機能する）プラグインフィルターのモード（Postfixなど）で、Dr.Web for UNIX Mail ServersとMTAが統合する特殊なケースです。
- Dr.Web Anti-SpamコンポーネントはARM64およびE2Kのアーキテクチャでは *サポートされていません*。

セキュリティサブシステムとの互換性

デフォルトでは、Dr.Web for UNIX Mail ServersはSELinuxをサポートしていません。またDr.Web for UNIX Mail Serversは、強制アクセスモデルを使用するGNU/Linuxシステム（たとえば、ユーザーとファイルに異なる権限レベルを付与するPARSEC強制アクセスサブシステムの備わったシステムなど）では、機能が制限されたモードで動作します。

SELinuxの備わったシステム（およびその他の強制アクセスモデルを使用するシステム）にDr.Web for UNIX Mail Serversをインストールする必要がある場合、Dr.Web for UNIX Mail Serversの全機能が動作するように、セキュリティサブシステムの追加設定を実行する必要があります。詳細は、セキュリティサブシステムを設定するのセクションを参照してください。

ライセンス

Dr.Web for UNIX Mail Serversを使用する権限は、Doctor Web社またはそのパートナーから購入した*ライセンス*によって付与されます。ユーザー権限を特定するライセンスパラメータは、Dr.Web for UNIX Mail Serversのインストール中にユーザーが同意する使用許諾契約（https://license.drweb.com/agreement/を参照）に従って設定されます。ライセンスには、ユーザーとベンダーに関する情報の他、以下を含む購入した製品の使用パラメータも含まれています。

- ユーザーに対してライセンスされたコンポーネントのリスト
- Dr.Web for UNIX Mail Serversのライセンス有効期間
- その他の制限（購入した製品を使用できるコンピューターの台数など）

評価目的のために、ユーザーは*試用期間*を有効化することもできます。有効化が成功すると、ユーザーは、有効化した試用期間全体にわたってDr.Web for UNIX Mail Serversの全機能を使用できます。

各Doctor Web製品ライセンスには、ユーザーのコンピューターに保存されている特別なファイルに関連付けられた一意のシリアル番号があります。このファイルは、ライセンスパラメータに従ってコンポーネントの動作を制限し、*ライセンス*キーファイルと呼ばれます。試用期間が有効になると、*デモ*キーファイルと呼ばれる特別なキーファイルが自動的に生成されます。

コンピューターでライセンスまたは試用期間が有効になっていない場合、Dr.Web for UNIX Mail Serversコンポーネントはブロックされます。さらに、ウイルスデータベースとコンポーネントの更新プログラムをDoctor Web更新サーバーからダウンロードすることもできません。ただし、企業またはインターネットサービスプロバイダーによって管理されるアンチウイルスネットワークの一部として集中管理サーバーに接続することで、Dr.Web for UNIX Mail Serversを有効化することができます。この場合、アンチウイルスの動作と更新は集中管理サーバーによって管理されます。

インストールとアンインストール

このセクションの内容

- Dr.Web for UNIX Mail Serversをインストールする
- Dr.Web for UNIX Mail Serversをアップグレードする
- Dr.Web for UNIX Mail Serversをアンインストールする
- セキュリティサブシステムを設定する
- 追加情報:
  - Dr.Web for UNIX Mail Serversパッケージとファイル
  - コンポーネントのカスタムインストールとアンインストール

このセクションでは、Dr.Web for UNIX Mail Serversバージョン11.1をインストールおよびアンインストールする方法について説明します。また、最新の更新を入手する方法や、Dr.Web for UNIX Mail Serversの以前のバージョンがすでにコンピューターにインストールされている場合に新しいバージョンにアップグレードする手順も記載されています。

さらに、Dr.Web for UNIX Mail Serversコンポーネントのカスタムインストールとアンインストール手順(Dr.Web for UNIX Mail Serversの動作中に生じたエラーの解決方法や、機能セットを限定してインストールする方法など)、Dr.Web for UNIX Mail Serversのインストールと動作に必要になる可能性がある高度なセキュリティサブシステム(SELinuxなど)の設定についても確認できます。

これらの手順を進めるには、スーパーユーザー権限(*root*ユーザーの権限)が必要です。権限を昇格するには、`su`コマンドを使用してカレントユーザーを変更するか、または`sudo`コマンドを使用して、指定されたコマンドを別のユーザーの権限で実行します。

Dr.Web for UNIX Mail Serversと、他社のアンチウイルス製品との互換性は保証されていません。1台のマシンに2つのアンチウイルスがインストールされることで、OSのエラーを引き起こし、重要なデータが失われる可能性があります。Dr.Web for UNIX Mail Serversをインストールする前に、他社のアンチウイルス製品をコンピューターから削除することが強く推奨されます。

お使いのコンピューターに他のDr.Webアンチウイルス製品がユニバーサルパッケージ(`.run`)からすでにインストールされていて、さらに別のDr.Webアンチウイルス製品をインストールする場合(たとえば、ユニバーサルパッケージからDr.Web for Linuxをインストールしていて、さらにDr.Web for UNIX Mail Serversをインストールする場合など)、インストールされている製品のバージョンがインストールするDr.Web for UNIX Mail Serversのバージョンと同じであることを確認してください。新しくインストールする製品のバージョンがインストールされている製品のものよりも新しい場合、インストール前に、インストールされているDr.Web for UNIX Mail Serversを新しくインストールする製品のバージョンにアップグレードする必要があります。

FreeBSD OSの場合、Dr.Web for UNIX Mail Serversはユニバーサルパッケージからのみインストールできます。

Dr.Web for UNIX Mail Serversをインストールする

Dr.Web for UNIX Mail Serversをインストールするには、以下の手順のいずれか1つを行います。

1. Doctor Webの公式Webサイトから、UNIXシステム用のユニバーサルパッケージを含むインストールファイルをダウンロードします。パッケージにはインストーラが含まれています（インストールプログラムはコマンドラインモード用に開発されているため、グラフィカルデスクトップモードで使用するにはターミナルエミュレーターが必要になります）。
2. Doctor Webの対応するパッケージリポジトリからネイティブパッケージをダウンロードします。

> FreeBSD OSの場合、Dr.Web for UNIX Mail Serversはユニバーサルパッケージからのみインストールできます。

> 古いバージョンのパッケージマネージャーを使用しているOS（ALT 8 SPなど）では、Dr.Web for UNIX Mail Serversユニバーサルパッケージをインストールすることを推奨します。

インストール中（ユニバーサル.`run`パッケージからだけでなく、パッケージマネージャーを使用してネイティブパッケージからも）、インストール結果を含むメッセージが`root@localhost`メールアドレスに送信されます。

上記のいずれかの方法でDr.Web for UNIX Mail Serversをインストールした後、そのコンポーネントに利用可能な修正があるか、新しいDr.Web for UNIX Mail Serversバージョンがリリースされている場合は、アンインストールまたは更新できます。必要に応じて、Dr.Web for UNIX Mail Serversを正しく動作させるために、GNU/Linuxのセキュリティサブシステムを設定することもできます。個々のコンポーネントの機能に問題がある場合は、Dr.Web for UNIX Mail Serversをアンインストールせずに、該当するコンポーネントのカスタムインストールおよびアンインストールを実行できます。

選択したDr.Web for UNIX Mail Serversのインストール方法に関係なく、インストールが完了したら、ライセンスを有効化して、受け取ったキーファイルをインストールする必要があります。さらに、Dr.Web for UNIX Mail Serversを集中管理サーバーに接続することもできます。詳細は、ライセンスを参照してください。これを行わない場合、アンチウイルス保護機能が無効になります。さらに、場合によっては、開始するのセクションで説明されているように、インストールしたDr.Web for UNIX Mail Serversの基本機能をカスタマイズする必要があります。

ユニバーサルパッケージをインストールする

Dr.Web for UNIX Mail Serversユニバーサルパッケージは、`drweb-<version>-av-mail-<OS>-<platform>.run`という名前のインストールファイルとして提供されます。ここで、*<OS>*はUNIXライクなOSのタイプ、*<Platform>*は、Dr.Web for UNIX Mail Serversが対象としているプラットフォーム（32ビットプラットフォームは`x86`、64ビットプラットフォームは`amd64`、`arm64`、`e2s`）です。例：

```
drweb-11.1.0-av-mail-linux-x86.run
```

このセクションではこれ以降、上記のフォーマットに対応するインストールファイルの名前を*<file_name>*`.run`と表記します。

Dr.Web for UNIX Mail Serversのコンポーネントをインストールするには以下の手順を行ってください。

1. ユニバーサルパッケージを含むインストールファイルがない場合は、Doctor Webの公式Webサイト（https://download.drweb.com/）からダウンロードしてください。
2. インストールファイルをコンピューターのハードディスクドライブに保存します。
3. たとえば次のコマンドを使用して、アーカイブを実行できるようにします。

   ```
   # chmod +x <file_name>.run
   ```

4. 次のコマンドを使用してアーカイブを実行します。

   ```
   # ./<file_name>.run
   ```

   ファイルプロパティ（パーミッション）を変更するときやファイルを実行するときは、どちらの場合にもグラフィカルシェルの標準的なファイルマネージャーを使用します。これにより、アーカイブの整合性チェックが実行され、その後にアーカイブファイルが一時ディレクトリに解凍され、インストールプログラムが起動されます。ユーザーがroot権限を持っていない場合、インストールプログラムはそのユーザーにrootパスワードを要求することで権限の昇格を試みます（`sudo`が使用されます）。この試みが失敗した場合、インストールプロセスは停止します。

   

   ファイルシステム内の一時ディレクトリへのパスに、解凍されるファイル用の十分な空き容量がない場合、インストールプロセスが停止し、対応するメッセージが表示されます。この場合は、`TMPDIR`システム環境変数の値を、十分な空き容量があるディレクトリを指すように変更し、インストールをやり直してください。`--target`オプションを使用することもできます。

   その後、コマンドラインモード用のインストーラが自動的に起動されます（グラフィカルデスクトップ環境で実行するには、ターミナルエミュレーターが必要です）。
5. インストーラの指示に従ってください。
6. 次のコマンドを実行することで、インストールプログラムをサイレントモードで起動することもできます。

   ```
   # ./<file_name>.run -- --non-interactive
   ```

   この場合、インストールプログラムはサイレントモードで起動され、ユーザーインターフェースなしで動作します（通常コマンドラインモードで表示されるダイアログも表示されません）。

このオプションを使用すると、Dr.Web使用許諾契約の規定に同意することになります。使用許諾契約のテキストは/opt/drweb.com/share/doc/LICENSEファイルにあります。ファイル拡張子は使用許諾契約の言語を示しています。このLICENSEファイルに拡張子がない場合、Dr.Web使用許諾契約は英語で記述されています。使用許諾契約の規定に同意しない場合は、インストール後にDr.Web for UNIX Mail Serversをアンインストールしてください。

インストールプログラムをサイレントモードで起動するには管理者(root)権限が必要です。権限を昇格するには、suまたはsudoコマンドを使用できます。

使用しているGNU/Linuxディストリビューションに SELinuxがある場合、インストールプロセスがセキュリティサブシステムによって中断される可能性があります。このような状況になった場合は、次のコマンドでSELinuxを*Permissive*モードに設定してください。

```
# setenforce 0
```

次に、インストーラを再起動させます。インストールが完了した後、製品コンポーネントが正常に動作するよう、SELinuxのセキュリティポリシーを設定します。

<opt_dir>、*<etc_dir>*、*<var_dir>*の表記規則の詳細は、はじめにを参照してください。

インストールプロセスが完了すると、解凍されたインストールファイルがすべて削除されます。

<file_name>.runファイル(インストール元のファイル)を保存しておくことを推奨します。これにより、バージョンの更新を必要とせずにDr.Web for UNIX Mail Serversやそのコンポーネントを再インストールすることが可能になります。

コマンドラインからインストールする

コマンドラインベースのインストールプログラムを起動すると、製品をインストールするように促すメッセージが表示されます。

1. インストールを開始するには、「Do you want to continue?」の質問に対してYesまたはYを入力します。インストーラを終了するには、NoまたはNと入力します。この場合、インストールはキャンセルされます。
2. その後、画面に表示されているDr.Web使用許諾契約の規約を確認する必要があります。ENTERキーを押してテキストを1行ずつ下にスクロールするか、またはSPACEキーを押してテキストを1画面ずつ下にスクロールします。

   使用許諾契約を上にスクロールするオプションはありません。

3. 使用許諾契約のテキストを読んだ後、規約に同意するように求められます。使用許諾契約に同意する場合は、YesまたはYを入力します。同意しない場合は、NoまたはNを入力します。後者の場合、インストーラは終了します。
4. 使用許諾契約の規約に同意すると、インストールが自動的に開始されます。手順の実行中、インストールされるDr.Webコンポーネントの一覧を含むインストールプロセスに関する情報が画面に表示されます。

5. インストールが正常に完了すると、Dr.Web for UNIX Mail Serversの動作を管理する方法を通知するメッセージが表示されます。

エラーが発生した場合は、エラーについて説明するメッセージが画面に表示された後、インストーラが終了します。インストールがエラーによって失敗した場合、エラーの原因を取り除き、インストールを再度開始してください。

リポジトリからインストールする

Dr.Web for UNIX Mail Serversのネイティブパッケージはhttps://repo.drweb.com/にあるDr.Webの公式リポジトリに保存されています。お使いのOSのパッケージマネージャーが使用するリポジトリのリストにDr.Webリポジトリを追加すると、OSのリポジトリから他のプログラムをインストールするのと同じようにネイティブパッケージから製品をインストールできます。必要な依存関係は自動的に解決されます。

以下で説明するコマンド(リポジトリの追加、デジタル署名キーのインポート、パッケージのインストールと削除に使用するコマンド)はすべて、スーパーユーザー(root)権限で実行する必要があります。この権限を昇格するには、(現在のユーザーを変更する)`su`コマンド、または(他のユーザーの権限で指定したコマンドを実行する)`sudo`コマンドを使用します。

FreeBSD OSの場合、Dr.Web for UNIX Mail Serversはユニバーサルパッケージからのみインストールできます。

以下のOS(パッケージマネージャー)の手順を参照してください。

- Debian、Mint、Ubuntu(apt)
- ALT Linux、PCLinuxOS(apt-rpm)
- Mageia、OpenMandriva Lx(urpmi)
- Red Hat Enterprise Linux、Fedora、CentOS(yum、dnf)
- SUSE Linux(zypper)

Debian、Mint、Ubuntu(apt)

リポジトリから Dr.Web for UNIX Mail Servers をインストールする

1. これらOS用のリポジトリはDoctor Webによって電子署名されています。リポジトリにアクセスするには、以下のコマンドを実行することで、デジタル署名キーをインポートし、パッケージマネージャーストレージに追加します。

```
# apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
8C42FC58D8752769
```

2. リポジトリを追加するには、`/etc/apt/sources.list`ファイルに以下のラインを追加します。

```
deb http://repo.drweb.com/drweb/debian 11.1 non-free
```

1および2の項目は、リポジトリから特別なDEBパッケージ（https://repo.drweb.com/drweb/drweb-repo11.1.deb）をダウンロードしてインストールすることでも実行できます。

3. リポジトリからDr.Web for UNIX Mail Serversをインストールするには、以下のコマンドを使用します。

```
# apt-get update
# apt-get install drweb-mail-servers
```

代わりのパッケージマネージャー（Synapticまたはaptitudeなどⅰ）を使用して製品をインストールすることもできます。パッケージの競合が発生した場合は、それを解決するためにaptitudeなどの代わりのマネージャーを使用することが推奨されます。

ALT Linux、PCLinuxOS（apt-rpm）

リポジトリからDr.Web for UNIX Mail Serversをインストールする

1. リポジトリを追加するには、/etc/apt/sources.listファイルに以下のラインを追加します。

```
rpm http://repo.drweb.com/drweb/altlinux 11.1/<arch> drweb
```

ここで、*<arch>*は、次に示すパッケージのアーキテクチャを表します。

- 32-bitバージョン：i386
- AMD64アーキテクチャ：x86_64
- ARM64アーキテクチャ：aarch64
- E2Kアーキテクチャ：e2s

2. リポジトリからDr.Web for UNIX Mail Serversをインストールするには、以下のコマンドを使用します。

```
# apt-get update
# apt-get install drweb-mail-servers
```

代わりのパッケージマネージャー（Synapticまたはaptitudeなどⅰ）を使用して製品をインストールすることもできます。

Mageia、OpenMandriva Lx（urpmi）

リポジトリからDr.Web for UNIX Mail Serversをインストールする

1. 以下のコマンドを使用してリポジトリを接続します。

```
# urpmi.addmedia drweb https://repo.drweb.com/drweb/linux/11.1/<arch>/
```

ここで、*<arch>*は、次に示すパッケージのアーキテクチャを表します。

- 32-bitバージョン：i386
- 64-bitバージョン：x86_64

2. リポジトリからDr.Web for UNIX Mail Serversをインストールするには、以下のコマンドを使用します。

```
# urpmi drweb-mail-servers
```

代わりのパッケージマネージャー（rpmdrakeなど）を使用して製品をインストールすることもできます。

Red Hat Enterprise Linux、Fedora、CentOS（yum、dnf）

リポジトリからDr.Web for UNIX Mail Serversをインストールする

1. 以下のコンテンツが含まれたdrweb.repoファイルを/etc/yum.repos.dディレクトリに追加します。

```
[drweb]
name=DrWeb-11.1
baseurl=https://repo.drweb.com/drweb/linux/11.1/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://repo.drweb.com/drweb/drweb.key
```

echoなどのコマンドを使用して上記のコンテンツをファイルにロギングし、出力をリダイレクトする場合は、$記号をエスケープする必要があります（\$）。

項目1は、リポジトリから特別なRPMパッケージ（https://repo.drweb.com/drweb/drweb-repo11.1.rpm）をダウンロードしてインストールすることでも実行できます。

2. リポジトリからDr.Web for UNIX Mail Serversをインストールするには、以下のコマンドを使用します。

```
# yum install drweb-mail-servers
```

Fedoraのバージョン22以降では、マネージャーyumの代わりにdnfを使用することが推奨されます。例：

```
# dnf install drweb-mail-servers
```

代わりのパッケージマネージャー（PackageKitまたはYumexなど）を使用して製品をインストールすることもできます。

SUSE Linux（zypper）

リポジトリからDr.Web for UNIX Mail Serversをインストールする

1. リポジトリを追加するには、以下のコマンドを使用します。

```
# zypper ar https://repo.drweb.com/drweb/linux/11.1/\$basearch/ drweb
```

2. リポジトリからDr.Web for UNIX Mail Serversをインストールするには、以下のコマンドを使用します。

```
# zypper refresh
# zypper install drweb-mail-servers
```

代わりのパッケージマネージャー（YaSTなど）を使用して製品をインストールすることもできます。

Dr.Web for UNIX Mail Serversをアップグレードする

Dr.Web for UNIX Mail Serversには2つの更新モードがあります。

1. Dr.Web for UNIX Mail Serversの現在のバージョンのサポート期間にリリースされたパッケージやコンポーネントの更新を入手する。通常、このような更新ではエラー修正やコンポーネント機能の軽微な改良が行われています。
2. Dr.Web for UNIX Mail Serversの新しいバージョンにアップグレードする。このアップグレードオプションは、お使いのDr.Web for UNIX Mail Serversの新しいバージョンをDoctor Webがリリースし、それに新しい機能が備わっている場合に使用されます。

保護されたサーバーでインターネットにアクセスできない場合でも、Dr.Web for UNIX Mail Serversではウイルスデータベースやアンチウイルスエンジンを更新できます。

パッケージとコンポーネントを更新する

該当するセクションに記載されている方法を使用してDr.Web for UNIX Mail Serversをインストールすると、パッケージマネージャーは自動的にDr.Webパッケージリポジトリに接続します。

- インストールがユニバーサルパッケージ（ファイル `.run`）から実行され、システムでDEBパッケージが使用されていたり（たとえば、Debian、Mint、UbuntuなどのOS）、OSにパッケージマネージャーがない場合（FreeBSD）、Dr.Webパッケージの動作には、それぞれのバージョンのパッケージマネージャー`zypper`が使用されます。これはDr.Web for UNIX Mail Serversのインストール時に自動的にインストールされます。

  このマネージャーが含まれる、更新されたDr.Webパッケージを入手してインストールするには、*<opt_dir>*`/bin`ディレクトリ（GNU/Linuxの場合は`/opt/drweb.com/bin`）に移動し、次のコマンドを実行します。

  ```
  # ./zypper refresh
  # ./zypper update
  ```

  FreeBSD OS 11.x for `amd64`では、更新に`zypper`マネージャーを使用すると、リポジトリの更新エラーが発生する場合があります。この場合は`compat10x-amd64`サポートパッケージをインストールして、再試行してください。

  パッケージをインストールするには、次のコマンドを使用します。

  ```
  # pkg install compat10x-amd64
  ```

- それ以外の場合は、お使いのOSで使用されているパッケージマネージャーの更新コマンドを使用します。次に例を示します。
  - Red Hat Enterprise LinuxとCentOSでは`yum`コマンドを使用します。
  - Fedoraでは`yum`または`dnf`コマンドを使用します。
  - SUSE Linuxでは`zypper`コマンドを使用します。
  - MageiaとOpenMandriva Lxでは`urpmi`コマンドを使用します。

- Alt Linux、PCLinuxOS、Debian、Mint、Ubuntuでは`apt-get`コマンドを使用します。

また、お使いのOS用に開発された別のパッケージマネージャーを使用することもできます。必要に応じて、使用しているパッケージマネージャーのマニュアルを参照してください。

新しいバージョンのDr.Web for UNIX Mail Serversがリリースされると、そのコンポーネントを含むパッケージは、新しいバージョンに対応するDr.Webリポジトリのセクションに配置されます。この場合、パッケージマネージャーを新しいDr.Webリポジトリセクションに切り替えて更新する必要があります（新しい製品バージョンにアップグレードするを参照）。

新しい製品バージョンにアップグレードする

このセクションの内容

- 注意事項
- アップグレードのためにユニバーサルパッケージをインストールする
- リポジトリからアップグレードする
- キーファイルの転送
- 集中管理サーバーとの接続を復元する

注意事項

新しいバージョンにアップグレードする前に、サーバーが新しいバージョンのシステム要件を満たしていることを確認してください。これには、必要なプログラムがインストールされていることを含みます（たとえば、プロキシモードでのDr.Web for UNIX Mail Servers動作の場合、MTA（メールサーバー）が必要です）。

お使いのバージョンのDr.Web for UNIX Mail Serversは、製品のインストール時と同じ方法でアップグレードする必要があります。

- Dr.Web for UNIX Mail Serversの現在のバージョンがリポジトリからインストールされている場合、アップグレードではリポジトリからのプログラムパッケージを更新する必要があります。
- Dr.Web for UNIX Mail Serversの現在のバージョンがユニバーサルパッケージからインストールされている場合、Dr.Web for UNIX Mail Serversをアップグレードするには、新しいバージョンの製品を含んでいる別のユニバーサルパッケージをインストールする必要があります。

製品バージョンのインストール方法を特定するには、Dr.Web for UNIX Mail Serversの実行可能ディレクトリに`uninst.sh`のアンインストールスクリプトが存在するかどうかを確認します。存在する場合、現在のバージョンはユニバーサルパッケージからインストールされています。存在しない場合には、リポジトリからインストールされています。

FreeBSD OSの場合、Dr.Web for UNIX Mail Serversはユニバーサルパッケージからのみインストールできます。

インストールに使用した方法でDr.Web for UNIX Mail Serversを更新できない場合は、現在のバージョンのDr.Web for UNIX Mail Serversをアンインストールしてから、何らかの方法で新しいバージョンをインストールしてください。以前のバージョンのDr.Web for UNIX Mail Serversのインストールおよびアンインストール手順は、

バージョン11.1向けの本マニュアルにあるインストールおよびアンインストールと同じです。追加情報については、最新バージョンのDr.Web for UNIX Mail Serversのユーザーマニュアルを参照してください。

Dr.Web for UNIX Mail Serversが集中管理モードで動作している場合は、集中管理サーバーのアドレスを記録することをお勧めします。また、サーバー証明書ファイルを保存することをお勧めします。

現在使用している接続パラメータを調べる際に問題が発生した場合は、現在お使いのバージョンのDr.Web for UNIX Mail Serversの管理者マニュアルをご確認いただき、アンチウイルスネットワーク管理者までお問い合わせください。

アップグレードのためにユニバーサルパッケージをインストールする

ユニバーサルパッケージからDr.Web for UNIX Mail Servers 11.1をインストールします。自動更新が不可能な場合は、新しいバージョンのインストール中に、コンピューターにインストールされているDr.Web for UNIX Mail Serversの古いバージョンのコンポーネントを自動的に削除するメッセージが表示されます。

更新中に、インストールされたバージョンのDr.Web for UNIX Mail Serversを削除する必要があり、Dr.Webの複数のサーバー製品（ファイルサーバー用、メールサーバー用、インターネットゲートウェイ用の製品など）がお使いのサーバーに一緒にインストールされている場合、アップグレードされない他のサーバー製品を完全に機能させるために（すなわち、ファイルサーバー用およびインターネットゲートウェイ用の製品を維持するために）、削除対象として以下のパッケージのみを選択する必要があります。

- `drweb-mail-servers-doc`
- `drweb-maild`.

リポジトリからアップグレードする

Dr.Webバージョン6.0.2の複数のサーバー製品がお使いのサーバーに一緒にインストールされている場合（ファイルサーバー用製品、メールサーバー用製品、インターネットゲートウェイ用製品がインストールされている場合など）、Dr.Web for UNIX Mail Servers 6.0.2をリポジトリからバージョン11.1にアップグレードすることはできません。この場合は、新しいバージョンのDr.Web for UNIX Mail Serversを別のマシンにインストールしてください。

Doctor Webのリポジトリからインストールされた現在のバージョンのDr.Web for UNIX Mail Serversをアップグレードする

1. 現在のバージョンのDr.Web for UNIX Mail Serversをアンインストールします。
2. リポジトリを変更します（現在のバージョンのパッケージリポジトリから11.1パッケージリポジトリへ）。
3. リポジトリから新しいバージョンのDr.Web for UNIX Mail Serversをインストールします。

11.1パッケージが保存されているリポジトリの名前は、リポジトリからインストールするのセクションで確認できます。リポジトリの変更方法の詳細については、お使いのOSディストリビューションのヘルプガイドを参照してください。

キーファイルの転送

Dr.Web for UNIX Mail Serversをアップグレードするために選択した方法に関係なく、現在のライセンスキーファイル（お持ちの場合）は自動的に転送され、新しいバージョンに必要な正しい場所にインストールされます。

> キーファイルの自動インストール中に問題が発生した場合は、手動でインストールできます。

有効なライセンスキーファイルを紛失した場合は、テクニカルサポートに連絡してください。

集中管理サーバーとの接続を復元する

可能であれば、アップグレード後に集中管理サーバーへの接続が自動的に復元されます（アップグレード前に製品が集中管理サーバーに接続されていた場合）。接続が自動的に復元されなかった場合は、アップグレードしたDr.Web for UNIX Mail Serversのアンチウイルスネットワークへの接続を再度確立するために、次のコマンドを実行します。

```
$ drweb-ctl esconnect <address> --Certificate <path to the certificate file>
```

接続処理に問題が発生した場合は、アンチウイルスネットワークの管理者までお問い合わせください。

インターネットに接続せずにデータベースを更新する

インターネット接続がブロックまたは制限されている高度に安全な環境では、*オフラインでウイルスデータベースを更新*できます。その場合は、インターネットに接続されているコンピューターに更新をダウンロードし、USBドライブまたはローカルネットワーク共有にコピーしてから、別の（インターネットに接続されていない）コンピューターにインストールする必要があります。更新手順はコマンドラインから実行する必要があります。

更新を入手する

1. インターネットに接続されているコンピューターで次のコマンドを実行します。

   ```
   $ drweb-ctl update --Path <a path to a directory to store updates>
   ```

2. ダウンロードした更新をUSBドライブまたはローカルネットワーク共有にコピーします。
3. 更新するコンピューターにローカルネットワーク共有またはリムーバブルドライブをマウントします。USBドライブから更新する場合は、次のコマンドを実行します。

   ```
   # mkdir /mnt/usb
   # mount <a path to the device> /mnt/usb
   ```

4. 次のコマンドで更新を適用します。

   ```
   $ drweb-ctl update --From /mnt/usb
   ```

Dr.Web for UNIX Mail Serversをアンインストールする

Dr.Web for UNIX Mail Serversをインストールした方法に応じて、次のいずれかの方法で製品を削除できます。

1. アンインストーラを起動して、ユニバーサルパッケージをアンインストールする。
2. システムのパッケージマネージャーを使用して、Doctor Webのリポジトリからインストールしたパッケージをアンインストールする。

ユニバーサルパッケージをアンインストールする

UNIXシステム向けのユニバーサルパッケージからインストールされたDr.Web for UNIX Mail Serversは、コマンドラインからアンインストールできます（このオプションでは、グラフィカルデスクトップ環境を使用している場合、端末エミュレーターが必要です）。

アンインストールツールはDr.Web for UNIX Mail Serversだけでなく、コンピューターにインストールされている他のすべてのDr.Web製品をアンインストールします。

Dr.Web for UNIX Mail Servers以外の他のDr.Web製品がコンピューターにインストールされている状況でDr.Web for UNIX Mail Serversのみを削除するには、自動削除ツールを実行する代わりにコンポーネントのカスタムインストールとアンインストールの手順を使用します。

コマンドラインからDr.Web for UNIX Mail Serversをアンインストールする

アンインストールツールは、*<opt_dir>*/binディレクトリ（GNU/Linuxでは/opt/drweb.com/bin）にあるuninst.shスクリプトによって起動されます。Dr.Web for UNIX Mail Serversのアンインストール手順については、コマンドラインからアンインストールするのセクションで説明されています。

次のコマンドを実行することで、アンインストールツールをサイレントモードで起動することもできます。

```
# env DRWEB_NON_INTERACTIVE=yes /opt/drweb.com/bin/uninst.sh
```

この場合、アンインストールツールはサイレントモードで実行され、ユーザーインターフェースなしで動作します（コマンドラインモードのプログラムダイアログを含みます）。

アンインストールツールをサイレントモードで起動するにはroot権限が必要です。権限を昇格するには、suまたはsudoコマンドを使用できます。

古いバージョンのパッケージマネージャーを使用しているOS（ALT 8 SPなど）でユニバーサルパッケージをアンインストールする場合、次のようなメッセージが表示されることがあります。

```
/etc/init.d/drweb-configd: No such or directory
```

これらのメッセージは、システムの機能に影響を与えるものではありません。アンインストールは正しく行われています。

コマンドラインからアンインストールする

コマンドラインベースのアンインストールプログラムが起動すると、製品を削除するメッセージがコマンドラインに表示されます。

1. 削除を開始するには、「Do you want to continue?」リクエストに対して「`Yes`」または「`Y`」を入力します。アンインストーラを終了するには、「`No`」または「`N`」と入力します。この場合、Dr.Web製品の削除はキャンセルされます。
2. アンインストールを確定すると、インストールされているすべてのDr.Web製品の自動アンインストールが開始されます。この手順の間、削除のプロセスに関する情報が画面に表示され、アンインストールログに記録されます。
3. プロセスが完了すると、アンインストールプログラムは自動的に終了します。

リポジトリからインストールしたDr.Web for UNIX Mail Serversをアンインストールする

以下のOS（パッケージマネージャー）の手順を参照してください。

- Debian、Mint、Ubuntu（apt）
- ALT Linux、PCLinuxOS（apt-rpm）
- Mageia、OpenMandriva Lx（urpmi）
- Red Hat Enterprise Linux、Fedora、CentOS（yum、dnf）
- SUSE Linux（zypper）

以下に記載される、パッケージのアンインストールに使用されるコマンドはスーパーユーザー（root）権限で実行する必要があります。権限を昇格するには、`su`コマンド（カレントユーザーを変更する）または`sudo`コマンド（指定されたコマンドを別のユーザーの権限で実行する）を使用します。

Debian、Mint、Ubuntu（apt）

Dr.Web for UNIX Mail Serversのルートメタパッケージをまとめてアンインストールするには、次のコマンドを入力します。

```
# apt-get remove drweb-mail-servers
```

ルートのメタパッケージをすべての依存ファイルと一緒にアンインストールする必要がある場合は、次のように`--autoremove`オプションを使用します。

```
# apt-get remove drweb-mail-servers --autoremove
```

不要になったすべてのパッケージを自動的にアンインストールするには、次のコマンドを入力します。

```
# apt-get autoremove
```

アンインストールの特別な側面

1. 最初のコマンドのケースでは、`drweb-mail-servers`パッケージのみをアンインストールします。このパッケージの依存関係を解決するのに自動的にインストールされた可能性のある他のパッケージはシステムに残ります。
2. 2番目のコマンドのケースでは、`drweb-mail-servers`パッケージと、他のパッケージの依存関係を解決するために自動的にインストールされ、依存パッケージのアンインストールなどにより不要になったすべてのパッケージをアンインストールします。
3. 3番目のコマンドのケースでは、他のパッケージの依存関係を解決するために自動的にインストールされ、依存パッケージのアンインストールなどにより不要になったパッケージをすべてアンインストールします。
   このコマンドはDr.Web for UNIX Mail Serversのパッケージだけでなく、使用されていないすべてのパッケージをアンインストールします。

代わりのマネージャー（Synapticまたはaptitudeなどを使用してパッケージをアンインストールすることもできます。

ALT Linux、PCLinuxOS（apt-rpm）

この場合、Dr.Web for UNIX Mail Serversのアンインストールは、DebianおよびUbuntu上でのアンインストールと同じです（上記を参照）。

代わりのマネージャー（Synapticまたはaptitudeなどを使用してパッケージをアンインストールすることもできます。

ALT 8 SPでは、ユニバーサルパッケージのアンインストール時に次のようなメッセージが表示されることがあります。

```
/etc/init.d/drweb-configd: No such or directory
```

これらのメッセージは、システムの機能に影響を与えるものではありません。アンインストールは正しく行われています。

Mageia、OpenMandriva Lx（urpme）

Dr.Web for UNIX Mail Serversをアンインストールするには、次のコマンドを入力します。

```
# urpme drweb-mail-servers
```

不要になったすべてのパッケージを自動的にアンインストールするには、次のコマンドを入力します。

```
# urpme --auto-orphans drweb-mail-servers
```

アンインストールの特別な側面

1. 最初のコマンドのケースでは、`drweb-mail-servers`パッケージのみをアンインストールします。このパッケージの依存関係を解決するのに自動的にインストールされた可能性のある他のパッケージはシステムに残ります。
2. 2番目のコマンドのケースでは、`drweb-mail-servers`パッケージと、他のパッケージの依存関係を解決するために自動的にインストールされ、依存パッケージのアンインストールなどにより不要になったすべてのパッケージをアンインストールします。
このコマンドはDr.Web for UNIX Mail Serversのパッケージだけでなく、使用されていないすべてのパッケージをアンインストールします。

代わりのマネージャー（rpmdrakeなど）を使用してパッケージをアンインストールすることもできます。

Red Hat Enterprise Linux、Fedora、CentOS（yum、dnf）

インストールされているすべてのDr.Webパッケージをアンインストールするには、次のコマンドを入力します（一部のOSでは「`*`」記号を「`\*`」としてエスケープする必要があります）。

```
# yum remove drweb*
```

Fedoraのバージョン22以降では、マネージャー`yum`の代わりに`dnf`を使用することが推奨されます。例：

```
# dnf remove drweb*
```

アンインストールの特別な側面

これらのコマンドは、名前が「drweb」(Dr.Web製品名の標準的接頭辞)で始まるすべてのパッケージをアンインストールします。
これらのコマンドはDr.Web for UNIX Mail Serversのパッケージだけでなく、この接頭辞を持つパッケージをすべてアンインストールします。

代わりのマネージャー(PackageKitまたはYumexなど)を使用してパッケージをアンインストールすることもできます。

SUSE Linux(zypper)

Dr.Web for UNIX Mail Serversをアンインストールするには、次のコマンドを入力します。

```
# zypper remove drweb-mail-servers
```

インストールされているすべてのDr.Webパッケージをアンインストールするには、次のコマンドを入力します(一部のOSでは「*」記号を「*」としてエスケープする必要があります)。

```
# zypper remove drweb*
```

アンインストールの特別な側面

1. 最初のコマンドのケースでは、drweb-mail-serversパッケージのみをアンインストールします。このパッケージの依存関係を解決するのに自動的にインストールされた可能性のある他のパッケージはシステムに残ります。
2. 2番目のコマンドのケースでは、drweb-mail-serversパッケージと、他のパッケージの依存関係を解決するために自動的にインストールされ、依存パッケージのアンインストールなどにより不要になったすべてのパッケージをアンインストールします。

代わりのマネージャー(YaSTなど)を使用してパッケージをアンインストールすることもできます。

追加情報

Dr.Web for UNIX Mail Serversパッケージとファイル

パッケージ

Dr.Web for UNIX Mail Serversは以下のパッケージで構成されています。

パッケージ	コンテンツ
drweb-ase	Dr.Web Anti-Spam Engine(Dr.Web ASE)のコンポーネントファイル 一部のディストリビューションでは使用できません。

パッケージ	コンテンツ
drweb-bases	ウイルスデータベースファイル
drweb-boost	ブーストライブラリ
drweb-clamd	Dr.Web ClamDコンポーネントのファイル
drweb-cloudd	Dr.Web CloudDコンポーネントのファイル
drweb-common	メインの設定ファイル - drweb.ini、メインライブラリ、ドキュメント、Dr.Web for UNIX Mail Serversディレクトリの階層、製品設定とシステム環境に関する情報を収集するためのユーティリティ。 このパッケージのインストール中に、drwebという名前のユーザーとdrwebという名前のグループが作成されます。
drweb-configd	Dr.Web ConfigDのファイル
drweb-ctl	Dr.Web Ctlのファイル
drweb-documentation	HTMLフォーマットのDr.Web for UNIX製品ドキュメントファイル
drweb-dws	Webリソースカテゴリーのデータベースのファイル
drweb-engine	Dr.Web Virus-Finding Engineスキャンエンジンのファイル
drweb-esagent	Dr.Web ES Agentコンポーネントのファイル
drweb-filecheck	Dr.Web File Checkerコンポーネントのファイル
drweb-mail-servers-doc	PDFドキュメント
drweb-mail-servers	Dr.Web for UNIX Mail Serversのルートメタパッケージ
drweb-gated	SpIDer Gateコンポーネントのファイル
drweb-firewall	Dr.Web Firewall for Linuxコンポーネントのファイル
drweb-httpd	Dr.Web HTTPDコンポーネントと管理Webインターフェース（メタパッケージ）のファイル
drweb-httpd-bin	Dr.Web HTTPDコンポーネントのファイル
drweb-httpd-webconsole	管理Webインターフェースのファイル
drweb-icu	Unicodeサポートとインターナショナライゼーションのためのライブラリ
drweb-libs	メインライブラリファイル
drweb-lookupd	Dr.Web LookupDコンポーネントのファイル
drweb-lua	ネットワーク接続監視用に設計されたDr.Web for UNIX Mail Serversコンポーネントによって使用されるLuaインタプリタのファイル

パッケージ	コンテンツ
drweb-maild	Dr.Web MailDコンポーネントのファイル
drweb-netcheck	Dr.Web Network Checkerコンポーネントのファイル
drweb-openssl	OpenSSLライブラリ
drweb-protobuf	Google Protobufライブラリ
drweb-se	Dr.Web Scanning Engineコンポーネントのファイル
drweb-snmpd	Dr.Web SNMPDコンポーネントのファイル
drweb-update	Dr.Web Updaterコンポーネントのファイル
drweb-vaderetro	アンチスパムライブラリのファイル 一部のディストリビューションでは使用できません。
drweb-cgp-plugin	Dr.Web for UNIX Mail ServersとCommunigate Proを統合するためのプラグインのファイル 一部のディストリビューションでは使用できません。

セクションコンポーネントのカスタムインストールとアンインストールには、Dr.Web for UNIX Mail Serversの典型的なタスクに対する解決策を提供するカスタムインストール用の典型的なコンポーネントセットがあります。

ファイル

Dr.Web for UNIX Mail Serversのインストール後、その構成ファイルはファイルシステムの/opt、/etc、/varディレクトリに置かれます。

Dr.Web for UNIX Mail Serversディレクトリの構造:

ディレクトリ	コンテンツ
• GNU/Linuxの場合: /etc/init.d/ • FreeBSDの場合: /usr/local/etc/rc.d/	Dr.Web ConfigDデーモン用のdrweb-configdスクリプト
<etc_dir>	drweb.ini設定ファイルとdrweb32.keyキーファイル。また、次が含まれます。
certs/	– 使用中の証明書のファイル
<opt_dir>/	Dr.Web for UNIX Mail Serversのメインディレクトリ。次が含まれます。
bin/	– すべての製品コンポーネントの実行ファイル(Dr.Web Virus-Finding Engineを除く)

ディレクトリ	コンテンツ
include/	– 使用中のライブラリのヘッダーファイル
lib/	– 使用中のライブラリ
man/	– システムヘルプファイル：man
share/	– 以下を含む補助製品ファイル
cgp/	▫ Dr.Web for UNIX Mail ServersとCommunigate Proを統合するためのプラグインのファイル
doc/	▫ 製品ドキュメント（readmeファイル、使用許諾契約、パッケージがすでにインストールされている場合は管理者ガイド）
drweb-bases/	▫ Dr.Webのウイルスデータベースのファイル（インストール中に提供されたソースファイル）
scripts/	▫ 補助スクリプトファイル
<var_dir>/	以下を含む補助ファイルと一時ファイル：
bases/	– Dr.Webウイルスデータベースのファイル（更新バージョン）
cache/	– 更新のキャッシュ
drl/	– 使用中の更新サーバーのリスト
dws/	– Webリソースカテゴリーのデータベースのファイル
lib/	– ダイナミックリンクライブラリとしてのDr.Web Virus-Finding Engineスキャンエンジン（drweb32.dll）と、集中管理モードで作業するための設定、さらにDr.Web for UNIX Mail Serversディストリビューションに含まれている場合は、メールメッセージをスキャンしてスパムを検出するためのライブラリ
update/	– ダウンロード中に更新を一時的に保存するためのディレクトリ

ディレクトリで使用される表記規則の詳細については、はじめにを参照してください。

コンポーネントのカスタムインストールとアンインストール

このセクションの内容

- カスタムインストール用の一般的なコンポーネントキット
- Dr.Web for UNIX Mail Serversコンポーネントのインストールとアンインストール：
  - リポジトリからインストールする
  - ユニバーサルパッケージからインストールする

必要に応じて、該当するそれぞれのパッケージをインストール／アンインストールすることで、特定のDr.Web for UNIX Mail Serversコンポーネントのみをインストール／アンインストールできます。カスタムコンポーネントのインストールまたはアンインストールは、製品のインストールと同じ方法で実行します。

コンポーネントを再インストールするには、まず初めにそのコンポーネントをアンインストールし、その後再度インストールしてください。

カスタムインストール用の一般的なコンポーネントキット

リポジトリまたはユニバーサルパッケージからルートメタパッケージをインストールする代わりに、機能を制限してDr.Web for UNIX Mail Serversをインストールする必要がある場合は、必要な機能を提供するコンポーネントパッケージのみをインストールできます。依存関係を解決するために必要なパッケージは自動的にインストールされます。以下の表は、一般的なDr.Web for UNIX Mail Serversタスクを解決するために設計されたコンポーネントセットを示しています。**インストールするパッケージ**列には、特定のコンポーネントスイートを取得するためにインストールする必要があるパッケージのリストがあります。

カスタムコンポーネントキット	インストールするパッケージ	インストールされるコンポーネント
コンソールスキャンのための最小キット	• drweb-filecheck • drweb-se	• Dr.Web ConfigD • Dr.Web Ctl • Dr.Web File Checker • Dr.Web Scanning Engine • Dr.Web Updater • ウイルスデータベース
ClamAVのエミュレーションのためのスイート（clamd）	• drweb-clamd • drweb-se	• Dr.Web ClamD • Dr.Web ConfigD • Dr.Web Ctl • Dr.Web File Checker • Dr.Web Network Checker • Dr.Web Scanning Engine • Dr.Web Updater • ウイルスデータベース
MTAに接続可能なフィルターとしてメールをスキャンするためのスイート	• drweb-antispam • drweb-dws • drweb-maild • drweb-netcheck • drweb-se • drweb-vaderetro	• Dr.Web ASE**** • Dr.Web ConfigD • Dr.Web Ctl • Dr.Web MailD • Dr.Web Network Checker • Dr.Web Scanning Engine* • Dr.Web Updater*** • Dr.Web URL Checker • Webリソースカテゴリーのデータベース** • ウイルスデータベース* • スパムフィルター****

カスタムコンポーネントキット	インストールするパッケージ	インストールされるコンポーネント
メールメッセージのアンチウイルススキャンが必要ない場合は、drweb-netcheckおよびdrweb-seパッケージをインストールする必要はありません。 Dr.Web Network Checkerを介してスキャン用のデータを受信する別のサーバーでアンチウイルススキャンが実行される場合、drweb-seパッケージのインストールがスキップされる可能性があります。 望ましくないWebリソースのカテゴリーにURLが該当するかをチェックする必要がない場合は、drweb-dwsパッケージのインストールがスキップされる可能性があります。 メールメッセージのスパムをスキャンする必要ない場合は、drweb-antispamおよびdrweb-vaderetroパッケージのインストールがスキップされる可能性があります。		
SMTP、POP3、IMAPプロトコルの透過プロキシモードでメールをスキャンするためのスイート	• drweb-antispam • drweb-dws • drweb-firewall • drweb-gated • drweb-maild • drweb-netcheck • drweb-se • drweb-vaderetro	• Dr.Web ASE**** • Dr.Web ConfigD • Dr.Web Ctl • Dr.Web Firewall for Linux • Dr.Web MailD • Dr.Web Network Checker • Dr.Web Scanning Engine* • Dr.Web Updater*** • Dr.Web URL Checker • SpIDer Gate

カスタムコンポーネントキット	インストールするパッケージ	インストールされるコンポーネント
メールメッセージのアンチウイルススキャンが必要ない場合は、drweb-netcheckおよびdrweb-seパッケージをインストールする必要はありません。 Dr.Web Network Checkerを介してスキャン用のデータを受信する別のサーバーでアンチウイルススキャンが実行される場合、drweb-seパッケージのインストールがスキップされる可能性があります。 望ましくないWebリソースのカテゴリーにURLが該当するかをチェックする必要がない場合は、drweb-dwsパッケージのインストールがスキップされる可能性があります。 メールメッセージのスパムをスキャンする必要ない場合は、drweb-antispamおよびdrweb-vaderetroパッケージのインストールがスキップされる可能性があります。		• Webリソースカテゴリーのデータベース** • ウイルスデータベース* • スパムフィルター****

* drweb-seパッケージがインストールされていない場合、このコンポーネントはインストールされません。
** drweb-dwsパッケージがインストールされていない場合、このコンポーネントはインストールされません。
*** Dr.Web Updaterコンポーネントは、ウイルスデータベース、Webリソースカテゴリーデータベース、およびスパムフィルターがインストールされている場合にのみインストールされます。
**** スパムをスキャンするパッケージがインストールされていない場合、このコンポーネントはインストールされません。

リポジトリからインストールされたDr.Web for UNIX Mail Serversコンポーネントのインストールとアンインストール

Dr.Web for UNIX Mail Serversをリポジトリからインストールした場合、コンポーネントのカスタムインストール／アンインストールには、お使いのOSで使用されているパッケージマネージャーの各コマンドを使用します。以下はその例です。

1. CentOS上にインストールされているDr.Web for UNIX Mail ServersからDr.Web ClamD（`drweb-clamd`パッケージ）をアンインストールするには、次のコマンドを使用します。

```
# yum remove drweb-clamd
```

2. Ubuntu OSにインストールされているDr.Web for UNIX Mail ServersにDr.Web ClamD（`drweb-clamd`パッケージ）を追加でインストールするには、次のコマンドを使用します。

```
# apt-get install drweb-clamd
```

必要に応じて、お使いのOSで使用されているパッケージマネージャーのヘルプを参照してください。

ユニバーサルパッケージからインストールされたDr.Web for UNIX Mail Serversコンポーネントのインストールとアンインストール

Dr.Web for UNIX Mail Serversがユニバーサルパッケージからインストールされていて、コンポーネントのパッケージを追加でインストールまたは再インストールする場合、Dr.Web for UNIX Mail Serversのインストール元のインストールファイル（`.run`拡張子の付いたファイル）が必要です。このファイルを保存していない場合は、Doctor Web公式サイトからダウンロードしてください。

インストールファイルを解凍する

`.run`ファイルを実行する際は、以下のコマンドラインパラメータを指定することもできます。

`--noexec` - インストールプロセスを開始せずに、Dr.Web for UNIX Mail Serversのインストールファイルを解凍します。ファイルは`TMPDIR`環境変数で指定されたディレクトリに置かれます（通常は`/tmp`）。

`--keep` - インストール完了後にDr.Web for UNIX Mail Serversのインストールファイルとインストールログを自動的に削除しません。

`--target` *<directory>* - Dr.Web for UNIX Mail Serversのインストールファイルを、指定されたディレクトリ*<directory>*に解凍します。

.runファイルの起動時に指定できるコマンドラインパラメータの一覧を表示するには、以下のコマンドを入力します。

```
$ ./<file_name>.run --help
```

カスタムインストールでは、解凍されたインストールファイルを使う必要があります。それらのファイルが含まれたディレクトリがない場合は、最初に以下のコマンドを入力してインストールファイルを解凍します。

```
$ ./<file_name>.run --noexec --target <directory>
```

コマンドが実行された後、ディレクトリ*<directory>*内に、ネストされたディレクトリの名前*<file_name>*が現れます。

コンポーネントのカスタムインストール

RUNインストールファイルには、Dr.Web for UNIX Mail Serversのすべてのコンポーネントのパッケージ（RPMフォーマットで）とサポートファイルが含まれています。各コンポーネントのパッケージファイルは以下の構造を持っています。

```
<component_name>_<version>~linux_<platform>.rpm
```

*<version>*は製品リリースのバージョンと時間が含まれたストリングで、*<platform>*はDr.Web for UNIX Mail Serversが対象としているプラットフォームです。Dr.Web for UNIX Mail Serversのコンポーネントが含まれているパッケージの名前はすべて「drweb」プレフィックスで始まります。

パッケージマネージャーはインストールキットでのパッケージのインストール時に有効になります。カスタムインストールでは、サービススクリプト`installpkg.sh`を使用する必要があります。その際、まずインストールパッケージのコンテンツをディレクトリに解凍する必要があります。

パッケージをインストールするには、スーパーユーザー権限（*root*ユーザーの権限）が必要です。権限を昇格するには、`su`コマンドを使用してカレントユーザーを変更するか、または`sudo`コマンドを使用して、指定されたコマンドを別のユーザーの権限で実行します。

コンポーネントパッケージのインストールまたは再インストールを開始するには、解凍したインストールキットのあるディレクトリに移動し、コンソールから（またはグラフィカルモードのターミナルであるコンソールエミュレーターから）以下のコマンドを実行します。

```
# ./scripts/installpkg.sh <package_name>
```

例：

```
# ./scripts/installpkg.sh drweb-clamd
```

Dr.Web for UNIX Mail Servers全体のインストールを開始する必要がある場合は、以下のコマンドを使用して自動インストールスクリプトを実行します。

```
$ ./install.sh
```

その他、製品のルートメタパッケージを実行することで、すべてのDr.Web for UNIX Mail Serversパッケージをインストールできます（不足しているか、誤って削除してしまったコンポーネントをインストールするため）。

```
# ./scripts/installpkg.sh drweb-mail-servers
```

コンポーネントのカスタムアンインストール

お使いのOSがRPMフォーマットのパッケージを使用している場合、コンポーネントのカスタムアンインストールでは、OSのパッケージマネージャーの該当するアンインストールコマンドを使用します。

- Red Hat Enterprise LinuxとCentOSでは`yum remove` *<package_name>*コマンドを使用します。
- Fedoraでは`yum remove` *<package_name>*または`dnf remove` *<package_name>*コマンドを使用します。
- SUSE Linuxでは`zypper remove` *<package_name>*コマンドを使用します。
- MageiaとOpenMandriva Lxでは`urpme` *<package_name>*コマンドを使用します。
- Alt LinuxとPCLinuxOSでは`apt-get remove` *<package_name>*コマンドを使用します。

例（Red Hat Enterprise Linuxの場合）:

```
# yum remove drweb-clamd
```

お使いのOSがDEBパッケージを使用している場合（MSVS 3.0 OSの場合も）、またはOSにパッケージマネージャーがない場合（FreeBSD）、カスタムアンインストールでは、Dr.Web for UNIX Mail Serversのインストール中に自動的にインストールされるパッケージマネージャー`zypper`を使用する必要があります。これを行うには、*<opt_dir>*`/bin`（GNU/Linuxの場合は`/opt/drweb.com/bin`）ディレクトリに移動して、以下のコマンドを実行します。

```
# ./zypper remove <package_name>
```

例:

```
# ./zypper remove drweb-clamd
```

Dr.Web for UNIX Mail Serversをアンインストールする必要がある場合は、以下のコマンドを入力して自動削除スクリプトを実行します。

```
# ./uninst.sh
```

コンポーネントを再インストールするには、まずそのコンポーネントをアンインストールし、その後、インストールキットからのカスタムインストールまたはフルインストールを実行することで再度インストールします。

セキュリティサブシステムを設定する

OSに強化セキュリティサブシステムSELinuxが実装されている場合や、PARSECなどの強制アクセス制御システム（UNIXで使用されていた従来の任意モデルではないもの）が使用されている場合は、それらがデフォルト設定になっているとDr.Web for UNIX Mail Serversの動作に問題が生じます。そのような場合にDr.Web for UNIX Mail Serversを正常に動作させるためには、セキュリティサブシステムやDr.Web for UNIX Mail Serversの設定を変更する必要があります。

SELinuxのセキュリティポリシーの設定に関する以下の詳細を参照してください。

SELinuxのセキュリティポリシーを設定する

GNU/LinuxディストリビューションにSELinux（*Security-Enhanced Linux*）が含まれている場合は、インストール後にDr.Web for UNIX Mail Serversのサービスコンポーネント（スキャンエンジンなど）を正常に動作せるために、SELinuxのセキュリティポリシーを設定することが必要になる場合があります。

ユニバーサルパッケージを使用したインストールの問題

SELinuxが有効になっている場合、Dr.Web for UNIX Mail Serversコンポーネントを動作させる*drweb*ユーザーの作成がブロックされることがあり、インストールファイル（`.run`）からのインストールに失敗する場合があります。

*drweb*ユーザーを作成できないために、このファイルからのDr.Web for UNIX Mail Serversのインストールに失敗する場合は、`getenforce`コマンドを使用してSELinuxの動作モードを確認してください。このコマンドは現在のスキャンモードを出力します。

- `Permissive` - 保護は有効ですが、許可方式が使用されています。セキュリティポリシーに違反するアクションは拒否されませんが、そのアクションに関する情報はログに記録されます。
- `Enforced` - 保護は有効で、制御方式が使用されています。セキュリティポリシーに違反するアクションはブロックされ、そのアクションに関する情報はログに記録されます。
- `Disabled` - SELinuxはインストールされていますが、有効になっていません。

SELinuxが*Enforced*モードで動作している場合は、*Permissive*モードに変更してください。それには、以下のコマンドを使用します。

```
# setenforce 0
```

このコマンドはSELinuxの*Permissive*モードを一時的に（次の再起動まで）有効にします。

> `setenforce`コマンドで有効にした動作モードに関係なく、OSの再起動後、SELinuxは設定で指定された安全な動作モードに戻ります（SELinuxの設定ファイルは通常、`/etc/selinux`ディレクトリにあります）。

Dr.Web for UNIX Mail Serversが正常にインストールされた後、製品を起動させる前に*Enforced*モードを再度有効にしてください。それには、以下のコマンドを使用します。

```
# setenforce 1
```

Dr.Web for UNIX Mail Serversの動作に関する問題

SELinuxの実行中にいくつかのDr.Web for UNIX Mail Serversコンポーネント（`drweb-se`や`drweb-filecheck`など）が起動できないことがあります。これにより、オブジェクトのスキャンやファイルシステムのモニタリングができなくなります。これらのコンポーネントが起動できない場合は、`syslog`サービスによって管理されるシステムログ（通常このログは`/var/log/`ディレクトリにあります）に119および120のエラーメッセージが表示されます。

SELinuxセキュリティシステムによってアクセスが拒否された場合、そのようなイベントのログが記録されます。一般的に、システムでauditデーモンが使用されている場合、audit（監査）に関するログ

が/var/log/audit/audit.logファイルに保存されます。それ以外の場合、ブロックされた動作に関するメッセージが一般的なログファイル(/var/log/messagesまたは/var/log/syslog)に保存されます。

SELinuxにブロックされているために製品のスキャンコンポーネントが機能しない場合は、該当するコンポーネントに対して特別な*セキュリティポリシー*を設定する必要があります。

一部のGNU/Linuxディストリビューションには、以下のユーティリティが備わっていません。その場合、ユーティリティの追加パッケージをインストールする必要がある場合があります。

SELinuxのセキュリティポリシーを設定する

1. SELinuxのポリシーソースコードを記述したファイル(.teファイル)を新たに作成します。このファイルでは、記述されているポリシーモジュールに関連した制限を規定します。このポリシーソースコードは以下のいずれかの方法で作成できます。
   1) audit2allowユーティリティの使用は、最もシンプルな方法です。ユーティリティはシステムログファイル内のアクセス拒否に関するメッセージからpermissiveルールを生成します。自動でメッセージを検索するよう設定するか、手動でログファイルへのパスを指定できます。

この方法は、Dr.Web for UNIX Mail Serversのコンポーネントが SELinuxのセキュリティポリシーに違反していて、それらのイベントが監査ログファイルに記録されている場合にのみ使用できます。そうでない場合、そのようなイベントが起こるのを待つか、policygentoolユーティリティを使用して強制的にpermissiveポリシーを作成してください(下記参照)。

---

audit2allowユーティリティは、policycoreutils-pythonパッケージまたはpolicycoreutils-develパッケージ(Red Hat Enterprise Linuxの場合はCentOS、Fedora OSの場合はバージョンにより異なる)にあり、DebianおよびUbuntu OSの場合はpython-sepolgenパッケージにあります。

audit2allowの使用例:

```
# grep drweb-se.real /var/log/audit/audit.log | audit2allow -M drweb-se
```

この例では、drweb-seコンポーネントに対するアクセス拒否メッセージを見つけるために、audit2allowユーティリティが/var/log/audit/audit.logファイル内で検索を実行します。

作成されるファイルは、ポリシーソースファイルdrweb-se.teと、インストール可能なdrweb-se.ppポリシーモジュールの2つです。

システム監査ログ内でセキュリティ違反イベントが見つからなかった場合、ユーティリティはエラーメッセージを返します。

ほとんどの場合、audit2allowユーティリティで作成したポリシーファイルを変更する必要はありません。したがって、手順4のdrweb-se.ppポリシーモジュールのインストールに進むことを推奨します。

audit2allowユーティリティはsemoduleコマンドの呼び出しを出力します。この出力をコマンドラインにコピーして実行すると、手順4が完了します。Dr.Web for UNIX Mail Serversコンポーネント用に自動的に生成されたセキュリティポリシーを変更する場合にのみ、手順2に進みます。

2) policygentoolユーティリティを使用する。そのためには、異なる方法で処理するコンポーネントの名前とその実行ファイルへのフルパスを指定します。

Red Hat Enterprise LinuxとCentOS向けのselinux-policyパッケージに含まれているpolicygentoolユーティリティは正常に機能しない場合があります。その場合はaudit2allowユーティリティを使用してください。

policygentoolを使用したポリシー作成の例:

- drweb-seコンポーネントの場合:

```
# policygentool drweb-se /opt/drweb.com/bin/drweb-se.real
```

- drweb-filecheckコンポーネントの場合:

```
# policygentool drweb-filecheck /opt/drweb.com/bin/drweb-filecheck.real
```

ドメインを作成するための一般的なプロパティをいくつか指定するように求められます。その後、ポリシーを決定する以下の3つのファイルが(コンポーネントごとに)作成されます。

<module_name>.te、*<module_name>*.fc、*<module_name>*.if

2. 必要に応じて、生成されたポリシーソースファイル*<module_name>.te*を編集し、その後、checkmoduleユーティリティを使用して、ローカルポリシーのこのソースファイルをバイナリ形式に変換(.modファイル)します。

コマンドを正常に実行するには、システムにcheckpolicyパッケージがインストールされている必要があります。

使用例:

```
# checkmodule -M -m -o drweb-se.mod drweb-se.te
```

3. semodule_packageユーティリティを使用して、インストール用のポリシーモジュールを作成します(.ppファイル)。

例:

```
# semodule_package -o drweb-se.pp -m drweb-se.mod
```

4. 作成されたポリシーモジュールをインストールするには、semoduleユーティリティを使用します。

例:

```
# semodule -i drweb-se.pp
```

SELinuxの動作と設定に関する詳細は、お使いのUNIXディストリビューションのマニュアルを参照してください。

開始する

1. インストールされたDr.Web for UNIX Mail Serversの使用を開始するために、キーファイルを入手してインストールした上で有効化する必要があります。
2. Dr.Web for UNIX Mail Serversの動作確認のためにさらにスキャンすることをお勧めします。
3. Dr.Web for UNIX Mail Serversを、*Milter*、*Spamd*、または*Rspamd*拡張機能を介して動作する、あるいはSMTP統合モードで動作する外部フィルターとして接続して、使用するメールサーバーと統合します。また、Dr.Web for UNIX Mail ServersをDr.Web vxCubeと統合して、メール添付ファイルをスキャンすることもできます。
4. SMTPプロキシモードでDr.Web for UNIX Mail Serversを使用する場合は、まず、トランジットMTA機能を実行するメールサーバーをインストールして設定します（インストールされていない場合）。
5. GNU/Linuxベースのシステムの場合、メールサーバーやMUAに対して透過型のプロキシモードを設定できます。このモードでは、Dr.Web for UNIX Mail Serversとメールサーバーとの実際の統合を実行する必要はありません。プロトコルSMTP、POP3、IMAPとの透過型の統合がサポートされています。
6. どのコンポーネントが実行されているかを確認し、サーバーの保護に必要な場合には、デフォルトで無効になっているコンポーネントを追加で有効にします（ディストリビューションによって異なりますが、Dr.Web ClamDまたはDr.Web SNMPDコンポーネントなど）。

   追加コンポーネントを有効にすること以外に、デフォルトの設定を調整するなど、他のアクションを実行する必要がある場合があります。

   インストール済みおよび実行中のコンポーネントとその設定のリストを表示するには、次のいずれかを使用します。

   - コマンドラインベース管理ツールのDr.Web Ctl。`drweb-ctl appinfo`、`drweb-ctl cfshow`、および`drweb-ctl cfset`コマンドを使用します。
   - Dr.Web for UNIX Mail Serversの管理用ウェブインターフェース（初期設定では、ウェブブラウザから`https://127.0.0.1:4443/`にアクセスすると利用できます）。

Dr.Web for UNIX Mail Serversは、メールメッセージに対して次のアクションのみを実行します。

- 管理者が設定した基準に準拠していることと、スパムの兆候をスキャンしていることを確認するための*メールメッセージチェック*（DNSxLブラックリストをチェックする設定がされている場合、そのリストの送信者ドメインのチェックも実施）
- 悪意のあるWebサイト、または望ましくないカテゴリーに属するWebサイトへの*リンクの検索*
- *悪意のある添付ファイルの検出*

スキャン用のメールメッセージの受信に使用されたプロトコルとメールメッセージを送信した側（MTA/MDAまたはMUA）がスキャン用の転送済みメールメッセージの変更をサポートしている場合、Dr.Web for UNIX Mail Serversは標準的なアクション「無視」と「拒否」に加え、事前に定義された再圧縮テンプレートの1つに基づいてメールメッセージを*再圧縮*できます（再圧縮中は、すべての脅威がメールに添付された保護アーカイブに移動され、脅威や望ましくない内容に関する通知がメールの本文に追加されます）。その上、メールのヘッダーを追加、修正する基本機能がサポートされています。

*その他*のすべてのアクション（管理者への通知の送信、添付ファイルの完全な削除、名前変更など）が必要な場合は、保護されたメールサーバー（MTA/MDA）を*介して実装する必要があります。必要に応じて、他社の開発者から対応する処理用に設計された一連の特定のフィルタープラグインを入手し、接続することにより、保護されたメールサーバー経由でそれらを実装する必要があります。*

スパムの兆候に対するメールメッセージのスキャン機能は、Dr.Web for UNIX Mail Serversのディストリビューションによっては利用できない可能性があります。

製品の登録と有効化

このセクションの内容

- ライセンスを購入・登録する
- デモライセンスを取得する
- キーファイルのインストール
- 2回目以降の登録

ライセンスを購入・登録する

ライセンスを購入すると、製品コンポーネントとウイルスデータベースの更新がDoctor Web更新サーバーから定期的にダウンロードされます。さらに、購入した製品をインストールまたは使用するときに問題が発生した場合は、Doctor Webまたはそのパートナーが提供するテクニカルサポートサービスを利用できます。

Dr.Web製品の購入や製品のシリアル番号の入手は、パートナーまたはオンラインストアから可能です。ライセンスオプションの詳細については、Doctor Web公式Webサイト（https://license.drweb.com/license_manager）にアクセスしてください。

ライセンス登録は、ユーザーがDr.Web for UNIX Mail Serversの正規ユーザーであることを証明し、ウイルスデータベースの定期的な更新を含むアンチウイルスの機能を有効にするために必要です。インストールが完了したら、製品を登録してライセンスを有効化することを推奨します。購入したライセンスは、Doctor Webの公式Webサイト（https://products.drweb.com/register/v4）で有効化できます。

有効化の際には、購入したライセンスのシリアル番号を入力する必要があります。シリアル番号はDr.Web for UNIX Mail Serversと一緒に提供されるか、オンラインでライセンスを購入または更新した際にメールで提供されます。

デモライセンスを取得する

ご利用のDr.Web for UNIX Mail Serversの試用期間は、Doctor Webの公式Webサイト（https://download.drweb.com/demoreq/biz/v2）で確認できます。製品を選択して登録フォームに記入すると、Dr.Web for UNIX Mail Serversを有効化するためのシリアル番号またはキーファイルが記載されたメールが届きます。

同じコンピューターでの2回目以降の試用期間は、一定の期間が経過した後に利用できます。

Dr.Web Ctl（`drweb-ctl`）コマンドラインツールのライセンスコマンドを使用すると、登録されたライセンスのシリアル番号のデモキーファイルまたはライセンスされたキーファイルを自動的に取得できます。

キーファイルのインストール

キーファイルは、Dr.Web for UNIX Mail Serversの購入したライセンスまたは有効化した試用期間に対応する、ローカルコンピューター上に保存される特別なファイルです。このファイルには提供されたライセンスまたは試用期間に関する情報が含まれ、また、このファイルに応じて使用権限が規定されます。

Dr.Web for UNIX Mail Serversの動作中、キーファイルはデフォルトの *<etc_dir>*ディレクトリ（GNU/Linuxの場合は `/etc/opt/drweb.com`）に`drweb32.key`という名前で置かれている必要があります。

Dr.Web for UNIX Mail Serversのコンポーネントは、キーファイルが使用可能かつ有効であるかどうかを定期的に確認します。編集されることを防ぐため、キーファイルはデジタル署名されています。キーファイルを編集すると無効になります。誤って無効にしてしまうことを防ぐため、キーファイルをテキストエディターで開かないことをお勧めします。

有効なキーファイル（正規またはデモライセンス）が見つからない場合、またはライセンスの有効期限が切れている場合、有効なキーファイルがインストールされるまでアンチウイルスコンポーネントの動作はブロックされます。

ライセンスキーファイルは有効期限が切れるまで保管しておき、Dr.Web for UNIX Mail Serversの再インストールや、別のコンピューターへのインストールにはそのライセンスキーファイルを使用することを推奨します。この場合、登録時に指定したものと同じ製品シリアル番号と顧客データを使用する必要があります。

メールメッセージでは、Dr.Webキーファイルは通常、zipアーカイブに圧縮されて転送されます。Dr.Web for UNIX Mail Serversのアクティベーション用のキーファイルを含むアーカイブは、通常は`drweb32.zip`または`agent.zip`という名前です。メッセージに*複数の*アーカイブが含まれている場合は、`agent.zip`アーカイブを使用する必要があります。

キーファイルをインストールする前に、アーカイブを適宜解凍し、そこからキーファイルを抽出して、使用可能な任意のディレクトリ（たとえば、ホームディレクトリやUSBフラッシュドライブ）に保存してください。

製品の有効なライセンスに対応するキーファイルをお持ちの場合（キーファイルをメールで受け取った場合、またはDr.Web for UNIX Mail Serversを別のサーバー上で使用する場合など）、そのキーファイルへのパスを指定することでDr.Web for UNIX Mail Serversを有効化できます。その場合は、次の操作を行います。

1. アーカイブの場合はキーファイルを解凍します。
2. 次のいずれかを実行してください:
   - キーファイルを*<etc_dir>*ディレクトリにコピーし、必要に応じてファイル名を`drweb32.key`に変更します。
   - Dr.Web for UNIX Mail Servers設定ファイルで、`KeyPath`パラメータ値にキーファイルパスを指定します。
3. 次のコマンドを入力して、Dr.Web for UNIX Mail Serversの設定をリロードします。

   ```
   # drweb-ctl reload
   ```

   すべての変更が適用されます。

また、次のコマンドを使用することもできます。

```
# drweb-ctl cfset Root.KeyPath <path to the key file>
```

この場合、Dr.Web for UNIX Mail Serversの再起動は不要です。キーファイルは*<etc_dir>*ディレクトリにコピーされず、元の場所に残ります。

<opt_dir>、*<etc_dir>*、*<var_dir>*の表記規則の詳細は、はじめにを参照してください。

キーファイルが*<etc_dir>*ディレクトリにコピーされない場合は、ユーザーはファイルが破損や削除から保護されていることを確認する必要があります。キーファイルがシステムから誤って削除される可能性があるため、この方法は推奨されません（たとえば、キーファイルが存在するディレクトリが定期的にクリーンアップされる場合）。キーファイルを紛失した場合は、サポートを要求して新しいキーファイルを取得できますが、要求できる回数には制限があります。

2回目以降の登録

キーファイルを紛失したが、既存のライセンスの有効期限が切れていない場合は、前回の登録時に指定した個人データを入力して、再度登録する必要があります。別のメールアドレスを使用できます。この場合、ライセンスキーファイルは新しく指定されたアドレスに送信されます。

キーファイルのリクエスト回数には制限があります。その数を超えてリクエストが送信された場合、キーファイルは配信されません。紛失したキーファイルを入手するには、Doctor Webテクニカルサポートに連絡して問題の詳細を説明し、シリアル番号の登録時に入力した個人データを伝えてください。ライセンスキーファイルはメールで送信されます。

キーファイルがメールで送信されたら、手動でインストールする必要があります。

製品の動作確認

EICAR (European Institute for Computer Anti-Virus Research) テストを行うと、ウイルスをシグネチャで検出するアンチウイルスプログラムの動作を確認できます。このテストは、新しくインストールしたアンチウイルスツールのウイルス検出の動作を、コンピューターを危険にさらすことなくテストできるように特別に設計されています。

EICAR は実際にはウイルスではありませんが、多くのアンチウイルスプログラムによってウイルスとして処理されるようにできています。この「ウイルス」を検出すると、Dr.Webのアンチウイルス製品は「EICAR Test File (NOT a Virus!)」という表示を出します。その他のアンチウイルスツールも同じようにユーザーに警告します。EICARテストファイルはMS DOS/MS Windows向けの68バイトのCOMファイルで、実行すると、ターミナル画面またはコンソールエミュレーターに次のラインを出力します。

```
EICAR-STANDARD-ANTIVIRUS-TEST-FILE!
```

EICARテストファイルは、次の文字列のみを含んでいます。

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

上記の文字列でファイルを作成すると、「ウイルス」として認識されるテストファイルができあがります。

Dr.Web for UNIX Mail Serversが正常に動作していれば、テストファイルはスキャンの種類に関係なくファイルシステムのスキャン中に検出され、検出された脅威について「EICAR Test File (NOT a Virus!)」と通知されます。

EICARテストを使用したDr.Web for UNIX Mail Serversの動作の確認を、コマンドラインから実行する場合のコマンドの例:

```
$ tail <opt_dir>/share/doc/drweb-se/readme.eicar | grep X5O > testfile &&
drweb-ctl rawscan testfile && rm testfile
```

このコマンドは、*<opt_dir>*`/share/doc/drweb-se/readme.eicar`ファイル(Dr.Web for UNIX Mail Serversに付属)からEICARテストファイルの本文を表す文字列を抽出し、それを現在のディレクトリに作成された`testfile`という名前のファイルに書き込みます。その後、このファイルをスキャンしてから削除します。

上記のテストを行うには、カレントディレクトリへの書き込みアクセスが必要です。また、ディレクトリに`testfile`という名前のファイルが含まれていないことを確認してください(必要に応じて、コマンド内でファイル名を変更してください)。

<opt_dir>、*<etc_dir>*、*<var_dir>*の表記規則の詳細は、はじめにを参照してください。

テストウイルスが検出されると、以下のメッセージが表示されます。

```
<path to the current directory>/testfile - infected with EICAR Test File (NOT a
Virus!)
```

テスト中にエラーが発生した場合は、既知のエラーの説明を参照してください(付録F. 既知のエラーを参照)。

フィルターとしてのMTAとの統合

このセクションの内容

- Dr.Web MailDを設定する
- MTAを設定する
- 一部のMTAの設定例

この統合方法では、メールメッセージスキャン用の外部フィルターとしてDr.Web MailDをメールサーバーに直接接続することになります。*Milter*、*Spamd*、*またはRspamd*インターフェースを使用するあらゆるメールサーバー（Exim、Sendmail、Postfixなど）がサポートされています。Postfixメールサーバーを使用する場合、コンポーネントはSMTPモードでも動作します（SMTPモードの動作原理の詳細については、Dr.Web vxCubeとの統合を参照）。

Dr.Web MailDのパラメータを設定する

1. Milter、Spamd、Rspamd経由の接続

お使いのメールサーバーにDr.Web MailDを統合するには、設定ファイルの`[MailD]`で次のパラメータの値を編集します。

- **Dr.Web MailDのMTAとの統合パラメータ。**まず、使用するインターフェース（*Milter*、*Spamd*、*Rspamd*）を決定し、MTA接続のパラメータと、選択したインターフェースで受信するメールスキャンのパラメータを指定する必要があります。特定のインターフェースを介してMTAとの統合を制御するDr.Web MailDのすべてのパラメータには、その名前にそれぞれのプレフィックス（`Milter*`、`Spamd*`、`Rspamd*`）が付いています。
  1. *<interface>*`Socket`は、対応するインターフェースを介してMTAからスキャン済みメールメッセージを取得するためにDr.Web MailDによって使用されるUNIXまたはネットワークソケットです。
  2. メールメッセージスキャンの長さとリソース強度を制限するパラメータ（`ScanTimeout`、`HeuristicAnalysis`、`PackerMaxLevel`、`ArchiveMaxLevel`、`MailMaxLevel`、`ContainerMaxLevel`、`MaxCompressionRatio`）。詳細な調整が必要ない場合は、これらのパラメータの値を変更しないでください。
  3. メールフィルタリングルールの詳細な設定については、デフォルトのメールスキャン用Luaスクリプトを編集してください。
- **メールメッセージスキャン中のDr.Web MailDの一般的な動作パラメータ。**`TemplateContacts`パラメータでは、脅威やスパムが検出された場合のメッセージ送信先となるメールサーバー管理者のアドレスを指定します。`ReportLanguages`パラメータでは、サービスレポートの生成時に使用する言語を指定します。`RepackPassword`パラメータの値で、再圧縮(repack)時にメールメッセージに追加される、脅威を含む保護されたアーカイブ用のパスワードの生成方法を指定します。これらのパラメータの詳細については、該当するセクションを参照してください。

すべての設定を調整したら、Dr.Web for UNIX Mail Serversを次のコマンドで再起動します。

```
# drweb-ctl reload
```

設定デーモンDr.Web ConfigDは、次のコマンドでも再起動できます。

```
# service drweb-configd restart
```

*Milter*を介してMTAとやり取りする場合、Luaスクリプトはメッセージに適用されるアクションを返します。

*Spamd*を介して対話する場合、Luaスクリプトは、`SPAM`または`THREAT`という単語を含む`report`変数を返します。その結果はMTAの設定（Eximの場合はACLなど）に従って処理され、メッセージが拒否されるか、送信者が警告を受けることになります。

*Rspamd*を介して対話する場合、Luaスクリプトは、`ADD_HEADER`または`REJECT`のいずれかの値を持つアクション変数を返します。その結果はMTAの設定（Eximの場合はACLなど）に従って処理され、ヘッダーが結果に追加される（さらに、受信者に送信される）か、または拒否されます。

2. SMTPモードの接続

お使いのメールサーバーにDr.Web MailDを統合するには、設定ファイルの`[MailD]`セクションで次のパラメータの値を編集します。

1. `SmtpSocket` - MTAからチェック対象のメールメッセージを取得するためにDr.Web MailDによって使用されるUNIXまたはネットワークソケット。
2. `SmtpSenderRelay` - 処理済みのメールメッセージを送信するためにDr.Web MailDによって使用されるUNIXまたはネットワークソケット。
3. 追加パラメータ（タイムアウト、利用可能な通信プロトコル、デバッグログへの出力）。パラメータには`Smtp`プレフィックスが付きます。詳細な調整が必要ない場合は、これらのパラメータの値を変更しないでください。
4. メールフィルタリングルールの詳細な設定については、デフォルトのメールスキャン用Luaスクリプトを編集してください。

すべての設定を調整したら、Dr.Web for UNIX Mail Serversを次のコマンドで再起動します。

```
# drweb-ctl reload
```

設定デーモンDr.Web ConfigDは、次のコマンドでも再起動できます。

```
# service drweb-configd restart
```

MTAのパラメータを設定する

1. Milter、Spamd、Rspamd経由の接続

MTAとDr.Web MailDの間のインタラクションを有効にするには、メールサーバーの設定を編集します。

1. メールメッセージをスキャンするときにMTAとDr.Web MailDのインタラクションに使用されるインターフェース（*Milter*、*Spamd*、*Rspamd*）を指定します。
2. 選択したインターフェースを介してMTAをDr.Web MailDに接続するためのパラメータを指定します（使用するソケットは、Dr.Web MailD設定にある、対応するインターフェースの*<interface>*`Socket`パラメータで指定されているものと一致する必要があります）。
3. インタラクションインターフェースを介してメールのスキャン結果を受信した後に、MTAが実行するアクションを指定します。

設定の変更後、MTAを再起動します。

2. SMTPモードの接続

MTAとDr.Web MailDの間のインタラクションを有効にするには、メールサーバーの設定を編集します。

1. Dr.Web MailDにメールメッセージを送信するためのクライアントのパラメータを設定します。
2. Dr.Web MailDによってチェックされたメッセージを送信するためのMTAのパラメータを設定します。
3. 指定したソケットを介したDr.Web MailDへのMTA接続のパラメータを設定します。

よく使われるMTAの設定例

以下はMTA Postfix、Sendmail、Exim、CommuniGate Proの典型的な設定の例です。SMTPモードに加えて、*Milter*、*Spamd*、*Rspamd*のインターフェースを介し、メールメッセージの外部フィルターとしてDr.Web MailDを接続します。

以下の例では、*<MailD socket>*、*<MailD IP address>*、および*<MailD port>*の値を、Dr.Web MailD設定の*<interface>*`Socket`パラメータで指定されているDr.Web MailDソケットのパラメータに置き換える必要があります。ここで*<interface>*は、選択したMTAインターフェースに対応するパラメータの名前にある、または`SmtpSocket`パラメータ(SMTPモードの場合)にあるプレフィックスです。SMTPモードではさらに、*<Postfix socket>*の値を、Dr.Web MailD設定の`SmtpSocketRelay`パラメータで指定したPostfixソケットの値に置き換える必要があります。

たとえば、Dr.Web MailDがネットワークソケットを使用して*Milter*インターフェース経由でMTAと統合され、MTAとDr.Web MailDの両方がローカルホスト上で動作し、Dr.Web MailDがポート12345で*Milter*経由の接続をリッスンする場合、この値は、Dr.Web for UNIX Mail Servers設定ファイルの`[MailD]`セクションで`MilterSocket`パラメータとして指定する必要があります。MTA設定では、*<MailD socket>*変数の値に`127.0.0.1:12345`を、*<MailD IP address>*変数のアドレスに`127.0.0.1`を、*<MailD port>*変数の値に12345をそれぞれ指定する必要があります。

場合によっては、Dr.Web MailDとの接続のソケットアドレスに、プレフィックス*<type>*と、MTA設定で使用されるアドレスのタイプ(`inet`、`inet6`、`unix`)を追加する必要があります。

1. Postfix

- *Milter*:

  MTA設定ファイル`main.cf`に以下の行を追加します。

```
smtpd_milters = <type>:<MailD socket>
milter_content_timeout = 300s
milter_default_action = tempfail
milter_protocol = 2
```

`smtpd_milters`および`milter_protocol`パラメータのみが必須です。それ以外のパラメータは省略できます。

- SMTPモードの場合:
  - MTA設定ファイルmain.cfに以下の行を追加します。

```
# Client parameters for sending email messages to MailD to be checked

scan      unix  -       -       n       -       10      smtp

        -o smtp_send_xforward_command=yes

        -o disable_mime_output_conversion=yes

        -o smtp_generic_maps=


# MTA parameters for sending messages checked by Dr.Web MailD

<Postfix socket> inet  n       -       n       -       10      smtpd

    -o content_filter=

    -o
receive_override_options=no_unknown_recipient_checks,no_header_body_check
s,no_milters

    -o smtpd_helo_restrictions=

    -o smtpd_client_restrictions=

    -o smtpd_sender_restrictions=

    -o smtpd_relay_restrictions=

    -o smtpd_recipient_restrictions=permit_mynetworks,reject

    -o mynetworks=127.0.0.0/8

    -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

  - MTA設定ファイルmain.cfに以下の行を追加します。

```
content_filter = scan:<MailD socket>

receive_override_options = no_address_mappings
```

Dr.Web MailDとPostfixのホストが異なる場合、mynetworksとauthorized_xforward_hostsの値はDr.Web MailDホストの値に置き換える必要があります。

2. Sendmail

- *Milter*:

  MTAプロトタイプ設定ファイル`sendmail.mc`に以下の行を追加します。

  ```
  INPUT_MAIL_FILTER(`drweb-milter', `S=<MailD socket>, F=T')
  ```

  サンプルファイル`sendmail.mc`を変更したら、次のいずれかのコマンドで、このファイルをアクティブな設定ファイル`sendmail.cf`に変更します。

  ```
  make -C /etc/mail
  sendmailconfig
  m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
  ```

  上記のコマンドはすべて、Sendmailの設定ファイルが`/etc/mail`ディレクトリにあることを前提としています。

3. Exim

- *Spamd*:

  MTA設定ファイル`exim.conf`に以下の行を追加します。

  ```
  spamd_address = <MailD socket>
  acl_smtp_data = acl_check_data

  acl_check_data:
   warn spam  = nobody:true
   add_header = X-Spam_score: $spam_score\n\
   X-Spam_score_int: $spam_score_int\n\
   X-Spam_bar: $spam_bar\n\
   X-Spam_report: $spam_report

   deny message = This message scored $spam_score spam points.
   spam = nobody:true
   condition = ${if >{$spam_score_int}{10000}{true}{false}}

   accept message = This message scored $spam_score spam points.
   spam = nobody:true
   condition = ${if >{$spam_score_int}{1000}{true}{false}}
   remove_header = Subject
   add_header = Subject: [SPAM] $rh_Subject
  ```

- *Rspamd*:

  MTA設定ファイル`exim.conf`に以下の行を追加します。

  ```
  spamd_address = <MailD socket> variant=rspamd
  acl_smtp_data = acl_check_data

  acl_check_data:
   # Add header fields
  ```

```
 warn spam  = nobody:true
 add_header = X-Spam_score: $spam_score\n\
  X-Spam_score_int: $spam_score_int\n\
  X-Spam_bar: $spam_bar\n\
  X-Spam_report: $spam_report

 # Reject the message with proper description if Rspamd filter tells to
do so
 deny spam = nobody:true
 message = ${extract{2}{:}{$spam_action}}
 condition = ${if eq{${extract{1}{:}{$spam_action}}}{reject}}

 # Accept the message otherwise
 accept
```

Dr.Web MailDとの統合はバージョン4.6（またはそれ以降）のEximで、オプション `WITH_CONTENT_SCAN=yes`でコンパイルした場合に利用できます。

4. CommuniGate Pro

- *Rspamd*:

1. CommunigateProと連動するには特別なモジュールが必要です。これはDr.Webリポジトリに含まれており、標準のパッケージマネージャーを介してインストールできます。

   Debian、Ubuntu、Mintの場合：

   ```
   # apt-get install drweb-cgp-plugin
   ```

   Red Hat Enterprise LinuxとCentOSの場合：

   ```
   # yum install drweb-cgp-plugin
   ```

   Fedoraの場合：

   ```
   # dnf install drweb-cgp-plugin
   ```

2. モジュールは`/opt/drweb.com/share/cgp/`にインストールされます。このディレクトリに移動し、ファイル`CgpDrweb_AS_AV.py`を次のように実行可能にします。

   ```
   # cd /opt/drweb.com/share/cgp/
   # chmod +x CgpDrweb_AS_AV.py
   ```

3. Webインターフェースを使用してCommuniGate Proを設定します。
   - **Setting** → **General** → **Helpers**に移動します。次のようにモジュールをCommuniGate Proに接続します。
     - **Content Filtering**セクションで新しいフィルターを設定し、**Enabled**に切り替えます。
     - フィルター名を指定します（例：`CgpDrweb_AS_AV`）。
     - **Program Path**パラメータで、スクリプトファイルへのパス（GNU/Linuxの場合は`/opt/drweb.com/share/cgp/CgpDrweb_AS_AV.py`）と、スクリプトを起動するための

オプション（`-r`はソケットのアドレスとポート、`-u`または`--rspamd-unix-socket`はUNIXソケットへのパス、`--debug`はデバッグモードで起動）を指定します。

Helpersを使用できるように、**Expert**または**Advanced**ビューモードを有効化します（CommunigatePro設定の**Preferences** → **Interface**）。
可能なすべてのオプションのリストを表示するには、次のコマンドを実行します。

```
# ./CgpDrweb_AS_AV.py --help
```

- 変更内容を保存します。

- **Setting** → **Mail** → **Rules**に移動します。
  - 新しいルール名（`CgpDrweb_AS_AV`など）を指定し、**Add Rule**をクリックします。
  - **Highest**ルール設定を選択し、変更内容を保存します。
  - ルール名の右側にある**Edit**をクリックします。
  - **Data**ドロップダウンメニューで**Message Size**を選択し、**Operation**フィールドで**less than**を選択し、**Parameter**フィールドで*40960000*を指定します。
  - **Action**フィールドで**ExternalFilter**を選択し、**Parameter**で、以前に作成したフィルターの名前（この場合は**CgpDrweb_AS_AV**）を選択します。
  - 変更内容を保存します。
- 脅威検出レスポンスルールを追加し、その名前（`Drweb_threats`など）を指定して、**Add Rule**をクリックします。
  - ルールのプライオリティー`5`を指定し、変更内容を保存します。
  - ルールの右側にある**Edit**をクリックします。ルールの条件を2回追加します。
    - **Data**ドロップダウンリストで**Header Field**を選択し、**Operation**フィールドで**is**を選択し、**Parameter**フィールドで`X-Spam-Action: reject`を指定します。
    - **Data**ドロップダウンリストで**Header Field**を選択し、**Operation**フィールドで**is**を選択し、**Parameter**フィールドで`X-Spam-Symbol-1: threat*`を指定します。

    Actionフィールドで**Reject with**を選択します。**Parameter**でテキスト（`The message contains threat(s)`など）を指定します。

    変更内容を保存します。
- 脅威検出レスポンスルールを追加し、その名前（`Drweb_spam`など）を指定して、**Add Rule**をクリックします。
  - ルールのプライオリティー`5`を指定し、変更内容を保存します。
  - ルールの右側にある**Edit**をクリックします。ルールの条件を追加します。
    - **Data**ドロップダウンメニューで**Header Field**を選択します。
    - **Operation**フィールドで**is**を選択します。
    - **Parameter**フィールドで`X-Spam-Action: tag`を選択します。
  - **Action**フィールドで**Tag Subject**を選択し、**Parameter**でヘッダープレフィックス（`[SPAM]`など）を指定します。
  - 変更内容を保存します。

4. 以下のファイルのコンテンツをコピーして、`hook.lua`として保存します。

```
--Message scanning procedure,
--transmitted using the Rspamd protocol

function rspamd_hook(ctx)

--Message scanning to detect threats
if ctx.message.has_threat() then
  return {
    score = 900,
    threshold = 100,
    action = "reject",
    symbols = {
      {
        name = "threat",
        score = 900
      }
    }
  }
end

--Message scanning to detect spam
if ctx.message.spam.score > 100 then
  return {
    score = ctx.message.spam.score,
    threshold = 100,
    action = "tag",
    symbols = {
      {
        name = "spam",
        score = ctx.message.spam.score
      }
    }
  }
end

return {
   score = ctx.message.spam.score,
   threshold = 100,
   action = "accept",
   symbols = {
    {
      name = "The message is clean",
      score = 0
    }
   }
  }
end
```

5. 次のコマンドを実行します。

```
# drweb-ctl cfset MailD.RspamdHttpSocket <socket address>:<port>
# drweb-ctl cfset MailD.RspamdHook <path to hook>
```

フックのコード(hook.lua)を編集する場合は、変更後に次のようにDr.Web ConfigDを再起動する必要があります。

```
# service drweb-configd restart
```

Dr.Web vxCubeとの統合

このセクションの内容

- Dr.Web vxCubeについて
- Dr.Web vxCubeとの統合の基本原則
- SMTPモード
- BCCモード

Dr.Web MailDは外部メールチェックフィルターとしてメールサーバーに接続し、メール添付ファイルを解析するDr.Web vxCubeと統合することができます。

Dr.Web vxCubeについて

Dr.Web vxCubeは、潜在的に悪意のあるファイルを解析し、選択された環境での動作に関する詳細なレポートを作成し、検出された脅威を駆除するツールを生成するWebサービスです。Dr.Web vxCubeは解析のためにハードウェア仮想化を使用します。これにより、Dr.Web vxCubeは高速で動作し、解析対象のファイルから見えないようになります。

Dr.Web vxCubeは高度なマルウェア検出機能を備えており、シグネチャデータベースにまだ存在せず、他の解析方法では見過ごされる可能性のある最新の脅威を特定できます。

Dr.Web vxCubeとの統合の基本原則

Dr.Web vxCubeを使用する場合、Dr.Web MailDコンポーネントは解析のために、vxCube APIを通じてDr.Web vxCubeにメール添付ファイルを送信します。Dr.Web vxCubeはすべての添付ファイルをスキャンし、ファイルのステータス（クリーン、疑わしい、または危険）の判定をレポートにしてDr.Web MailDに返します。動作モードに応じて、Luaスクリプトがこのレポートを使用して各メールを処理するか、レポートがシステム管理者にメール送信されます。

Dr.Web vxCubeをメールスキャンに使用するには、有効なDr.Web vxCubeのライセンスが必要です。

Dr.Web MailDとDr.Web vxCubeはSMTPモードとBCCモードのどちらかで統合できます。これらのモードではメールサーバーへの接続に*Milter*、*Spamd*、および*Rspamd*インターフェースを使用しません。

SMTPモード

SMTPモードではメールのトラフィックが能動的にフィルタリングされ、メールに適用されるルールを設定できます。

SMTPモードはPostfixメールサーバーでサポートされています。

SMTPモードで運用する場合、Dr.Web vxCubeとの統合はオプションです。Dr.Web vxCubeを使用しない場合、Dr.Web MailDがすべてのチェックを行います。

動作

1. クライアントはメールを送信するときに、ローカルエリアネットワークの外部からアクセス可能なMTAソケットに接続します。
2. MTAは受信メッセージをチェック待ちのメッセージのキューに保存し、その保存ステータスをユーザーに通知し、スキャンのためにメッセージをDr.Web MailDに転送します。
3. Dr.Web MailDはメールメッセージを整理するために、それらをディスクに保存し、Dr.Web MailDコンポーネントのDr.Web Mail Quarantineを通じてそのメタデータをSQLiteデータベースに保存します。
4. Dr.Web MailDは、明示的に設定されたLuaスクリプトを使用してメッセージをチェックします。

   メッセージのチェックはDr.Web MailDまたはDr.Web vxCubeを使用して行えます。それらの製品がDr.Web vxCubeと統合されている場合、Dr.Web MailDは、選択されている処理手順に応じて、メッセージ全体をDr.Web vxCubeに転送して添付ファイルを抽出することも、その設定で指定されているタイプの添付ファイルを独自に抽出してDr.Web vxCubeに転送することもできます。いずれの方法でも、Dr.Web vxCubeは添付ファイルを受信すると、悪意のあるコードがないか各ファイルを解析し、添付ファイルのセキュリティステータスに関する判定を行い、その判定結果をDr.Web MailDに送信します。
5. Dr.Web MailDは、メッセージ処理手順（*hook*）を含むLuaスクリプトを使用して、メッセージに適用する適切なアクション（pass、reject、return error to senderなど）を決定します。

   通過したチェック済みのメッセージは、プロセスで変更することもできます。たとえば、件名を付けたり、既存の件名を編集したりできます。添付ファイルに脅威が発見された場合、その添付ファイルはアーカイブされます。
6. Dr.Web MailDはメッセージをMTAキューに返します。そのメッセージは、チェック済みのメッセージ専用に予約され、ローカルネットワークの外部からはアクセスできないソケットに転送されます。
7. MTAはチェック済みメールのキューにメッセージを保存し、保存ステータスをDr.Web MailDに通知し、受信者へのメッセージの配信を試みます。配信に成功した場合、メッセージはMTAのキューから削除されます。失敗した場合は、一定時間後に配信が再試行されます。

MTAを設定する

外部メールフィルターとしてDr.Web MailDをSMTPモードで接続するMTA Postfixの設定例については、フィルターとしてのMTAとの統合のセクションを参照してください。

Dr.Web MailDを設定する

Dr.Web vxCubeに加えて、Dr.Web MailDをSMTPモードのメールサーバーとも統合するには、設定ファイルの多くのパラメータ、すなわちDr.Web MailD設定（`[MailD]`セクション）およびDr.Web ConfigD設定（`[Root]`セクション）のパラメータが正しく設定されていることを確認する必要があります。

SMTPモードでのMTAとの統合を制御するDr.Web MailDの主要パラメータには、すべてその名前にプレフィックス`Smtp`が付きます。

Dr.Web vxCubeとの統合を行わず本製品をSMTPモードで動作させるためには、`[MailD]`セクションで次のパラメータを設定する必要があります。

- `SmtpSocket` - Dr.Web MailDがMTAからチェック済みのメールメッセージを受信するために使用するソケット。UNIXまたはネットワークソケットを使用できます。
- `SmtpSenderRelay` - Dr.Web MailDがチェック済みのメールメッセージを転送するために使用するMTAソケット。UNIXまたはネットワークソケットを使用できます。

必要に応じてオプションのパラメータを設定することもできます。SMTPモードに影響するパラメータには`Smtp`プレフィックスが付いています。

SMTPモードで動作するDr.Web MailDのメール処理ルールは、`SmtpHook`パラメータによって決定されます。その値として入力されたデフォルトのスクリプトを使用することも、編集することもできます。Luaスクリプトによるメール処理の詳細については、Luaでのメール処理を参照してください。

Dr.Web vxCubeと統合した本製品をSMTPモードで動作させるためには、`[Root]`セクションで次のパラメータも設定する必要があります。

- `UseVxcube=Yes` - MTAに接続された外部フィルターとして、メール添付ファイルの解析にDr.Web vxCubeを使用します。
- `VxcubeApiAddress` - Dr.Web vxCube APIサーバーが稼働しているホストのドメイン名（FQDN）またはIPアドレス。
- `VxcubeApiKey` - Dr.Web vxCubeのAPIキー。

必要に応じてオプションのパラメータを設定することもできます。Dr.Web vxCubeとの統合に影響するパラメータには`Vxcube`プレフィックスが付いています。

BCCモード

BCCモードはメールのトラフィックセキュリティを受動的にモニタリングするもので、潜在的に悪意のあるファイルから受信者を能動的に保護するものではありません。

BCCモードは、*ブラインドカーボンコピー*（BCC）の送信をサポートするすべてのMTAでサポートされています。

このモードは、特にDr.Web vxCubeとの統合に使用されます。このモードでは、メールのチェック（およびフィルタリング）はDr.Web vxCubeによって実行されます。

動作

1. クライアントはメールを送信するときに、ローカルエリアネットワークの外部からアクセス可能なMTAソケットに接続します。
2. MTAは元のメッセージを受信者に送信し、そのブラインドカーボンコピーを作成して、Dr.Web MailD設定で指定されている、固有のドメインを持つ内部メールアドレスに転送します。
3. ローカルDNSサーバーは、システム管理者が作成したMXレコードに従って、LANの外部からアクセスできないDr.Web MailDのリスニングソケットに対応するIPアドレスにメッセージのコピーを送信します。
4. Dr.Web MailDはメールメッセージをチェック待ちのキューとチェック済みメールのキューに整理するために、それらをディスクに保存し、Dr.Web Mail Quarantineコンポーネントを通じてそのメタデータをSQLiteデータベースに保存します。
5. Dr.Web MailDは設定に基づいて、メール添付ファイルに含まれる特定のタイプのファイルを識別し、解析のためにDr.Web vxCubeに転送します。
6. Dr.Web vxCubeは、悪意のあるコードが含まれていないか各ファイルを解析し、添付ファイルのセキュリティステータスについて判定を行い、各添付ファイルの解析に関するレポートをDr.Web MailDに送信します。
7. Dr.Web MailDは、メッセージに添付されているすべてのファイルに関するレポートを1つの共通レポートに集約します。
8. 少なくとも1つのファイルが「疑わしい」または「危険」と判定された場合、Dr.Web MailDはDr.Web MailD設定で指定されているメールアドレスで、システム管理者に集計したレポートを転送します。

MTAを設定する

BCCモードでMTAとDr.Web MailDのインタラクションを確保するには、一意のドメインを持つメールアドレスにメールメッセージのブラインドカーボンコピーが転送されるように、MTAの設定を変更する必要があります。

設定の変更後、MTAを再起動します。

Dr.Web MailDにブラインドカーボンコピーを転送するには、MTAで指定したメールアドレスのドメインのMXレコードをローカルDNSサーバーに追加します。このレコードでは、Dr.Web MailDのリスニングソケットが指定されている必要があります（共通設定ファイルの`[MailD]`セクションの`BccSocket`パラメータ）。

Dr.Web MailDを設定する

Dr.Web vxCubeに加えて、Dr.Web MailDをBCCモードのメールサーバーとも統合するには、設定ファイルの多くのパラメータ、すなわちDr.Web MailD設定（`[MailD]`セクション）およびDr.Web ConfigD設定（`[Root]`セクション）のパラメータが正しく設定されていることを確認する必要があります。

BCCモードでのMTAとの統合を制御するDr.Web MailDの主要パラメータには、すべてその名前に`Bcc`プレフィックスが付いています。

次のパラメータに値を設定する必要があります。

- `[Root]`セクション:
  - `UseVxcube=Yes` - MTAに接続された外部フィルターとして、メール添付ファイルの解析にDr.Web vxCubeを使用します。
  - `VxcubeApiAddress` - Dr.Web vxCube APIサーバーが稼働しているホストのドメイン名（FQDN）またはIPアドレス。
  - `VxcubeApiKey` - Dr.Web vxCubeのAPIキー。
- `[MailD]`セクション:
  - `BccSocket` - Dr.Web MailDがMTAからチェック済みのメールメッセージを受信するために使用するソケット。UNIXまたはネットワークソケットを使用できます。
  - `BccReporterAddress` - Dr.Web MailDがチェック済み添付ファイルに関するレポートの送信に使用する送信元アドレス。
  - `BccReporterPassword` - Dr.Web MailDがチェック済み添付ファイルに関するレポートの送信に使用するメールアカウントのパスワード。
  - `BccReportRecipientAddress` - Dr.Web MailDがチェック済み添付ファイルに関するレポートの送信に使用する送信先アドレス。
  - `BccSmtpServer` - メッセージの送信に使用されるMTAのアドレス。ドメイン、IPアドレス、またはUNIXソケットを指定できます。

必要に応じてオプションのパラメータを設定することもできます。BCCモードに影響するパラメータにはプレフィックス`Bcc`、Dr.Web vxCubeとの統合に影響するパラメータはプレフィックス`Vxcube`が付いています。

SMTPプロキシモードでDr.Web for UNIX Mail Serversを使用する

このセクションの内容

- メールサーバーのスキャンパラメータを設定する
- Dr.Web MailDの設定を行う
- PostfixのSMTPプロキシ設定例

この統合方法では、SMTPプロトコル経由でメールメッセージを中継するメールサーバー（Exim、Sendmail、Postfixなど）をインストールし、およびDr.Web MailDがメールメッセージスキャン用の外部フィルターとしてこのメールサーバーに接続します。*Milter*、*Spamd*、および*Rspamd*インターフェースをサポートするすべてのメールサーバーを統合できます。

メールサーバーのスキャンパラメータを設定する

SMTPプロキシを実現するには、メールサーバーがメールメッセージを受信し、*Milter*、*Spamd*、または*Rspamd*インターフェースを介してメールスキャン用の外部フィルターとして接続されたDr.Web MailDでメールメッセージをスキャンし、指定されたルーティングルールに従ってメールメッセージの配信チェーンの最後または次の中間MTAにメールメッセージを送信するように設定する必要があります。

Milter、*Spamd*、または*Rspamd*インターフェースを介してDr.Web MailDをメールスキャン用の外部フィルターとして接続するために必要なMTAパラメータは、「フィルターとしてのMTAとの統合」のセクションに記載されています。

メールメッセージの送受信のルーティング設定は、インストールされているメールサーバーによって異なります。次の例は、Postfixメールサーバーの場合の設定です。

Dr.Web MailDの設定を行う

Dr.Web MailDとメールサーバーを統合するには、設定ファイル内のDr.Web MailDの設定セクション（`[MailD]`セクション）にあるパラメータの値を確認し、必要に応じて変更する必要があります。このような設定の例は、「フィルターとしてのMTAとの統合」のセクションにあります。

PostfixのSMTPプロキシ設定例

以下に示す例では、次のことを想定しています。

- Postfixがドメイン`example1.org`および`example2.com`からメールボックスに送信されたメールメッセージを受信する（メールメッセージのルーティングテーブルが`/etc/postfix/transport`ファイルで指定されている）。
- ネストされた脅威やスパムに関するメッセージのスキャンは、Dr.Web MailDによって*Milter*インターフェースを介して実行される。
- Dr.Web MailDはホスト`10.20.30.40`でポート`1234`をリッスンする。

1. `main.cf`設定ファイルの内容：

```
#Domains, for which the mail message scanning and transmission will be
performed
```

```
#email messages.
relay_domains = example1.org, example2.com

#Settings for connecting to an external Milter filter that performs
#message scan for viruses and spam.
smtpd_milters = inet:10.20.30.40:1234
milter_protocol = 2

Transport table (mail routing settings).
transport_maps = hash:/etc/postfix/transport
```

2. transportファイルの内容:

```
#String format:
#<transfer domain> <connection type>:<MTA address>:<listening port
number>

#All incoming and outgoing mail for the domain "example1.org"
#will be transmitted after scanning to MTA, located
#at the host "relay.example1.org" (on the default port for
#SMTP protocol)
example1.org    smtp:relay.example1.org

#All incoming and outgoing mail for the domain "example2.com"
#will be transmitted after scanning to MTA, located
#at the host with the IP address 2.2.2.2 to port 10025
example2.com    smtp:2.2.2.2:10025
```

透過プロキシモードでDr.Web for UNIX Mail Serversを使用する

このセクションの内容

- Dr.Web MailDのパラメータを設定する
- 透過プロキシのパラメータを設定する
- スキャン設定

このオプションはGNU/Linux OSの製品ディストリビューションでのみ使用できます。

Milter、*Spamd*または*Rspamd*インターフェースによって、またはClamAVプロトコルによってDr.Web for UNIX Mail Serversとお使いのメールサーバーを統合できない場合は、Dr.Web Firewall for Linuxを使って保護できます。その場合は、Dr.Web for UNIX Mail Serversをインストールしたサーバーに届くすべてのデータがSpIDer Gateネットワーク接続モニター(透過プロキシモード)によってチェックされるように、Dr.Web Firewall for Linuxを設定する必要があります。

Dr.Web MailDのパラメータを設定する

Dr.Web for UNIX Mail Serversを設定するには、設定ファイルの[MailD]セクションで次のパラメータの値を編集します。

- TemplateContactsとReportLanguagesのパラメータを使用して、脅威やスパムを含むメールメッセージを再圧縮する際のメール生成パラメータを指定します。
- TemplateContactsパラメータには、脅威やスパムが検出された場合のメッセージ送信先となるメールサーバー管理者のアドレスを指定します。
- RepackPasswordパラメータの値で、再圧縮(repack)時にメールメッセージに追加される、脅威を含む保護されたアーカイブ用のパスワードの生成方法を指定します。

透過プロキシのパラメータを設定する

透過プロキシモードを設定するには、設定ファイルの[LinuxFirewall]セクションにある値を次のように変更します。

パラメータ	必要な値
InspectSmtp	• SMTP経由で転送されるデータを監視する必要がある場合はOn(*MUAとMTA間またはMTAとMTA間のデータ転送*) • SMTP経由で転送されるデータを監視する必要がない場合はOff
InspectPop3	• POP3経由で転送されるデータを監視する必要がある場合はOn(*MUAとMDA間のデータ転送*) • POP3経由で転送されるデータを監視する必要がない場合はOff
InspectImap	• IMAP経由で転送されるデータを監視する必要がある場合はOn(*MUAとMDA間のデータ転送*) • IMAP経由で転送されるデータを監視する必要がない場合はOff
AutoconfigureIptables	Yes
AutoconfigureRouting	Yes
LocalDeliveryMark	Auto
ClientPacketsMark	Auto
ServerPacketsMark	Auto
TproxyListenAddress	127.0.0.1:0 Dr.Web Firewall for Linuxの操作に特別なIPアドレスまたはポートを使用する場合は、ここで指定します
OutputDivertEnable	• 送信接続(MTAによって発信された接続など、現在のホストで開始された接続)を監視する必要がある場合はYes

パラメータ	必要な値
	• 送信接続を監視する必要がない場合はNo
OutputDivertNfqueueNumber	Auto
OutputDivertConnectTransparently	No
InputDivertEnable	• 受信接続(リモートホストで開始される接続で、そのサーバー側は現在のホストで動作するMTAなどのアプリケーション)を監視する必要がある場合はYes • 受信接続を監視する必要がない場合はNo
InputDivertNfqueueNumber	Auto
InputDivertConnectTransparently	Yes

Dr.Web Firewall for Linuxの設定を表示および変更するには、次の方法を使用します。

- コマンドラインベース管理ツールDr.Web Ctl(drweb-ctl cfshowおよびdrweb-ctl cfsetコマンドを使用します)。

- Dr.Web for UNIX Mail Serversの管理用Webインターフェース(デフォルトでは、Webブラウザからhttps://127.0.0.1:4443/にアクセスすると利用できます)。

SSL/TLSの安全な接続を使用するメール配信チャネルにDr.Web for UNIX Mail Serversを統合する

1. 次のコマンドを実行して対応するパラメータの値を指定することで、SSL/TLS経由で送信されるトラフィックのスキャンを有効にします。

   ```
   # drweb-ctl cfset LinuxFirewall.UnwrapSsl Yes
   ```

   スキャンルールが自動的に更新されるように、drweb-ctlツールのcfsetコマンドまたは管理Webインターフェースを使用することを推奨します。スキャンルールはこのパラメータに依存します。
2. 証明書をエクスポートします。この証明書はSSL/TLS接続のためにDr.Web for UNIX Mail Serversによって使用されます。

   ```
   $ drweb-ctl certificate > <cert_name>.pem
   ```

3. 取得した証明書を、信頼できる証明書のシステムリストに追加し、それをメールクライアントおよびサーバー用の信頼できる証明書として指定します。詳細は、付録E. SSL証明書を生成するのセクションを参照してください。

スキャンパラメータを設定する

設定ファイルの[LinuxFirewall]セクションで、次のパラメータの値を設定します。

1. メールメッセージスキャンの長さとリソース強度を制限するパラメータ(ScanTimeout、HeuristicAnalysis、PackerMaxLevel、ArchiveMaxLevel、MailMaxLevel、

ContainerMaxLevel、MaxCompressionRatio）。詳細な調整が必要ない場合は、これらのパラメータの値を変更しないでください。

2. メールメッセージ内のリンクやファイルをスキャンするための設定を指定するBlock*パラメータ。
3. 受信したメールメッセージをスキャンできない場合に、Dr.Web MailDが取るべきアクションを指定するBlockUnchecked。このパラメータがYesに設定されている場合、メッセージは拒否されます。

フィルタリングルールの詳細な設定については、Luaプロシージャまたは RuleSetルールを編集してください。

すべての設定を調整したら、Dr.Web for UNIX Mail Serversを次のコマンドで再起動します。

```
# drweb-ctl reload
```

設定デーモンDr.Web ConfigDは、次のコマンドでも再起動できます。

```
# service drweb-configd restart
```

簡単な説明

このセクションの内容

- メールサーバーの操作:
  - Milter、Spamd、またはRspamdを介してDr.Web for UNIX Mail ServersをフィルターとしてMTAに接続する方法
  - Dr.Web for UNIX Mail ServersをアンチウイルスフィルターClamdとしてMTAに接続する方法
  - SMTPプロキシモード用に製品を設定する方法
  - MTAの透過プロキシモードを設定する方法
- Dr.Web for UNIX Mail Serversの一般的な動作:
  - Dr.Web for UNIX Mail Serversを再起動する方法
  - 集中管理サーバーに接続する方法
  - 集中管理サーバーから切断する方法
  - Dr.Web for UNIX Mail Serversを有効化する方法
  - Dr.Web for UNIX Mail Serversをアップグレードする方法
  - Dr.Web for UNIX Mail Serversコンポーネントを追加または削除する方法
  - Dr.Web for UNIX Mail Serversコンポーネントの動作を管理する方法
  - Dr.Web for UNIX Mail Serversのログを表示する方法

Milter、Spamd、またはRspamdを介してDr.Web for UNIX Mail ServersをフィルターとしてMTAに接続する方法

フィルターとしてのMTAとの統合セクションの指示に従ってください。

Dr.Web for UNIX Mail ServersをアンチウイルスフィルターClamdとしてMTAに接続する方法

外部アプリケーションとの統合セクションの指示に従ってください。

この場合、メールスキャン(スパムの兆候のスキャンを含む)用に設計された特別なコンポーネントDr.Web MailDは使用されません。MTAによって送信されたメールメッセージは、アンチウイルスによってのみスキャンされます。脅威が検出された場合、メッセージ処理はメールサーバーによって直接実行されます。

SMTPプロキシモード用にDr.Web for UNIX Mail Serversを設定する方法

SMTPプロキシモードでDr.Web for UNIX Mail Serversを使用するセクションの指示に従ってください。

MTAの透過プロキシモードを設定する方法

透過プロキシモードでDr.Web for UNIX Mail Serversを使用するセクションの指示に従ってください。

Dr.Web for UNIX Mail Serversを再起動する方法

Dr.Web for UNIX Mail Serversがすでに実行されているときに再起動するには、Dr.Web ConfigD設定デーモンを管理するスクリプトを使用することもできます。デーモンを起動、停止、または再起動すると、それぞれDr.Web for UNIX Mail Serversが起動、停止、または再起動されます。

Dr.Web ConfigDの動作を制御するシェルスクリプトは、標準のOSディレクトリ(GNU/Linuxの場合は`/etc/init.d/`、FreeBSDの場合は`/usr/local/etc/rc.d/`)にあります。スクリプトの名前は`drweb-configd`です。次のパラメータがあります。

パラメータ	説明
`start`	実行されていない場合は、Dr.Web ConfigDを起動します。Dr.Web ConfigDが起動すると、Dr.Web ConfigDはDr.Web for UNIX Mail Serversに必要なすべてのコンポーネントを起動します。
`stop`	実行されている場合は、Dr.Web ConfigDをシャットダウンします。Dr.Web ConfigDがシャットダウンすると、Dr.Web ConfigDはDr.Web for UNIX Mail Serversのすべてのコンポーネントもシャットダウンします。
`restart`	Dr.Web ConfigDを再起動(シャットダウンしてから起動)します。Dr.Web ConfigDはシャットダウンしてから、Dr.Web for UNIX Mail Serversのすべてのコンポーネントを起動します。Dr.Web ConfigDが実行されていない場合、このパラメータには起動と同じ効果があります。
`condrestart`	実行されている場合にのみ、Dr.Web ConfigDを再起動します。
`reload`	コンポーネントが実行されている場合は、HUPシグナルをDr.Web ConfigDに送信します。Dr.Web ConfigDはこのシグナルをDr.Web for UNIX Mail Serversのすべてのコンポーネントに転送します。このパラメータは、すべてのコンポーネントに設定を再度読み込ませるために使用されます。
`status`	Dr.Web ConfigDの現在の状態をコンソールに出力します。

たとえば、GNU/Linux OSでDr.Web for UNIX Mail Serversを再起動(実行されていない場合は起動)するには、次のコマンドを使用します。

```
# /etc/init.d/drweb-configd restart
```

集中管理サーバーに接続する方法

1. 集中管理サーバーのアドレスとその証明書のファイルをアンチウイルスネットワーク管理者から入手します。ワークステーションのIDとパスワードや、メイングループと課金プラングループのIDなど、追加パラメータが必要になる場合もあります。

2. Dr.Web for UNIX Mail Serversで提供されるDr.Web Ctlコマンドラインツールの`esconnect`コマンドを使用します。

   接続するには、サーバーの証明書ファイルへのパスを指定して、`--Certificate`オプションを使用する必要があります。`--Login`と`--Password`パラメータを使用することで、ホストのID（集中管理サーバー上での表記は、端末の識別子）と集中管理サーバーの認証用パスワードも入力できます。この場合、サーバーへの接続は、正しいIDとパスワードのペアを指定した場合にのみ確立されます。パラメータが指定されない場合、サーバーへの接続は、（サーバーの設定に応じて、自動的またはアンチウイルスネットワークの管理者によって）サーバーで承認されている場合にのみ確立されます。

   さらに、`--Newbie`オプション（新しいユーザーとして接続する）を使用することもできます。このモードがサーバーで許可されている場合、この接続が承認されると、サーバーは自動的に一意のIDとパスワードのペアを生成します。これはその後、このエージェントがサーバーに接続する際に使用されます。

   

   このモードでは、すでにこのホストの別アカウントがサーバーに存在している場合でも、集中管理サーバーはそのホストの新しいアカウントを生成します。

   Dr.Web for UNIX Mail Serversに集中管理サーバーへの接続を指示するコマンドの標準的な例は次のとおりです。

   ```
   # drweb-ctl esconnect <server address> --Certificate <path to the certificate file>
   ```

集中管理サーバーへの接続を確立すると、サーバーに設定されている権限とDr.Web ES Agentコンポーネントの`MobileMode`設定パラメータの値に応じて、Dr.Web for UNIX Mail Serversは集中管理モードまたはモバイルモードで動作します。無条件にモバイルモードを使用できるようにするには、パラメータの値を`On`に設定します。集中管理モードで動作させるには、パラメータの値を`Off`に設定します。

集中管理サーバーに接続されているDr.Web for UNIX Mail Serversにモバイルモードへの切り替えを指示するコマンドの標準的な例は次のとおりです。

```
# drweb-ctl cfset ESAgent.MobileMode On
```

使用する集中管理サーバーがモバイルモードをサポートしていない、または許可していない場合、`MobileMode`パラメータを調整してもDr.Web for UNIX Mail Serversの動作をモバイルモードに切り替えることはできません。

集中管理サーバーから切断する方法

Dr.Web for UNIX Mail Serversを集中管理サーバーから切断してその動作をスタンドアロンモードに切り替えるには、Dr.Web for UNIX Mail Serversで提供されるDr.Web Ctlコマンドラインツールの`esdisconnect`コマンドを使用します。

```
# drweb-ctl esdisconnect
```

Dr.Web for UNIX Mail Serversをスタンドアロンモードで使用するには、有効なライセンスキーファイルが必要です。それ以外の場合は、動作がスタンドアロンモードに切り替えられた後、Dr.Web for UNIX Mail Serversのアンチウイルス機能が*ブロック*されます。

Dr.Web for UNIX Mail Serversを有効化する方法

1. Doctor Web公式サイトhttps://products.drweb.com/register/v4から登録を実施します。
2. 登録時に指定したメールアドレスに、有効なライセンスキーファイルを含むアーカイブが送信されます（登録後にこのアーカイブをWebサイトから直接ダウンロードすることもできます）。
3. キーファイルのインストール手順を実行します。

Dr.Web for UNIX Mail Serversをアップグレードする方法

コンポーネントのバージョンを更新するか、新しいバージョンにアップグレードしてください。

アップグレード中に、現在のDr.Web for UNIX Mail Serversバージョンを削除するように求められることがあります。

Dr.Web for UNIX Mail Serversコンポーネントを追加または削除する方法

コンポーネントのカスタムインストールとアンインストールの手順に従います。

コンポーネントをインストール／アンインストールする場合、依存関係を解消するために他のDr.Web for UNIX Mail Serversコンポーネントを追加でインストールまたはアンインストールすることがあります。

コンポーネント動作を管理する方法

Dr.Web for UNIX Mail Serversコンポーネントのステータスを表示したり、それらの動作を管理したりするには、次のものを使用できます。

- コマンドラインベース管理ツールDr.Web Ctl（`drweb-ctl appinfo`、`drweb-ctl cfshow`、および`drweb-ctl cfset`コマンドを使用します。使用可能な管理コマンドのリストを表示するには、`drweb-ctl --help`コマンドを使用します）。
- Dr.Web for UNIX Mail Serversの管理用Webインターフェース（デフォルトでは、Webブラウザから`https://127.0.0.1:4443/`にアクセスすると利用できます）。

Dr.Web for UNIX Mail Serversのログを表示する方法

デフォルト設定に従って、すべてのDr.Web for UNIX Mail Serversコンポーネントの一般ログは`syslog`ファイルに表示されます（システムコンポーネント`syslog`によってメッセージをログに記録するためのファイルはシステムによって異なり、ディレクトリ`/var/log`にあります）。一般ログ設定は、設定ファイルの`[Root]`セクション（`Log`パラメータと`DefaultLogLevel`パラメータ）で定義されます。設定セクションの各コンポーネントには、`Log`パラメータと`LogLevel`パラメータがあります。それらのパラメータでログの保存場所と、コンポーネントがログに出力するメッセージのロギングレベルを設定します。

また、`drweb-ctl log`コマンドを使用することもできます。

ロギング設定を変更するには、コマンドライン管理ツールDr.Web CtlとDr.Web for UNIX Mail Servers管理Webインターフェース（インストールされている場合）を使用してください。

- エラーを特定するために、すべてのコンポーネントのログの出力先を別のファイルに設定し、デバッグ情報がログに出力されるようにすることを推奨します。そのためには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.Log <path to log file>
# drweb-ctl cfset Root.DefaultLogLevel DEBUG
```

- デフォルトのロギング方法とログレベルに戻すには、次のコマンドを実行します。

```
# drweb-ctl cfset Root.Log -r
# drweb-ctl cfset Root.DefaultLogLevel -r
```

Dr.Web for UNIX Mail Serversコンポーネント

このセクションでは、Dr.Web for UNIX Mail Serversのコンポーネントについて説明します。各コンポーネントの機能、動作原理、設定ファイルに保存されているパラメータに関する情報を確認できます。

Dr.Web ConfigD

Dr.Web ConfigD設定管理デーモンは、Dr.Web for UNIX Mail Serversの主要な制御コンポーネントです。すべてのDr.Web for UNIX Mail Serversコンポーネントの設定を一元的な環境に保存し、すべてのコンポーネントの動作を管理し、コンポーネント間で信頼できるデータのやり取りを整理します。

Dr.Web ConfigD設定管理デーモンは、次の機能を実行します。

- 設定に応じて、Dr.Web for UNIX Mail Serversのコンポーネントを起動、停止する
- 障害が発生した場合にコンポーネントを自動的に再起動する
- 他のコンポーネントのリクエストに応じてコンポーネントを起動する
- 設定の変更時にコンポーネントに通知する
- 設定パラメータを一元管理するためのインターフェースを提供する
- 使用中のライセンスファイルの情報をコンポーネントに伝える
- コンポーネントのライセンス情報を受け入れる
- 専用コンポーネントから新しいライセンス情報を受け取る
- ライセンス情報の変更時に実行中のコンポーネントに通知する

動作原理

Dr.Web ConfigDコンポーネントは常にroot権限で実行されます。また、他のDr.Web for UNIX Mail Serversコンポーネントを起動し、事前に開いていたソケットを介してそれらのコンポーネントと連携します。設定管理デーモンは、Dr.Web for UNIX Mail Serversのその他のコンポーネントから、情報ソケット（すべてのコンポーネントがアクセス可能）および管理ソケット（root権限で実行されるコンポーネントのみがアクセス可能）を介して接続を受け入れることができます。また、ファイルから、またはDr.Web ES Agentを介して集中管理サーバーから設定とライセンスに関する情報を読み込みます。さらに、設定パラメータの正しいデフォルト値を設定します。いずれかのコンポーネントが起動するか、SIGHUPシグナルを受信するまでに、すべてのDr.Web for UNIX Mail Serversコンポーネントの包括的で一貫性のある設定パラメーター式がDr.Web ConfigDに設定されます。

SIGHUPシグナルを受信すると、Dr.Web ConfigDは設定パラメータとライセンス情報を再度読み込みます。必要に応じて、デーモンはすべてのコンポーネント通知を送信して、設定を再度読み込むように指示します。

SIGTERMシグナルを受信すると、Dr.Web ConfigDはすべてのコンポーネントをシャットダウンした後、その動作を終了します。また、Dr.Web ConfigDはシャットダウン後、コンポーネントのすべての一時ファイルを削除します。

コンポーネントの連携の原則

1. 起動時に、すべてのコンポーネントはDr.Web ConfigDから設定パラメータとライセンスに関する情報を受け取ります。このパラメータのみが以降の操作で使用されます。
2. このデーモンは、管理されているすべてのコンポーネントから統合ログにメッセージを収集します。*stderr*コンポーネントに出力されたすべての情報は、Dr.Web ConfigDによって収集され、どのコンポーネントが出力したかを示すマークとともにDr.Web for UNIX Mail Serversの統合ログに書き込まれます。
3. シャットダウンすると、管理されているコンポーネントは終了コードを返します。コードが`101`、`102`、または`103`ではない場合、設定デーモンはこのコンポーネントを再起動します。そのため、コンポーネントが異常終了すると再起動され、*stderr*からのエラーメッセージがDr.Web for UNIX Mail Serversのログに登録されます。
   - コード101は、現在のライセンスでコンポーネントが動作できない場合に返されます。ライセンスパラメータを変更した後にのみ、コンポーネントが再起動されます。
   - コード102は、現在の設定パラメータでコンポーネントが動作できない場合に返されます。Dr.Web ConfigDは、設定パラメータが変更されたときにコンポーネントの再起動を試みます。
   - コード`103`は、Dr.Web ConfigDがリクエストに応じて起動したコンポーネント(Dr.Web Scanning EngineおよびDr.Web File Checker)が長い間アイドル状態だった場合に返されます。コンポーネントがシャットダウンするまでの時間は、その設定で指定されます(`IdleTimeLimit`パラメータ)。
   - 設定管理デーモンから受け取った設定パラメータをオンザフライで適用できない場合、Dr.Web ConfigDがそれを再起動するように、コンポーネントはコード`0`で存在します。
   - コンポーネントが設定デーモンに接続できない、または通信プロトコルエラーが発生する場合、コンポーネントは*stderr*に適切なメッセージを出力し、コード`1`で終了します。
4. シグナル交換は次のように構成されています。
   - Dr.Web ConfigDは`SIGHUP`シグナルをコンポーネントに送信し、変更した設定パラメータを適用します。
   - Dr.Web ConfigDは`SIGTERM`シグナルをコンポーネントに送信し、コンポーネントをシャットダウンします。このシグナルを受信した後、コンポーネントは30秒後にシャットダウンします。
   - 30秒経ってもコンポーネントがシャットダウンしない場合、Dr.Web ConfigDは`SIGKILL`シグナルを送信して強制的にシャットダウンします。

コマンドライン引数

設定デーモンDr.Web ConfigDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-configd [<parameters>]
```

設定デーモンDr.Web ConfigDは、以下のオプションを処理できます。

パラメータ	説明
`--help`	**機能**:コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**:`-h` **引数**:なし

`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし
`--config`	**説明**：指定した設定ファイルを今後の操作に使用します。 **短縮形**：`-c` **引数**：*<path to the file>* - 使用する設定ファイルへのパス。
`--daemonize`	**説明**：コンポーネントをデーモンモードで実行します。 **短縮形**：`-d` **引数**：なし
`--pid-file`	**説明**：指定されたPIDファイルを今後の操作に使用します。 **短縮形**：`-p` **引数**：*<path to the file>* - プロセスID（PID）の保存先ファイルへのパス。

例：

```
$ /opt/drweb.com/bin/drweb-configd -d -c /etc/opt/drweb.com/drweb.ini
```

このコマンドはDr.Web ConfigDを、`/etc/opt/drweb.com/drweb.ini`の設定ファイルを使用するデーモンとして実行します。

スタートアップノート

Dr.Web for UNIX Mail Servers の操作を有効にするには、Dr.Web ConfigDをデーモンとして実行する必要があります。通常は、Dr.Web ConfigDはOSの起動時に自動的に起動されます。そのため、Dr.Web ConfigDは、標準のOSディレクトリ（GNU/Linuxの場合は`/etc/init.d/`、FreeBSDの場合は`/usr/local/etc/rc.d/`）にある標準管理スクリプト`drweb-configd`と一緒に保存されます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントをリクエストするには、`man 1 drweb-configd`コマンドを使用します。

設定パラメータ

デーモンDr.Web ConfigDは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[Root]`セクションにある設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`DefaultLogLevel` *{logging level}*	すべてのDr.Web for UNIX Mail Serversコンポーネントについて、イベントロギングのデフォルトのロギングレベルを定義します。

パラメータ	説明
	このパラメータの値は、製品の中でロギングレベルを個別に設定できないすべてのコンポーネントに使用されます。 デフォルト値：Notice
LogLevel *{logging level}*	Dr.Web ConfigDのイベントロギングのロギングレベル。 デフォルト値：Notice
Log *{log type}*	設定デーモンのロギング方式と、このパラメータに別の値が指定されていないコンポーネントのロギング方式。 設定ファイルが読み込まれる前の初回起動時に、設定デーモンは次のパラメータ値を使用します。 • デーモンとして（-dオプションを付けて実行した場合）- SYSLOG:Daemon • その他の場合 - Stderr コンポーネントがバックグラウンドモードで動作している（コマンドラインから-dオプションを使用して起動した）場合は、Stderrの値をこのパラメータに*使用することはできません*。 デフォルト値：SYSLOG:Daemon
PublicSocketPath *{path to file}*	すべてのDr.Web for UNIX Mail Serversコンポーネント間の通信に使用されるソケットへのパス。 デフォルト値：/var/run/.com.drweb.public
AdminSocketPath *{path to file}*	昇格した（管理者の）権限を持つDr.Web for UNIX Mail Serversコンポーネント間の通信に使用されるソケットへのパス。 デフォルト値：/var/run/.com.drweb.admin
CoreEnginePath *{path to file}*	Dr.Web Virus-Finding Engineスキャンエンジンの動的ライブラリへのパス。 デフォルト値：*<var_dir>*/lib/drweb32.dll • GNU/Linuxの場合：/var/opt/drweb.com/lib/drweb32.dll • FreeBSDの場合：/var/drweb.com/lib/drweb32.dll
VirusBaseDir *{path to directory}*	ウイルスデータベースファイルがあるディレクトリへのパス。 デフォルト値：*<var_dir>*/bases • GNU/Linuxの場合：/var/opt/drweb.com/bases • FreeBSDの場合：/var/drweb.com/bases
KeyPath *{path to file}*	キーファイルへのパス（正規またはデモライセンス）。 デフォルト値：*<etc_dir>*/drweb32.key • GNU/Linuxの場合：/etc/opt/drweb.com/drweb32.key • FreeBSDの場合：/usr/local/etc/drweb.com/drweb32.key
CacheDir *{path to directory}*	キャッシュディレクトリへのパス（更新されたキャッシュとスキャンされたファイルに関する情報のキャッシュを保持するために使用されます）。

パラメータ	説明
	デフォルト値：*<var_dir>*/cache • GNU/Linuxの場合：/var/opt/drweb.com/cache • FreeBSDの場合：/var/drweb.com/cache
TempDir *{path to directory}*	一時ファイルがあるディレクトリへのパス。 デフォルト値：*システム環境変数*TMPDIR、TMP、TEMPまたはTEMPDIRからコピーされたパス（環境変数はこの順序で検索されます）。これらの環境変数がない場合は、/tmp。
RunDir *{path to directory}*	実行中のコンポーネントが有するすべてのPIDファイルと、Dr.Web for UNIX Mail Serversコンポーネント間の通信に使用されるソケットを含むディレクトリへのパス。 デフォルト値：/var/run
VarLibDir *{path to directory}*	Dr.Web for UNIX Mail Serversコンポーネントによって使用されるライブラリを含むディレクトリへのパス。 デフォルト値：*<var_dir>*/lib • GNU/Linuxの場合：/var/opt/drweb.com/lib • FreeBSDの場合：/var/drweb.com/lib
VersionDir *{path to directory}*	Dr.Web for UNIX Mail Serversコンポーネントの現在のバージョンに関する情報が格納されているディレクトリへのパス。 デフォルト値：*<var_dir>*/version • GNU/Linuxの場合：/var/opt/drweb.com/version • FreeBSDの場合：/var/drweb.com/version
DwsDir *{path to directory}*	インターネットリソースカテゴリーの自動的に更新されるデータベースのファイルを含むディレクトリへのパス。 デフォルト値：*<var_dir>*/dws • GNU/Linuxの場合：/var/opt/drweb.com/dws • FreeBSDの場合：/var/drweb.com/dws
AdminGroup *{group name | GID}*	Dr.Web for UNIX Mail Servers管理用の管理者権限を持つユーザーのグループ。*root*スーパーユーザーに加えて、これらのユーザーは、Dr.Web for UNIX Mail Serversコンポーネントの権限をスーパーユーザー権限に昇格させることができます。 デフォルト値：Dr.Web for UNIX Mail Serversのインストール中に*決定されます*。
TrustedGroup *{group name | GID}*	信頼するユーザーのグループ。このパラメータはネットワークトラフィックモニターコンポーネント（SpIDer Gate）の動作に使用されます。これらのユーザーのネットワークトラフィックはスキャンされずに、SpIDer Gateによってスキップされます。 ここに存在しないグループを指定することはできません。その場合、SpIDer Gateは起動に失敗します。

パラメータ	説明
	パラメータの値がない場合は、SpIDer Gate設定のOutputDivertパラメータにAuto値を指定することはできません。 デフォルト値：drweb
DebugIpc *{Boolean}*	詳細をデバッグレベルでログファイルに含めます（LogLevel = DEBUGの場合など）。IPCメッセージは、設定デーモンと他のコンポーネントとの間のやり取りを示します。 デフォルト値：No
UseCloud *{Boolean}*	悪意のあるファイルやURLに関する情報を受け取るためにDr.Web Cloudサービスを使用するかどうかを設定します。 デフォルト値：No
AntispamCorePath *{path to file}*	メールのスパムスキャンに使用されるサードパーティ製ライブラリのファイルへのパス（対応する機能が製品でサポートされている場合）。 デフォルト値：*<var_dir>*/lib/vaderetro.so • GNU/Linuxの場合：/var/opt/drweb.com/lib/vaderetro.so • FreeBSDの場合：/var/drweb.com/lib/vaderetro.so
AntispamDir *{path to directory}*	サードパーティ製ライブラリのファイルを含むディレクトリへのパス。 デフォルト値：*<var_dir>*/antispam • GNU/Linuxの場合：/var/opt/drweb.com/antispam • FreeBSDの場合：/var/drweb.com/antispam
VersionNotification *{Boolean}*	現在インストールされているDr.Web for UNIX Mail Serversバージョンのアップデートが利用可能であることをユーザーに通知します。 デフォルト値：Yes
UseVxcube *{Boolean}*	MTAに接続された外部フィルターのモードで、メール添付ファイルの解析にDr.Web vxCubeを使用します。 デフォルト値：No
VxcubeApiAddress *{string}*	Dr.Web vxCube APIサーバーが稼働しているホストのドメイン名（FQDN）またはIPアドレス。 デフォルト値：*（未設定）*
VxcubeApiKey *{string}*	Dr.Web vxCubeのAPIキー。 デフォルト値：*（未設定）*
VxcubeProxyUrl *{connection address}*	Dr.Web vxCubeへの接続に使用されるプロキシサーバーのアドレス。 認証のないHTTPプロキシにのみ対応しています。 以下の値を使用できます。*<connection address>* - http://*<host>*:*<port>*のフォーマットで示したプロキシサーバーの接続パラメータ。ここで、 • *<host>*はプロキシサーバーのホストアドレスです（IPアドレスまたはドメイン名、つまりFQDN）。

パラメータ	説明
	• *<port>* は使用するポートです。 デフォルト値：*（未設定）*

Dr.Web Ctl

このセクションの内容

- 概要
- リモートホストスキャン

概要

特別なDr.Web Ctlユーティリティ（`drweb-ctl`）を使用することで、OSのコマンドラインからDr.Web for UNIX Mail Serversの動作を管理できます。このユーティリティを使用して次の動作を実行できます。

- ブートレコードを含む、ファイルシステムオブジェクトのスキャンを開始する
- リモートネットワークホストでファイルのスキャンを開始する（下記の注意を参照）
- アンチウイルスコンポーネント（ディストリビューションに応じてウイルスデータベース、スキャンエンジンなど）の更新を開始する
- Dr.Web for UNIX Mail Servers設定のパラメータを確認・変更する
- Dr.Web for UNIX Mail Serversコンポーネントのステータスや検出された脅威に関する統計を確認する
- 集中管理サーバーに接続、または集中管理サーバーとの接続を切断する
- 隔離を表示し、隔離されたオブジェクトを管理する（Dr.Web File Checkerコンポーネント経由で）
- 集中管理サーバーに接続、または集中管理サーバーとの接続を切断する

Dr.Web for UNIX Mail Serversを管理するためのユーザーコマンドは、Dr.Web ConfigD設定デーモンの動作中にのみ適用されます（デフォルトでは、このコンポーネントはシステム起動時に自動的に起動します）。

一部のコントロールコマンドはスーパーユーザー権限を必要とします。

権限を昇格させるには`su`コマンド（カレントユーザーを変更する）または`sudo`コマンド（指定したコマンドを他のユーザーの権限で実行する）を使用します。

`drweb-ctl`ツールは、Dr.Web for UNIX Mail Serversの動作を管理するコマンドの自動補完をサポートしています（コマンドシェル内でこのオプションが有効になっている場合）。コマンドシェルが自動補完を許可していない場合、このオプションの設定を行うことができます。方法については、お使いのOSディストリビューションのマニュアルを参照してください。

シャットダウンする際、ツールはPOSIX準拠システムの表記規則に従って終了コードを返します。操作が正常に完了した場合は0(ゼロ)、それ以外の場合は0(ゼロ)以外です。

ツールが0以外(non-null)の終了コードを返すのは、内部エラーの場合のみであるという点に注意してください(ツールがコンポーネントに接続できなかった、リクエストされた操作を実行できなかったなど)。ツールが脅威を検出(そして駆除)した場合は、リクエストされた操作(`scan`など)が正常に実行されたため、0(null)終了コードを返します。検出された脅威と適用されたアクションのリストを明らかにする必要がある場合、コンソールに表示されたメッセージを分析してください。

すべてのエラーのコードについては、付録F. 既知のエラーのセクションのリストをご確認ください。

リモートホストスキャン

Dr.Web for UNIX Mail Serversを使用して、リモートネットワークホストにあるファイルの脅威に対するスキャンを実行できます。このようなホストには、フルコンピューティングマシン(ワークステーションやサーバーなど)だけでなく、ルーター、セットトップボックス、いわゆるモノのインターネット(IoT)と呼ばれるその他のスマートデバイスも含まれます。リモートスキャンを実行するには、リモートホストが*SSH*(*セキュアシェル*)または*Telnet*を介したリモート端末アクセスを提供する必要があります。デバイスにアクセスするには、リモートホストのIPアドレスとドメイン名、*SSH*または*Telnet*を介してリモートでシステムにアクセスするユーザーの認証情報を知っている必要があります。このユーザーは、スキャン済みファイルへのアクセス権限(少なくとも読み取り権限)を持っている必要があります。

この機能は、リモートホスト上の悪意のあるファイルや疑わしいファイルの検出にのみ使用できます。リモートスキャンの手段を用いた脅威の排除(すなわち、悪意のあるオブジェクトの隔離への移動、削除および修復)はできません。リモートホストで検出された脅威を排除するには、このホストによって直接提供される管理ツールを使用する必要があります。たとえば、ルーターやその他のスマートデバイスの場合は、ファームウェア更新のメカニズムを使用できます。コンピューティングマシンの場合は、コンピューティングマシンへの接続(任意でリモート端末モードを使用)とファイルシステムのそれぞれの操作(ファイルの削除または移動など)、またはコンピューティングマシンにインストールされているアンチウイルスソフトウェアの実行を介して実行できます。

リモートスキャンはコマンドラインツール`drweb-ctl`からのみ実行できます(`remotescan`コマンドを使用します)。

コマンドライン呼び出しフォーマット

1. 製品を管理するためのコマンドラインユーティリティを呼び出すコマンドフォーマット

Dr.Web for UNIX Mail Serversの動作を管理するコマンドラインツールの呼び出しフォーマットは以下のとおりです。

```
$ drweb-ctl [<general options> | <command> [<argument>] [<command options>]]
```

各パラメータは次のとおりです。

- *<general options>* - コマンドが指定されていない場合に起動時に適用できる、またはあらゆるコマンドにおいて適用できるオプションです。起動時に必須ではありません。

- *<command>* - Dr.Web for UNIX Mail Serversによって実行されるコマンド（スキャンの開始、隔離されたオブジェクトのリストの出力など）です。
- *<argument>* - コマンド引数です。指定されたコマンドに依存します。コマンドによってはない場合もあります。
- *<command options>* - 指定されたコマンドの動作を管理するためのオプションです。一部のコマンドでは省略できます。

2. 全般的なオプション

以下の全般的なオプションを使用できます。

オプション	説明
`-h, --help`	全般的なヘルプ情報を表示して終了します。いずれかのコマンドに関するヘルプ情報を表示させるには、以下の呼び出しを使用します。 `$ drweb-ctl <command> -h`
`-v, --version`	モジュールバージョンに関する情報を表示して終了します。
`-d, --debug`	指定されたコマンドの実行時にデバッグ情報を表示します。コマンドが指定されていない場合は実行できません。以下の呼び出しを使用します。 `$ drweb-ctl <command> -d`

3. コマンド

Dr.Web for UNIX Mail Serversを管理するコマンドは以下のグループに分けることができます。

- アンチウイルススキャンのコマンド
- 更新を管理し、集中管理モードでの動作を管理するコマンド
- 設定を管理するコマンド
- 検出された脅威および隔離を管理するコマンド
- 情報に関するコマンド

コマンドラインから製品のこのコンポーネントに関するヘルプを要求するには、`man 1 drweb-ctl`のコマンドを使用します。

3.1. アンチウイルススキャンのコマンド

アンチウイルススキャンを管理するコマンドには以下のものがあります。

コマンド	説明
`scan <path>`	**機能**：ファイルスキャンコンポーネントDr.Web File Checkerによる、指定されたファイルまたはディレクトリのスキャンを開始します。 **引数** *<path>* - スキャンするファイルまたはディレクトリへのパスです（パスは相対パスでも可）。 `--stdin`または`--stdin0`オプションを使用する場合、この引数は省略できます。特定の条件を満たす複数のファイルを指定するには、`find`ユーティリティ（使用例参照）および`--stdin`または`--stdin0`オプションを使用します。 **オプション** `-a [--Autonomous]`は、指定されたスキャンを実行し、完了後にそれらを終了させるために、Dr.Web Scanning EngineとDr.Web File Checkerの自律コピーを実行します。自律コピーによるスキャン中に検出された脅威は、`threats`コマンドによって表示される、検出された脅威のリストに追加されず（下記参照）、それらの脅威に関する情報は集中管理サーバーには送信されません（Dr.Web for UNIX Mail Serversが集中管理サーバーで管理されている場合）。 `--stdin` - スキャンのためのパスのリストを標準的な入力文字列（*stdin*）から取得します。リスト内のパスは改行文字（`\n`）で区切られている必要があります。 `--stdin0` - スキャンのためのパスのリストを標準的な入力文字列（*stdin*）から取得します。リスト内のパスはヌル文字（`\0`）で区切られている必要があります。 `--stdin`および`--stdin0`オプションを使用する場合、リストのパスに検索のパターンまたは正規表現を含めることはできません。`--stdin`および`--stdin0`オプションを使用して、外部ユーティリティ（`scan`コマンドの`find`など）によって生成されるパスリストを処理することをお勧めします（使用例を参照）。 `--Exclude <path>` - 除外するパスです。パスは相対パスにすることができ、ファイルマスクを含むことができます（ワイルドカード「`?`」と「`*`」、シンボルクラス「`[ ]`」、「`[! ]`」、「`[^ ]`」を使用できます）。 任意オプション、複数回設定できます。 `--Report <type>` - スキャンレポートのタイプを指定します。 使用可能な値： • `BRIEF` - 短いレポート • `DEBUG` - 詳細なレポート • `JSON` - JSON形式のシリアル化されたレポート

コマンド	説明
	デフォルト値：BRIEF --ScanTimeout <*number*> - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に0が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：0 --PackerMaxLevel <*number*> - 圧縮されたオブジェクトをスキャンする際のネスティングレベルの上限を指定します。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ArchiveMaxLevel <*number*> - 他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）をスキャンするときの最大ネスティングレベルを設定します（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のアーカイブ内のアーカイブはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MailMaxLevel <*number*> - 他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）をスキャンするときの最大ネスティングレベルを設定します（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ContainerMaxLevel <*number*> - 他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）をスキャンするときの最大ネスティングレベルを設定します。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MaxCompressionRatio <*ratio*> - スキャンされるオブジェクトの最大圧縮率を指定します。 値は2以上にする必要があります。 デフォルト値：3000 --MaxSizeToExtract <*size*> - アーカイブに含まれるファイルの最大サイズを指定します。このパラメータの値よりサイズが大きいファイルは、スキャン時にスキップされます。デフォルトでは、アーカイブ内のファイルのサイズ制限はありません。サイズは、サフィックス（b、kb、mb、gb）を付けた数値で指定します。サ

コマンド	説明
	フィックスが指定されていない場合、値はバイト単位のサイズとして扱われます。 デフォルト値：なし `--HeuristicAnalysis` *<On|Off>* - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：`On` `--OnKnownVirus` *<action>* - シグネチャベースの解析を使用して検出された既知の脅威に対して適用されるアクションです。 可能なアクション：`Report`、`Cure`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnIncurable` *<action>* - 修復不可能な脅威が検出された場合、または修復アクション（`Cure`）が失敗した場合に適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnSuspicious` *<action>* - ヒューリスティック解析によって検出された疑わしいオブジェクトに対して適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnAdware` *<action>* - 検出されたアドウェアに対して適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnDialers` *<action>* - 検出されたダイアラーに対して適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnJokes` *<action>* - 検出されたジョークプログラムに対して適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnRiskware` *<action>* - 検出されたリスクウェアに対して適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnHacktools` *<action>* - 検出されたハッキングツールに対して適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete` デフォルト値：`Report`

コマンド	説明
	 コンテナ（アーカイブ、メールメッセージなど）内のファイルで脅威が検出された場合は、ファイルを削除するアクション（Delete）の代わりにコンテナの隔離への移動（Quarantine）が実行されます。 --FollowSymlinks - シンボリックリンクを自動的に解決します。
bootscan *<disk drive>* | ALL	**機能**：ファイルスキャンコンポーネントDr.Web File Checkerを介して、指定されたディスク上のブートレコードのスキャンを開始します。MBRとVBRの両方のレコードがスキャンされます。 **引数** *<disk drive>* - ブートレコードをスキャンするディスクデバイスのブロックファイルへのパス。スペースで区切って複数のディスクデバイスを指定できます。引数は必須です。デバイスファイルの代わりにALLを指定した場合は、使用可能なすべてのディスクデバイスにあるすべてのブートレコードが確認されます。 **オプション** -a [--Autonomous]は、指定されたスキャンを実行し、完了後にそれらを終了させるために、Dr.Web Scanning EngineとDr.Web File Checkerの自律コピーを実行します。自律コピーによるスキャン中に検出された脅威は、threatsコマンドによって表示される、検出された脅威のリストに追加されず（下記参照）、それらの脅威に関する情報は集中管理サーバーには送信されません（Dr.Web for UNIX Mail Serversが集中管理サーバーで管理されている場合）。 --Report *<type>* - スキャンレポートのタイプを指定します。 使用可能な値： • BRIEF - 短いレポート • DEBUG - 詳細なレポート • JSON - JSON形式のシリアル化されたレポート デフォルト値：BRIEF --ScanTimeout *<number>* - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に0が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：0 --HeuristicAnalysis *<On|Off>* - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：On --Cure *<Yes|No>* - 脅威が検出された際に修復を試みる動作を有効または無効にします。 値にNoが指定された場合、検出された脅威に関する通知のみが表示されます。 デフォルト値：No --ShellTrace - ブートレコードをスキャンする際の、追加のデバッグ情報の表示を有効にします。

コマンド	説明
procscan	**機能**：Dr.Web File Checkerコンポーネントによる、現在実行中のシステムプロセスのコードを含んだ実行ファイルのスキャンを開始します。悪意のある実行ファイルが検出された場合、それらは駆除され、そのファイルによって実行されたすべてのプロセスを強制的に終了します。 **引数**：なし **オプション** `-a [--Autonomous]`は、指定されたスキャンを実行し、完了後にそれらを終了させるために、Dr.Web Scanning EngineとDr.Web File Checkerの自律コピーを実行します。自律コピーによるスキャン中に検出された脅威は、`threats`コマンドによって表示される、検出された脅威のリストに追加されず（下記参照）、それらの脅威に関する情報は集中管理サーバーには送信されません（Dr.Web for UNIX Mail Serversが集中管理サーバーで管理されている場合）。 `--Report` *<type>* - スキャンレポートのタイプを指定します。 使用可能な値： • `BRIEF` - 短いレポート • `DEBUG` - 詳細なレポート • `JSON` - JSON形式のシリアル化されたレポート デフォルト値：`BRIEF` `--ScanTimeout` *<number>* - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に`0`が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：`0` `--HeuristicAnalysis` *<On|Off>* - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：`On` `--PackerMaxLevel` *<number>* - 圧縮されたオブジェクトをスキャンする際のネスティングレベルの上限を指定します。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 値に`0`が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：`8` `--OnKnownVirus` *<action>* - シグネチャベースの解析を使用して検出された既知の脅威に対して適用されるアクションです。 可能なアクション：`Report`、`Cure`、`Quarantine`、`Delete` デフォルト値：`Report` `--OnIncurable` *<action>* - 修復不可能な脅威が検出された場合、または修復アクション（`Cure`）が失敗した場合に適用されるアクションです。 可能なアクション：`Report`、`Quarantine`、`Delete`

コマンド	説明
	デフォルト値：Report --OnSuspicious <*action*> - ヒューリスティック解析によって検出された疑わしいオブジェクトに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnAdware <*action*> - 検出されたアドウェアに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnDialers <action> - 検出されたダイアラーに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnJokes <*action*> - 検出されたジョークプログラムに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnRiskware <*action*> - 検出されたリスクウェアに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnHacktools <*action*> - 検出されたハッキングツールに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report 実行ファイルで脅威が検出された場合、Dr.Web for UNIX Mail Serversは、そのファイルによって開始されたすべてのプロセスを終了するという点に注意してください。
netscan [<*path*>]	**機能**：ネットワークデータスキャン用のDr.Web Network Checkerエージェントを介して、指定されたファイルまたはディレクトリの分散スキャンを開始します。UNIX向けDr.Webが動作している他のホストへの接続が設定されていない場合、スキャンはローカルで利用可能なスキャンエンジン経由でのみ実行されます（scanコマンドと同様）。 **引数** <*path*> - スキャンするファイルまたはディレクトリへのパスです。 この引数が指定されていない場合、入力スレッドstdinを介して受信したデータがスキャンされます。 **オプション** --Report <*type*> - スキャンレポートのタイプを指定します。

コマンド	説明
	使用可能な値： • BRIEF - 短いレポート • DEBUG - 詳細なレポート • JSON - JSON形式のシリアル化されたレポート デフォルト値：BRIEF --ScanTimeout <*number*> - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に0が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：0 --HeuristicAnalysis <*On*|*Off*> - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：On --PackerMaxLevel <*number*> - 圧縮されたオブジェクトをスキャンする際のネスティングレベルの上限を指定します。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ArchiveMaxLevel <*number*> - 他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）をスキャンするときの最大ネスティングレベルを設定します（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のアーカイブ内のアーカイブはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MailMaxLevel <*number*> - 他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）をスキャンするときの最大ネスティングレベルを設定します（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ContainerMaxLevel <*number*> - 他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）をスキャンするときの最大ネスティングレベルを設定します。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8

コマンド	説明
	`--MaxCompressionRatio` *<ratio>* - スキャンされるオブジェクトの最大圧縮率を指定します。 値は2以上にする必要があります。 デフォルト値：3000 `--MaxSizeToExtract` *<size>* - アーカイブに含まれるファイルの最大サイズを指定します。このパラメータの値よりサイズが大きいファイルは、スキャン時にスキップされます。デフォルトでは、アーカイブ内のファイルのサイズ制限はありません。サイズは、サフィックス（b、kb、mb、gb）を付けた数値で指定します。サフィックスが指定されていない場合、値はバイト単位のサイズとして扱われます。 デフォルト値：なし `--Cure` *<Yes/No>* - 脅威が検出された際に修復を試みる動作を有効または無効にします。 値に`No`が指定された場合、検出された脅威に関する通知のみが表示されます。 デフォルト値：`No`
`flowscan` *<path>*	機能：「flow」メソッドを使用して、指定されたファイルまたはディレクトリのスキャンを、Dr.Web File Checkerを介して開始します。 ⚠ ファイルやディレクトリのオンデマンドスキャンには、`scan`コマンドを使用することをお勧めします。 **引数** *<path>* - スキャンするファイルまたはディレクトリへのパスです。 **オプション** `--ScanTimeout` *<number>* - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に`0`が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：`0` `--HeuristicAnalysis` *<On/Off>* - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：`On` `--PackerMaxLevel` *<number>* - 圧縮されたオブジェクトをスキャンする際のネスティングレベルの上限を指定します。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 値に`0`が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：`8` `--ArchiveMaxLevel` *<number>* - 他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）をスキャンするときの最大ネスティングレベルを設定

コマンド	説明
	します（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のアーカイブ内のアーカイブはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 `--MailMaxLevel` *<number>* - 他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）をスキャンするときの最大ネスティングレベルを設定します（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 `--ContainerMaxLevel` *<number>* - 他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）をスキャンするときの最大ネスティングレベルを設定します。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 `--MaxCompressionRatio` *<ratio>* - スキャンされるオブジェクトの最大圧縮率を指定します。 値は2以上にする必要があります。 デフォルト値：3000 `--OnKnownVirus` *<action>* - シグネチャベースの解析を使用して検出された既知の脅威に対して適用されるアクションです。 可能なアクション：Report、Cure、Quarantine、Delete デフォルト値：Report `--OnIncurable` *<action>* - 修復不可能な脅威が検出された場合、または修復アクション（Cure）が失敗した場合に適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report `--OnSuspicious` *<action>* - ヒューリスティック解析によって検出された疑わしいオブジェクトに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report `--OnAdware` *<action>* - 検出されたアドウェアに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report `--OnDialers` *<action>* - 検出されたダイアラーに対して適用されるアクションです。

コマンド	説明
	可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnJokes <action> - 検出されたジョークプログラムに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnRiskware <action> - 検出されたリスクウェアに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report --OnHacktools <action> - 検出されたハッキングツールに対して適用されるアクションです。 可能なアクション：Report、Quarantine、Delete デフォルト値：Report コンテナ（アーカイブ、メールメッセージなど）内のファイルで脅威が検出された場合は、ファイルを削除するアクション（Delete）の代わりにコンテナの隔離への移動（Quarantine）が実行されます。
rawscan <path>	機能：Dr.Web File Checkerを使用せずに、指定されたファイルまたはディレクトリのDr.Web Scanning Engineによる「raw」スキャンを直接開始します。 「raw」スキャンで検出された脅威はthreatsコマンドで表示される、検出された脅威のリストには含まれないということに注意してください（下記参照）。 このコマンドは、Dr.Web Scanning Engineの機能をデバッグするためにのみ使用することをお勧めします。ファイル内で検出された脅威のうち少なくとも*1つ*の脅威が駆除されている場合、コマンドは「cured」（修復済み）ステータスを出力します（*すべての*脅威が駆除されているとは限りません）。そのため、徹底的なファイルスキャンが必要な場合にこのコマンドを使用することは*お勧めできません*。その場合、scanコマンドを使用することをお勧めします。 **引数** <path> - スキャンするファイルまたはディレクトリへのパスです。 **オプション** --ScanEngine <path> - Dr.Web Scanning EngineのUNIXソケットへのパスです。指定していない場合、スキャンエンジンの自律インスタンスが開始されます（スキャンが完了するとシャットダウンされます）。 --Report <type> - スキャンレポートのタイプを指定します。

コマンド	説明
	使用可能な値： • BRIEF - 短いレポート • DEBUG - 詳細なレポート • JSON - JSON形式のシリアル化されたレポート デフォルト値：BRIEF --ScanTimeout <*number*> - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に0が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：0 --PackerMaxLevel <*number*> - 圧縮されたオブジェクトをスキャンする際のネスティングレベルの上限を指定します。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ArchiveMaxLevel <*number*> - 他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）をスキャンするときの最大ネスティングレベルを設定します（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のアーカイブ内のアーカイブはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MailMaxLevel <*number*> - 他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）をスキャンするときの最大ネスティングレベルを設定します（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ContainerMaxLevel <*number*> - 他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）をスキャンするときの最大ネスティングレベルを設定します。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MaxCompressionRatio <*ratio*> - スキャンされるオブジェクトの最大圧縮率を指定します。 値は2以上にする必要があります。

コマンド	説明
	デフォルト値：3000 `--MaxSizeToExtract` *<size> - アーカイブに含まれるファイルの最大サイズを指定します。このパラメータの値よりサイズが大きいファイルは、スキャン時にスキップされます。デフォルトでは、アーカイブ内のファイルのサイズ制限はありません。サイズは、サフィックス（b、kb、mb、gb）を付けた数値で指定します。サフィックスが指定されていない場合、値はバイト単位のサイズとして扱われます。* デフォルト値：なし `--HeuristicAnalysis` *<On|Off>* - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：`On` `--Cure` *<Yes|No>* - 脅威が検出された際に修復を試みる動作を有効または無効にします。 値に`No`が指定された場合、検出された脅威に関する通知のみが表示されます。 デフォルト値：`No` `--ListCleanItem` - スキャンされたコンテナ内で見つかったクリーンな（感染していない）ファイルのリストの出力を有効にします。 `--ShellTrace` - ファイルをスキャンする際の、追加のデバッグ情報の表示を有効にします。 `--Output` *<path to file>* - コマンドの出力を指定されたファイルに複製します。
`remotescan` *<host> <path>*	**機能**：*SSH*または*Telnet*を使用して接続することにより、指定されたリモートホスト上の指定されたファイルまたはディレクトリのスキャンを開始します。 ⚠ リモートスキャンで検出された脅威は駆除されず、`threats`コマンドで表示される、検出された脅威のリストには含まれないということに注意してください（下記参照）。 この機能はリモートホストの悪意のあるファイルや疑わしいファイルの検出にのみ使用できます。リモートホストで検出された脅威を排除するには、このホストから直接提供される管理ツールを使用する必要があります。たとえば、ルーターの場合、セットトップボックスおよびその他のスマートデバイス、つまりファームウェア更新のためのメカニズムを使用することができます。コンピューティングマシンの場合、それらのマシン（オプションとして、リモートターミナルモードを使用）およびそれらのファイルシステムのそれぞれの操作（ファイルの削除または移動など）に接続するか、それらにインストールされたアンチウイルスソフトウェアを実行して行うことができます。 **引数** • *<host>* - リモートホストのIPアドレスまたはドメイン名です。 • *<path>* - スキャンするファイルまたはディレクトリへのパスです（パスは絶対パスでなければなりません）。

コマンド	説明
	オプション `-m [--Method]` *<SSH|Telnet>* - リモートホスト接続方法（プロトコル）です。 方法が指定されていない場合は、SSHが使用されます。 `-l [--Login]` *<name>* - 選択されたプロトコル経由でリモートホストでの承認に使用されるログインID（ユーザー名）です。 ユーザー名が指定されていない場合、コマンドを起動したユーザー名を用いてリモートホストに接続しようとします。 `-i [--Identity]` *<path to file>* - 選択されたプロトコル経由で指定されたユーザーの認証に使用されるプライベートキーが含まれるファイルへのパスです。 `-p [--Port]` *<number>* - 選択されたプロトコル経由で接続するリモートホストのポート番号です。 デフォルト値：選択したプロトコル用のデフォルトポート（SSHでは`22`、Telnetでは`23`） `--ForceInteractive` - SSHインタラクティブセッションを使用します（*SSH*接続の場合のみ）。 *オプション機能です。* `--TransferListenAddress` *<address>* - リモートデバイスからスキャン用に送信されるファイルを受信するためにリッスンされるアドレスです。 オプション機能です。指定されなかった場合、任意のアドレスが使用されます。 `--TransferListenPort` *<port>* - リモートデバイスからスキャン用に送信されるファイルを受信するためにリッスンされるポートです。 オプション機能です。指定されなかった場合、任意のポートが使用されます。 `--TransferExternalAddress` *<address>* - スキャン用にファイルを送信するためにリモートデバイスに指定されるアドレスです。 オプション機能です。指定されなかった場合、"`--TransferListenAddress`"の値、またはすでに確立されているセッションの送信アドレスが使用されます。 `--TransferExternalPort` *<port>* - スキャン用にファイルを送信するためにリモートデバイスに指定されるポートです。 オプション機能です。指定されなかった場合、自動的に決定されたポートが使用されます。 `--Password` *<password>* - 選択されたプロトコルを介してユーザー認証に使用されるパスワードです。 パスワードはプレーンテキストとして転送されることに注意してください。 `--Exclude` *<path>* - スキャンの対象から除外するパスです。パスにはファイルマスクを含むことができます（ワイルドカード「`?`」と「`*`」、シンボルクラス「`[ ]`」、「`[! ]`」、「`[^ ]`」を使用することができます）。パス（ファイルマスクを含むパスを含む）は絶対パスである必要があります。 任意オプション、複数回設定できます。 `--Report` *<type>* - スキャンレポートのタイプを指定します。

コマンド	説明
	使用可能な値： • BRIEF - 短いレポート • DEBUG - 詳細なレポート • JSON - JSON形式のシリアル化されたレポート デフォルト値：BRIEF --ScanTimeout <*number*> - 1つのファイルをスキャンする際のタイムアウトをミリ秒で指定します。 値に0が指定された場合、スキャンにかかる時間は制限されません。 デフォルト値：0 --PackerMaxLevel <*number*> - 圧縮されたオブジェクトをスキャンする際のネスティングレベルの上限を指定します。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ArchiveMaxLevel <*number*> - 他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）をスキャンするときの最大ネスティングレベルを設定します（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のアーカイブ内のアーカイブはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MailMaxLevel <*number*> - 他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）をスキャンするときの最大ネスティングレベルを設定します（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --ContainerMaxLevel <*number*> - 他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）をスキャンするときの最大ネスティングレベルを設定します。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 値に0が指定された場合、ネストされたオブジェクトはスキャン中にスキップされます。 デフォルト値：8 --MaxCompressionRatio <*ratio*> - スキャンされるオブジェクトの最大圧縮率を指定します。 値は2以上にする必要があります。

コマンド	説明
	デフォルト値：3000 `--MaxSizeToExtract` *<size> - アーカイブに含まれるファイルの最大サイズを指定します。このパラメータの値よりサイズが大きいファイルは、スキャン時にスキップされます。デフォルトでは、アーカイブ内のファイルのサイズ制限はありません。サイズは、サフィックス（b、kb、mb、gb）を付けた数値で指定します。サフィックスが指定されていない場合、値はバイト単位のサイズとして扱われます。* デフォルト値：なし `--HeuristicAnalysis` *<On|Off>* - スキャン中のヒューリスティック解析を有効または無効にします。 デフォルト値：`On`
`checkmail` *<path to file>*	**機能**：脅威、スパムの兆候、悪意のあるリンク、メール処理ルールへの不適合を検出するために、ファイルに保存されたメールメッセージのスキャンを実行します（メール処理コンポーネントであるDr.Web MailDを使用します）。コンソールの出力スレッド（*stdout*）には、スキャンの結果と、メール処理コンポーネントのDr.Web MailDによるスキャン中にこのメッセージに対して適用されたアクションが表示されます。 **引数** *<path to file>* - スキャンが必要なメールメッセージのファイルへのパスです。必須の引数です。 **オプション** `--Report` *<type>* - スキャンレポートのタイプを指定します。 使用可能な値： • `BRIEF` - 短いレポート • `DEBUG` - 詳細なレポート • `JSON` - JSON形式のシリアル化されたレポート デフォルト値：`BRIEF` `-r [--Rules]` *<list of rules>* - メールメッセージのスキャン中に従うルールのリストを指定します。 ルールが指定されなかった場合、デフォルトで指定されている以下のルールセットが適用されます。 `threat_category in (KnownVirus, VirusModification, UnknownVirus, Adware, Dialer) : REJECT` `total_spam_score gt 0.80 : REJECT` `url_category in (InfectionSource, NotRecommended, CopyrightNotice) : REJECT` Dr.Web Anti-Spamがインストールされていない場合、スパムのスキャンルール（2番目の文字列）はセットから自動的に除外されます。 `-c [--Connect]` *<IP>:<port>* - スキャンされるメッセージの送信者の接続用アドレスとして使用されるネットワークソケットを指定します。 `-e [--Helo]` *<name>* - メッセージを送信したクライアントの識別子を指定します（IPアドレスまたはFQDNホスト、SMTPコマンド`HELO/EHLO`の場合）。

コマンド	説明
	`-f [--From]` *<email>* - 送信者のメールアドレスを指定します（SMTPコマンド`MAIL FROM`の場合）。 アドレスが指定されていない場合、それぞれのメールのアドレスが使用されます。 `-t [--Rcpt]` *<email>* - 受信者のメールアドレスを指定します（SMTPコマンド`RCPT TO`の場合）。 アドレスが指定されていない場合、それぞれのメールのアドレスが使用されます。 メール処理コンポーネントがインストールされていない場合、このコマンドを呼び出すとエラーが返されます。
`mailquarantine`	**機能**：メールメッセージキューを管理する補助的なDr.Web Mail Quarantineコンポーネントを設定します。 **引数**：なし **オプション** `--Flush` - 指定されたキューから即時処理用のキューに、スケジュールされたメッセージを移動します。`--Queue`オプションが指定されている必要があります。以下の呼び出しを使用します。 `$ drweb-ctl mailquarantine --Queue <queue> --Flush` `--Show` - 指定されたメッセージキューを表示します。`--Queue`オプションが指定されている必要があります。以下の呼び出しを使用します。 `$ drweb-ctl mailquarantine --Queue <queue> --Show` `--Stat` - すべてのメッセージキューに関する統計情報を表示します。 `--CheckHealth` - メッセージデータベースの一貫性チェックを実行します。 `--FixHealth` - メッセージデータベースの一貫性エラーを修正します。 `-q [--Queue]` *<queue>* - 処理するメッセージキューを指定します。 使用可能な値： • `SmtpFresh` - SMTPモードでチェックされるメッセージ • `SmtpAccepted` - SMTPモードでチェックされ、承認されたメッセージ • `BccFresh` - BCCモードでチェックされるメッセージ • `BccAccepted` - BCCモードでチェックされ、承認されたメッセージ `-l [--Limit]` *<number>* - 選択されたキューからのメッセージの最大表示数を設定します。 `-d [--Debug]` - 指定されたコマンドの実行時にデバッグ情報を表示します。コマンドが指定されていない場合は実行できません。以下の呼び出しを使用します。

コマンド	説明
	`$ drweb-ctl mailquarantine <command> -d`

3.2. 更新および集中管理モードでの動作を管理するコマンド

更新および集中管理モードでの動作を管理するコマンドには、以下のものがあります。

コマンド	説明
`update`	**機能**：Dr.Web MeshDを介して、Doctor Webの更新サーバーまたはローカルクラウドから、アンチウイルスコンポーネント（ディストリビューションによって、ウイルスデータベース、スキャンエンジンなど）の更新を開始し、更新プロセスがすでに実行されている場合はそれを終了するか、または更新ファイルの最新の更新を以前のバージョンへロールバックします。 Dr.Web for UNIX Mail Serversが集中管理サーバーに接続されている場合、このコマンドは効力を持ちません。 **引数**：なし **オプション** `-l [--local-cloud]` - Dr.Web for UNIX Mail Serversに接続されたローカルクラウドを使用して更新をダウンロードします。このオプションが指定されていない場合、更新はDoctor Web更新サーバーからダウンロードされます（デフォルトの動作）。 `--From <path>` - 指定されたディレクトリからオフラインで更新を適用します。 `--Path <path>` - オフラインで更新するファイルを指定されたディレクトリに保存します。このディレクトリにすでにファイルがある場合は、それらが更新されます。 `--Rollback` - 最後の更新をロールバックし、更新されたファイルの以前のバージョンを復元します。 `--Stop` - 実行中の更新プロセスを終了します。
`esconnect <server>[:<port>]`	**機能**：指定された集中管理サーバー（Dr.Web Enterprise Serverなど）にDr.Web for UNIX Mail Serversを接続します。動作モードの詳細については、動作モードを参照してください。 **引数** • *<server>* - 集中管理サーバーが動作しているホストのIPアドレスまたはホスト名です。この引数は必須です。 • *<port>* - 集中管理サーバーによって使用されるポート番号です。この引数は任意であり、集中管理サーバーが標準以外のポートを使用する場合にのみ指定する必要があります。

コマンド	説明
	オプション `--Certificate` *<path>* - 接続する集中管理サーバーの証明書ファイルへのパスです。 `--Login` *<ID>* - 集中管理サーバーへの接続に使用されるログインID(ワークステーションID)です。 `--Password` *<password>* - 集中管理サーバーへの接続用パスワードです。 `--Group` *<ID>* - ワークステーションが接続時に追加されるグループのIDです。 `--Rate` *<ID>* - ワークステーションが集中管理サーバーグループの1つに含まれている場合に、そのワークステーションに適用される課金プライングループのIDです(`--Group`オプションと一緒にのみ指定できます)。 `--Compress` *<On|Off>* - 送信されたデータの強制的な圧縮を有効(`On`)または無効(`Off`)にします。指定しない場合、圧縮の使用はサーバーによって決定されます。 `--Encrypt` *<On|Off>* - 送信されたデータの強制的な暗号化を有効(`On`)または無効(`Off`)にします。指定しない場合、暗号化の使用はサーバーによって決定されます。 `--Newbie` - 「新規端末」として接続します(サーバーで新しいアカウントを取得します)。 このコマンドは、`drweb-ctl`を*root*権限で実行する必要があります。必要に応じて、`su`または`sudo`コマンドを使用してください。
`esdisconnect`	**機能**: Dr.Web for UNIX Mail Serversを集中管理サーバーから切断し、その動作をスタンドアロンモードに切り替えます。 Dr.Web for UNIX Mail Serversがすでにスタンドアロンモードで動作している場合、このコマンドは効力を持ちません。 **引数**: なし **オプション**: なし このコマンドは、`drweb-ctl`を*root*権限で実行する必要があります。必要に応じて、`su`または`sudo`コマンドを使用してください。

3.3. 設定を管理するコマンド

設定を管理するコマンドには以下のものがあります。

コマンド	説明
`cfset` *<section>*. *<parameter>* *<value>*	**機能**：Dr.Web for UNIX Mail Serversの現在の設定で、指定されたパラメータのアクティブな値を変更します。 **引数** • *<section>* - パラメータのある設定ファイルのセクション名です。この引数は必須です。 • *<parameter>* - パラメータの名前です。この引数は必須です。 • *<value>* - 新しいパラメータ値です。この引数は必須です。 パラメータ値を指定するには、*<section>.<parameter> <value>*という形式を使用します。代入記号「=」はここでは使用しません。 複数のパラメータ値を指定する場合は、追加するパラメータ値の数だけ`cfset`コマンドの呼び出しを繰り返す必要があります。パラメータ値のリストに新しい値を追加するには、`-a`オプションを使用します（以下を参照）。文字列"*<value 1>, <value 2>*"が*<parameter>*の1つの値と見なされてしまうため、文字列 *<parameter> <value 1>, <value 2>*を引数として指定することはできません。 設定ファイルに関する詳細は、付録D. Dr.Web for UNIX Mail Servers設定ファイルのセクションを参照してください。また、`man 5 drweb.ini`によって表示されるドキュメントページも参照してください。 **オプション** `-a [--Add]` - 現在のパラメータ値を置き換えず、指定された値をリストに追加します（リストとして指定された複数の値を持つことのできるパラメータに対してのみ使用可能）。このオプションは、タグを付けたパラメータの新しいグループを追加する場合にも使用してください。 `-e [--Erase]` - 現在のパラメータ値を置き換えず、指定された値をリストから削除します（リストとして指定された複数の値を持つことのできるパラメータに対してのみ使用可能）。 `-r [--Reset]` - パラメータ値をデフォルトにリセットします。その際、コマンド内で*<value>*は必要なく、指定された場合は無視されます。 オプションは必須ではありません。指定されなかった場合は、現在のパラメータ値（パラメータに複数の値がある場合は値の全リスト）が指定された値に置き換えられます。 Dr.Web ClamDの接続ポイントの個々のパラメータを記述するセクションでは、`-r`オプションを使用すると、個々の設定セクションのパラメータ値が、該当するコンポーネントの設定にある親パラメータの値に置き換えられます。

コマンド	説明
	Dr.Web ClamDの新しい接続ポイント *<point>* を追加する必要がある場合は、次のコマンドを使用します。 `cfset ClamD.Endpoint.<point> -a` 例：`cfset ClamD.Endpoint.point1 -a` このコマンドは、`drweb-ctl`をroot権限で実行する必要があります。必要に応じて、`su`または`sudo`コマンドを使用してください。
`cfshow [<section>[.<parameter>]]`	**機能**：Dr.Web for UNIX Mail Serversの現在の設定のパラメータを表示します。 パラメータを表示するコマンドは *<section>.<parameter> = <value>* のように指定します。インストールされていないコンポーネントのセクションとパラメータは表示されません。 **引数** • *<section>* - 表示するパラメータのある設定ファイルのセクション名です。この引数は任意です。指定されなかった場合、すべての設定ファイルセクションのパラメータが表示されます。 • *<parameter>* - 表示するパラメータの名前です。この引数は任意です。指定されなかった場合、セクションのすべてのパラメータが表示されます。それ以外の場合は、このパラメータのみが表示されます。セクション名なしにパラメータが指定された場合、すべての設定ファイルセクションにある、その名前を持つすべてのパラメータが表示されます。 **オプション** `--Uncut` - すべての設定パラメータを表示します（現在インストールされているコンポーネントのセットによって使用されているもの以外も含む）。このオプションが指定されていない場合、インストールされたコンポーネントの設定に使用されているパラメータのみが出力されます。 `--Changed` - デフォルトの値と異なる値を持つパラメータのみを表示します。 `--Ini` - パラメータ値をINIファイルフォーマットで表示します。まず角括弧内でセクション名が指定され、次に *<parameter> = <value>* ペアでセクションパラメータが表示されます（1行につき1ペア）。 `--Value` - 指定されたパラメータの値のみを表示します（この場合、*<parameter>* 引数は必須です）。
`reload`	**機能**：Dr.Web ConfigD設定デーモンに`SIGHUP`シグナルを送信します。 このシグナルを受信すると、Dr.Web ConfigD設定デーモンは設定を再読み込みし、そこに加えられた変更をDr.Web for UNIX Mail Serversのコンポーネントに送信します。次に、Dr.Web for UNIX Mail Serversログを再度開き、ウイルスデータベースを使用するコンポーネント（スキャンエンジンを含む）を再起動させ、異常終了したコンポーネントの再起動を試みます。 **引数**：なし **オプション**：なし

3.4. 検出された脅威および隔離を管理するコマンド

検出された脅威および隔離を管理するコマンドには以下のものがあります。

コマンド	説明
`threats [<action> <object>]`	**機能**：検出された脅威のうち、識別子で選択されたものに対して、指定されたアクションを適用します。アクションの種類はコマンドオプションによって指定します。 アクションが指定されていない場合、検出されたが駆除されていない脅威に関する情報を表示します。脅威に関する情報は、オプションの`--Format`引数で指定されたフォーマットに従って表示されます。`--Format`引数が指定されていない場合は、各脅威に関する次の情報が表示されます。 • 脅威に対して割り当てられた識別子（順序数） • 感染したファイルへのフルパス • 脅威に関する情報（脅威の名前、Doctor Webの分類による脅威の種類） • ファイルに関する情報（サイズ、ファイル所有者のユーザー名、最後に変更された時間） • 脅威に対して適用された操作の履歴（検出、適用されたアクションなど） **引数**：なし **オプション** `--Format "<format string>"` - 脅威に関する情報を指定されたフォーマットで表示します。フォーマット文字列の説明は以下のとおりです。 このオプションがアクションオプションと一緒に指定されている場合は無視されます。 `-f [--Follow]` - 新しい脅威に関する新しいメッセージを待ち、それらを受け取り次第、表示します（CTRL+Cで待機を中断します）。 このオプションがアクションオプションと一緒に指定されている場合は無視されます。 `--Directory <list of directories>` - `<list of directories>`で指定したディレクトリ内のファイルで検出された脅威のみを表示します。 このオプションが以下のオプションと一緒に適用された場合は無視されます。 `--Cure <threat list>` - 指定された脅威の修復を試みます（脅威の識別子をコンマ区切りで指定）。 `--Quarantine <threat list>` - 指定された脅威を隔離に移します（脅威の識別子をコンマ区切りで指定）。 `--Delete <threat list>` - 指定された脅威を削除します（脅威の識別子をコンマ区切りで指定）。 `--Ignore <threat list>` - 指定された脅威を無視します（脅威の識別子をコンマ区切りで指定）。 検出されたすべての脅威に対してアクションを適用する必要がある場合は、`<threat list>`の代わりに`All`を指定します。例： `$ drweb-ctl threats --Quarantine All`

コマンド	説明
	この例では、検出された悪意のあるオブジェクトすべてを隔離に移します。
`quarantine [<action> <object>]`	**機能**：隔離内の指定されたオブジェクトに対してアクションを適用します。 アクションが指定されなかった場合、隔離されたオブジェクトに関する情報とそのIDが、隔離に移された元のファイルに関する簡単な情報と一緒に表示されます。隔離されたオブジェクトに関する情報は、オプションの`--Format`引数で指定されたフォーマットに従って表示されます。`--Format`引数が指定されていない場合は、隔離された各オブジェクトについて次の情報が表示されます。 • 隔離されたオブジェクトに対して割り当てられた識別子 • 隔離に移される前の、元のファイルへのパス • ファイルが隔離に移された日付 • ファイルに関する情報（サイズ、ファイル所有者のユーザー名、最後に変更された時間） • 脅威に関する情報（脅威の名前、Doctor Webの分類による脅威の種類） **引数**：なし **オプション** `-a [--Autonomous]` - 指定された隔離コマンドを実行するためにDr.Web File Checkerファイルスキャンコンポーネントの個別のインスタンスを開始し、完了後にそれを終了します。 このオプションは以下のオプションと一緒に適用できます。 `--Format "<format string>"` - 隔離されたオブジェクトに関する情報を指定されたフォーマットで表示します。フォーマット文字列の説明は以下のとおりです。 このオプションがアクションオプションと一緒に指定されている場合は無視されます。 `-f [--Follow]` - 新しい脅威に関する新しいメッセージを待ち、それらを受け取り次第、表示します（CTRL+Cで待機を中断します）。 このオプションがアクションオプションと一緒に指定されている場合は無視されます。 `--Discovery [<list of directories>,]` - 指定されたディレクトリのリストで隔離ディレクトリを検索し、脅威を検出すると、統合された隔離に追加します。*<list of directories>*が指定されていない場合は、ファイルシステムの共通の場所（ボリュームマウントポイントとユーザーホームディレクトリ）にある隔離ディレクトリを検索します。 このオプションは`-a（--Autonomous）`オプション（上記を参照）だけでなく、下記に一覧で示されている任意のオプションおよびアクションとともに指定できます。さらに、自律コピーとして`quarantine`コマンドを起動すると（`-a（--Autonomous）`オプションを指定して、`--Discovery`オプションは指定しない場合）、次の呼び出しと同じになります。 `quarantine --Autonomous --Discovery` `--Delete <object>` - 指定されたオブジェクトを隔離から削除します。 オブジェクトは隔離から永久に削除されることに注意してください。この操作は元に戻せません。

コマンド	説明
	`--Cure` *<object>* - 隔離内の指定されたオブジェクトの修復を試みます。 オブジェクトが修復された場合であっても、それは隔離内に残ります。修復されたオブジェクトを隔離から復元するには`--Restore`オプションを使用します。 `--Restore` *<object>* - 指定されたオブジェクトを隔離から元の場所に復元します。 このコマンドでは、`drweb-ctl`をroot権限で起動する必要がある場合があります。感染していても隔離からファイルを復元できます。 `--TargetPath` *<path>* -オブジェクトを隔離から指定された場所に復元します。指定された名前を持つファイルとして復元するか(*<path>*がファイルへのパスであった場合)、またはただ単に指定されたディレクトリに復元します(*<path>*がディレクトリへのパスであった場合)。パスは絶対パスでも相対パスでも構いません(現在のディレクトリを参照)。 このオプションは、`--Restore`オプションとの組み合わせでのみ使用できます。 *<object>*は隔離内のオブジェクトの識別子を指定します。隔離されたすべてのオブジェクトに対してアクションを適用する場合は、*<object>*の代わりに`All`を指定してください。例: `$ drweb-ctl quarantine --Restore All --TargetPath test` すべての隔離されたオブジェクトを、`drweb-ctl`コマンドが起動されたカレントディレクトリにある`test`サブディレクトリに復元します。 `--Restore All`では、追加のオプション`--TargetPath`(指定された場合)にはファイルへのパスではなくディレクトリへのパスを指定する必要があります。

脅威および隔離コマンド用のフォーマット出力

出力フォーマットは、オプションの`--Format`引数として指定されたフォーマット文字列を使用して定義されます。フォーマット文字列は引用符で囲んで指定する必要があります。フォーマット文字列には、特定の情報として表示される特殊なマーカーだけでなく、一般的な記号(「そのまま」で表示されるもの)を含めることができます。以下のマーカーを使用することができます。

1. `threats`と`quarantine`コマンドに共通:

マーカー	説明
`%{n}`	新しい文字列
`%{t}`	集計
`%{threat_name}`	Doctor Webの分類に従って検出された脅威(ウイルス)の名前
`%{threat_type}`	Doctor Webの分類に従った脅威の種類(「既知のウイルス」など)
`%{size}`	元のファイルサイズ
`%{origin}`	パスを含む元のファイルのフルネーム

マーカー	説明
%{path}	%{origin}の同義語
%{ctime}	元のファイルが変更された日時（*"%Y-%b-%d %H:%M:%S"*フォーマット、例："2018-Jul-20 15:58:01"）
%{timestamp}	%{ctime}と似ているが、*UNIXのタイムスタンプ*フォーマット
%{owner}	元のファイル所有者のユーザー名
%{rowner}	元のファイルのリモートユーザー所有者（該当しない場合や値が不明な場合は?と置き換えられます）

2. threatsコマンドに固有：

マーカー	説明
%{hid}	脅威に関連付けられているイベントの履歴にある脅威レコードのID
%{tid}	脅威のID
%{htime}	脅威に関連したイベントの日時
%{app}	脅威を処理したDr.Web for UNIX Mail ServersコンポーネントのID
%{event}	脅威に関連する最新イベント： • FOUND - 脅威が検出されました。 • Cure - 脅威は修復されました。 • Quarantine - 脅威のあるファイルが隔離に移されました。 • Delete - 脅威のあるファイルが削除されました。 • Ignore - 脅威は無視されました。 • RECAPTURED - 他のコンポーネントによって脅威が再度検出されました。
%{err}	エラーメッセージテキスト（エラーが空の文字列に置き換えられない場合）

3. quarantineコマンドに固有：

マーカー	説明
%{qid}	隔離されたオブジェクトのID
%{qtime}	オブジェクトを隔離に移動した日時
%{curetime}	隔離に移されたオブジェクトの修復を試みた日時（該当しない場合または値が不明の場合は?に置き換えられます）
%{cureres}	隔離されたオブジェクトの修復を試みた結果： • cured - 脅威は修復されています。 • not cured - 脅威は修復されていないか、修復が試みられていません。

例

```
$ drweb-ctl quarantine --Format "{%{n} %{origin}: %{threat_name} - %{qtime}%{n}}"
```

このコマンドは、次のタイプのレコードとして隔離内容を表示します。

```
{
  <path to file>:  <threat name>  -  <date of moving to quarantine>
}
…
```

3.5.情報に関するコマンド

情報に関するコマンドには以下のものがあります。

コマンド	説明
`appinfo`	**機能**：アクティブなDr.Web for UNIX Mail Serversコンポーネントに関する情報を出力します。 現在実行中の各コンポーネントに関する以下の情報が表示されます。 • 内部で使用される名前 • プロセス識別子GNU/Linux（PID） • 状態（実行中、停止など） • コンポーネントの動作がエラーによって終了した場合、エラーコード • 追加情報（任意） 設定デーモン（`drweb-configd`）については、以下の追加情報が表示されます。 • インストールされたコンポーネントのリスト - *Installed* • 設定デーモンによって起動する必要のあるコンポーネントのリスト - *Should run* **引数**：なし **オプション** `-f [--Follow]` - モジュールのステータス変更に関する新しい情報を待ち、それらを受け取り次第メッセージを表示します（CTRL+Cで待機を中断します）。
`baseinfo`	**機能**：スキャンエンジンの現在のバージョン、およびウイルスデータベースのステータスに関する情報を表示します。 以下の情報が表示されます。 • スキャンエンジンのバージョン • 現在使用されているウイルスデータベースがリリースされた日時 • （ウイルスデータベース内の）使用可能なウイルスレコードの数 • ウイルスデータベースおよびスキャンエンジンが最後に更新された時間 • スケジュールされている次回の自動更新の時間

コマンド	説明
	引数：なし オプション `-l [--List]` - ダウンロードされたウイルスデータベースのファイルと各ファイルのウイルスレコード数の全リストを表示します。
`certificate`	機能：Dr.Web for UNIX Mail Serversによって使用される、信頼できるDr.Web証明書のコンテンツを表示します。証明書を*<cert_name>*.pemファイルに保存するには、以下のコマンドを使用できます。 `$ drweb-ctl certificate > <cert_name>.pem` 引数：なし オプション：なし
`events`	機能：Dr.Web for UNIX Mail Serversイベントを表示します。その他、このコマンドを使用してイベントを管理（既読としてマーク、削除）できます。 引数：なし オプション `--Report` *<type>* - イベントレポートのタイプを指定します。 使用可能な値： • `BRIEF` - 短いレポート • `DEBUG` - 詳細なレポート • `JSON` - JSON形式のシリアル化されたレポート `-f [--Follow]` - 新しいイベントを待ち、発生時にそれらを表示します（CTRL+Cでスタンバイを中断します）。 `-s [--Since]` *<date, time>* - 指定されたタイムスタンプの前に発生したイベントを表示します（*<date, time>*は`YYYY-MM-DD hh:mm:ss`のフォーマットで指定します）。 `-u [--Until]` *<date, time>* - 指定されたタイムスタンプよりも前に発生したイベントを表示します（*<date, time>*は`YYYY-MM-DD hh:mm:ss`のフォーマットで指定します）。 `-t [--Types]` *<type list>* - 指定されたタイプのイベントのみを表示します（コンマで区切られます）。 次のイベントタイプを使用できます。 • `Mail` - メール内で脅威を検出したことを示します。 • `UnexpectedAppTermination` - コンポーネントの予期しないシャットダウン。 すべてのタイプのイベントを表示するには、`All`を使用します。 `--ShowSeen` - 既読イベントも表示します。 `--Show` *<list of events>* - リストアップされたイベントを表示します（イベント識別子はコンマで区切られます）。

コマンド	説明
	`--Delete` *<list of events>* - リストアップされたイベントを削除します（イベント識別子はコンマで区切られます）。 `--MarkAsSeen` *<list of events>* - リストアップされたイベントを既読としてマークします（イベント識別子はコンマで区切られます）。 すべてのイベントを「既読」としてマークする場合や削除する場合は、*<events list>*ではなく`All`を指定します。例： `$ drweb-ctl events --MarkAsSeen All` このコマンドはすべてのイベントを「既読」としてマークします。
`report` *<type>*	機能：Dr.Web for UNIX Mail Serversのイベントに関するレポートをHTML形式で作成します（ページ本文は指定したファイルに出力されます）。 引数 *<type>* - レポートを作成するイベントのタイプです（タイプを1つ指定します）。可能な値については、上記`events`コマンドの`--Types`オプションの説明を参照してください。この引数は必須です。 オプション `-o [--Output]` *<path to file>* - 指定したファイルにレポートを保存します。このオプションは必須です。 `-s [--Since]` *<date, time>* - 指定されたタイムスタンプよりも後に発生したイベントのレポートを作成します（*<date, time>*は`YYYY-MM-DD hh:mm:ss`のフォーマットで指定します）。 `-u [--Until]` *<date, time>* - 指定されたタイムスタンプよりも前に発生したイベントのレポートを作成します（*<date, time>*は`YYYY-MM-DD hh:mm:ss`のフォーマットで指定します）。 `--TemplateDir` *<path to directory>* - HTMLレポートテンプレートを含むディレクトリへのパスです。 `-s`、`-u`、`--TemplateDir`は必須のオプションではありません。 `$ drweb-ctl report Mail -o report.html` たとえば、上記のコマンドは、メールメッセージでのすべての脅威検出イベントに関するレポートをデフォルトのテンプレートで生成し、結果をカレントディレクトリの`report.html`ファイルに保存します。
`idpass` *<identifier>*	機能：指定された識別子を持つメールメッセージのスキャンを実行するコンポーネントDr.Web MailDによって生成され、メールメッセージから削除された脅威を含むアーカイブの保護に使用されるパスワードを表示します（コンポーネント設定で`RepackPassword`パラメータが`HMAC(`*<secret>*`)`に設定されている場合）。 引数 *<identifier>* - メールメッセージの識別子です。

コマンド	説明
	オプション `-s [--Secret]` *<secret>* - アーカイブパスワードの生成に使用するシークレットワードです。 コマンドの呼び出し時にシークレットワードが指定されていない場合、現在のシークレットワード *<secret>* が使用されます。このシークレットワードはDr.Web MailDの設定で指定します。また、`RepackPassword` パラメータが使用できない場合、または `HMAC(<secret>)` とは異なる値に設定されている場合、コマンドはエラーを返します。
`license`	**機能**：現在有効なライセンスに関する情報を表示するか、デモバージョンのライセンスを取得するか、またはすでに登録されているライセンス（すでにWebサイト上で登録されているものなど）のキーファイルを取得します。 オプションが指定されていない場合は以下の情報が表示されます（スタンドアロンモードのライセンスを使用している場合）。 • ライセンス番号 • ライセンスの有効期間が満了する日時 集中管理サーバーから受け取ったライセンスを使用している場合（集中管理モードまたはモバイルモードで製品を使用するため）、該当するメッセージが表示されます。 **引数**：なし **オプション** `--GetRegistered` *<serial number>* - 新しいキーファイルの提供に関する条件に違反（ライセンスが集中管理サーバーによって管理される場合に製品を集中管理モードで使用していないなど）していない場合、指定されたシリアル番号に対するライセンスキーファイルを取得します。 シリアル番号が試用期間用のものではない場合、まずDoctor WebのWebサイトでそれを登録する必要があります。 `--Proxy http://<username>:<password>@<server address>:<port number>` - プロキシサーバー経由でライセンスキーファイルを取得します（`--GetRegistered` オプションとのみ使用できます）。 Dr.Web製品のライセンスに関する詳細については、ライセンスのセクションを参照してください。 シリアル番号を登録するにはインターネット接続が必要です。
`log`	**機能**：Dr.Web for UNIX Mail Serversの最新のログレコードをコンソール画面に（*stdout* スレッドで）表示します（`tail` コマンドと同様）。 **引数**：なし **オプション** `-s [--Size]` *<number>* - 画面に表示される最後のログレコードの数。 `-c [--Components]` *<components list>* - そのレコードを表示する必要があるコンポーネントのIDのリストです。IDはコンマで区切って指定します。この引

コマンド	説明
	数が指定されていない場合、あらゆるコンポーネントによってログに記録されたすべてのレコードが表示されます。 インストールされているコンポーネントの実際のID(ログに表示される内部コンポーネント名など)は、appinfoコマンドを使用して指定できます(上記を参照)。 -f [--Follow] - ログ内の新しいメッセージを待ち、それらを受け取り次第メッセージを表示します(CTRL+Cキーを押して待機を中断します)。
stat	機能:ファイルを処理するコンポーネント、またはネットワークデータのスキャンエージェントDr.Web Network Checkerの動作に関する統計情報を出力します(CTRL+CまたはQを押すと統計情報の表示を中断します)。 出力される統計情報には以下が含まれます。 • スキャンを開始したコンポーネントの名前 • コンポーネントのPID • 最後の1分間、5分間、15分間に処理された1秒あたりの平均ファイル数 • スキャンされたファイルキャッシュの使用率 • 1秒あたりのスキャンエラーの平均数 分散スキャンエージェントについては、以下の情報が出力されます。 • スキャンを開始したローカルクライアントのリスト • スキャンのためにファイルを受信したリモートホストのリスト • スキャンのためにファイルを送信したリモートホストのリスト 分散スキャンエージェントのローカルクライアントについてはPIDと名前が、リモートクライアントについてはホストのアドレスとポートが出力されます。 ローカルとリモートの両方のクライアントについて、次の情報が出力されます。 • 1秒間にスキャンされたファイルの平均数 • 1秒あたりの送信および受信バイトの平均数 • 1秒あたりのエラーの平均数 引数:なし オプション -n [--netcheck] - ネットワークデータのスキャンエージェントの動作に関する統計情報を出力します。

使用例

このセクションでは、Dr.Web Ctl(drweb-ctl)ユーティリティの使用例を示します。

- オブジェクトのスキャン:
  - シンプルなスキャンのコマンド
  - 条件によって選択されたファイルのスキャン
  - 追加のオブジェクトのスキャン
- 設定の管理
- 脅威の管理

- 自律コピーモードでの動作例

1. オブジェクトのスキャン

1.1. シンプルなスキャンのコマンド

1. デフォルトのパラメータで/homeディレクトリのスキャンを実行する:

```
$ drweb-ctl scan /home
```

2. daily_scanファイルに含まれているパスをスキャンする(1行につき1つのパス):

```
$ drweb-ctl scan --stdin < daily_scan
```

3. *sda*ドライブ上のブートレコードのスキャンを実行する:

```
$ drweb-ctl bootscan /dev/sda
```

4. 実行中のプロセスのスキャンを実行する:

```
$ drweb-ctl procscan
```

1.2. 条件によって選択されたファイルのスキャン

以下は、スキャンの対象となるファイルの選択と、findユーティリティの操作結果の使用例です。取得したファイルのリストは、--stdinまたは--stdin0パラメータを指定してdrweb-ctl scanコマンドに送信されます。

1. findユーティリティによって返されたリストに含まれ、NUL(\0)記号で区切られたファイルをスキャンする:

```
$ find -print0 | drweb-ctl scan --stdin0
```

2. ファイルシステムの1つのパーティション上の、ルートディレクトリから始まり、すべてのディレクトリ内にあるすべてのファイルをスキャンする:

```
$ find / -xdev -type f | drweb-ctl scan --stdin
```

3. /var/log/messagesおよび/var/log/syslogファイルを除いて、ルートディレクトリから始まり、すべてのディレクトリ内にあるすべてのファイルをスキャンする:

```
$ find / -type f ! -path /var/log/messages ! -path /var/log/syslog |
drweb-ctl scan --stdin
```

4. ルートディレクトリから始まり、すべてのディレクトリ内にある*root*ユーザーのすべてのファイルをスキャンする:

```
$ find / -type f -user root | drweb-ctl scan --stdin
```

5. ルートディレクトリから始まり、すべてのディレクトリ内にある*root*および*admin*ユーザーのファイルをスキャンする:

```
$ find / -type f \( -user root -o -user admin \) | drweb-ctl scan --stdin
```

6. ルートディレクトリから始まり、すべてのディレクトリ内にある、UIDが1000～1005の範囲内にあるユーザーのファイルをスキャンする:

```
$ find / -type f -uid +999 -uid -1006 | drweb-ctl scan --stdin
```

7. ルートディレクトリから始まり、ネスティングレベルが5以下のすべてのディレクトリ内にあるファイルをスキャンする:

```
$ find / -maxdepth 5 -type f | drweb-ctl scan --stdin
```

8. サブディレクトリ内のファイルを無視して、ルートディレクトリ内にあるファイルをスキャンする:

```
$ find / -maxdepth 1 -type f | drweb-ctl scan --stdin
```

9. ルートディレクトリから始まり、すべてのディレクトリ内にあるファイルをすべてのシンボリックリンクをたどりながらスキャンする:

```
$ find -L / -type f | drweb-ctl scan --stdin
```

10. ルートディレクトリから始まり、すべてのディレクトリ内にあるファイルをシンボリックリンクをたどらずにスキャンする:

```
$ find -P / -type f | drweb-ctl scan --stdin
```

11. ルートディレクトリから始まり、すべてのディレクトリ内にある2017年5月1日以前に作成されたファイルをスキャンする:

```
$ find / -type f -newermt 2017-05-01 | drweb-ctl scan --stdin
```

1.3. 追加のオブジェクトのスキャン

1. リモートホスト*192.168.0.1*上の`/tmp`ディレクトリ内にあるオブジェクトを、*user*ユーザーとしてパスワード*passw*を使用してSSH経由でそれらに接続することでスキャンする:

```
$ drweb-ctl remotescan 192.168.0.1 /tmp --Login user --Password passw
```

2. `email.eml`ファイル内に保存されたメールメッセージを、デフォルトのルールセットを使用してスキャンする:

```
$ drweb-ctl checkmail email.eml
```

2. 設定の管理

1. 実行中のコンポーネントに関する情報を含む、現在のDr.Web for UNIX Mail Serversパッケージに関する情報を表示する:

```
$ drweb-ctl appinfo
```

2. アクティブな設定の`[Root]`セクションからすべてのパラメータを出力する:

```
$ drweb-ctl cfshow Root
```

3. アクティブな設定の[LinuxSpider]セクション内でStartパラメータに「No」を設定する（これによりファイルシステムモニターSpIDer Guardが無効になります）：

```
# drweb-ctl cfset LinuxSpider.Start No
```

このアクションを実行するにはスーパーユーザー権限が必要です。権限を昇格させるには、以下の例のようにsudoコマンドを使用できます。

```
$ sudo drweb-ctl cfset LinuxSpider.Start No
```

4. Dr.Web for UNIX Mail Serversのアンチウイルスコンポーネントを強制的に更新する：

```
$ drweb-ctl update
```

5. Dr.Web for UNIX Mail Serversのコンポーネント設定を再起動する：

```
# drweb-ctl reload
```

このアクションを実行するにはスーパーユーザー権限が必要です。権限を昇格させるには、以下の例のようにsudoコマンドを使用できます。

```
$ sudo drweb-ctl reload
```

6. サーバー証明書が/home/user/cscert.pemファイルである場合に、ホスト192.168.0.1で動作している集中管理サーバーにDr.Web for UNIX Mail Serversを接続する：

```
$ drweb-ctl esconnect 192.168.0.1 --Certificate /home/user/cscert.pem
```

7. settings.cfg設定ファイルを使用して、Dr.Web for UNIX Mail Serversを集中管理サーバーに接続する：

```
$ drweb-ctl esconnect --cfg <path to the settings.cfg file>
```

8. Dr.Web for UNIX Mail Serversを集中管理サーバーから切断する：

```
# drweb-ctl esdisconnect
```

このアクションを実行するにはスーパーユーザー権限が必要です。権限を昇格させるには、以下の例のようにsudoコマンドを使用できます。

```
$ sudo drweb-ctl esdisconnect
```

9. drweb-updateコンポーネントとdrweb-configdコンポーネントによってDr.Web for UNIX Mail Serversのログに作成された最後のログレコードを表示する：

```
# drweb-ctl log -c Update,ConfigD
```

3. 脅威の管理

1. 検出された脅威に関する情報を表示します。

```
$ drweb-ctl threats
```

2. 駆除されていない脅威を含むファイルをすべて隔離へ移動します。

```
$ drweb-ctl threats --Quarantine All
```

3. 隔離へ移されたファイルのリストを表示します。

```
$ drweb-ctl quarantine
```

4. 隔離からすべてのファイルを復元します。

```
$ drweb-ctl quarantine --Restore All
```

5. このメールメッセージに対して、パスワード生成にHMAC方法が使用されており、最新の秘密ワードがDr.Web MailDの設定で示されていることを条件に、ID *12345*で保護されたアーカイブのパスワードをメールメッセージに生成します。

```
$ drweb-ctl idpass 12345
```

4. 自律コピーモードでの動作例

1. 自律コピーモードでファイルをスキャンし、隔離します。

```
$ drweb-ctl scan /home/user -a --OnKnownVirus=Quarantine
$ drweb-ctl quarantine -a --Delete All
```

最初のコマンドは自律コピーモードで/home/userディレクトリにあるファイルをスキャンします。既知のウイルスが含まれるファイルは隔離に移されます。2番目のコマンドは隔離コンテンツを（自律コピーモードでも）処理し、すべてのオブジェクトを削除します。

設定パラメータ

コマンドラインから製品を管理するためのDr.Web Ctlツールの場合、Dr.Web for UNIX Mail Serversの統合された設定ファイルには、パラメータを含む独自のセクションはありません。設定ファイルの[Root]セクションに指定されているパラメータを使用します。

Dr.Web 管理Webインターフェース

このセクションの内容:

- 機能
- コンポーネントを管理する
- 脅威の管理
- 設定を管理する
- ローカルファイルをスキャンする
- メールアーカイブのパスワードを復元する

機能

Dr.Web for UNIX Mail ServersのWebインターフェースでは、以下の操作が可能です。

1. Dr.Web for UNIX Mail Serversコンポーネントの現在の状態を表示し、一部のコンポーネントを起動または停止します。
2. 更新のステータスを表示し、必要に応じて手動で更新プロセスを開始します。
3. 製品ライセンスのステータスを表示し、必要に応じてライセンスキーを読み込みます。
4. 検出された脅威のリストを表示し、隔離されたオブジェクトを管理します(Dr.Web File Checkerコンポーネントを使用してローカルファイルシステムで検出された脅威のみが表示されます)。
5. Dr.Web for UNIX Mail Serversに含まれるコンポーネントの設定を編集します。
6. Dr.Web for UNIX Mail Serversを集中管理サーバーに接続したり、スタンドアロンモードに切り替えたりします。
7. ローカルファイルのオンデマンドスキャンを開始します(ブラウザで開いたページにファイルをドラッグ&ドロップして実行する機能も含みます)。

Webインターフェースのシステム要件

以下のWebブラウザでは、Webインターフェースが正常に動作することが保証されています。

- Microsoft Internet Explorer - バージョン11以降
- Mozilla Firefox - バージョン25以降
- Google Chrome - バージョン30以降

Webインターフェースにアクセスする

Webインターフェースにアクセスするには、ブラウザのアドレスバーに次のアドレスを入力します。

```
https://<host_with_drweb>:<port>/
```

ここで、*<host_with_drweb>*は、Dr.Web for UNIX Mail ServersがDr.Web HTTPD Webインターフェースサーバーで動作するホストのIPアドレスまたは名前で、*<port>*は、Dr.Web HTTPDがリッスンしている(このホスト上の)ポートです。ローカルホスト上で動作するDr.Web for UNIX Mail Serversコンポーネントにアクセス

するには、IPアドレス127.0.0.1または名前localhostを使用します。デフォルトでは、*<port>*は4443です。

したがって、デフォルトではローカルホストのWebインターフェースにアクセスするために、ブラウザのアドレスバーに次のURLを入力します。

```
https://127.0.0.1:4443/
```

管理サーバーへの接続が確立されると、スタートページが開き、認証フォームが表示されます。管理機能にアクセスするには、Dr.Web for UNIX Mail Serversが動作しているホスト上で管理権限を持つユーザーのログインとパスワードを指定して、認証フォームに入力します。

必要に応じて、個人ユーザー証明書を使用してWebインターフェースで認証を提供できます。それを行うには:

1. 認証局証明書で署名された個人証明書を作成します。
2. 管理用のWebインターフェースに接続するために使用されるブラウザで、署名済み証明書をユーザー認証証明書としてインポートします。
3. Dr.Web HTTPDの設定(パラメータAdminSslCA)で、個人証明書に署名する認証局証明書へのパスを指定します。

Webインターフェースでの許可に個人ユーザー証明書を使用する場合、許可フォームは表示されず、ユーザーは*root*として許可されます。

必要に応じて、付録E. SSL証明書を生成するのセクションを参照してください。

メインメニュー

認証に成功すると表示されるWebインターフェースの左ペインにメインメニューがあります。そのメニューアイテムでは以下の操作を実行できます。

- **メイン** - Dr.Web for UNIX Mail Serversのインストール済みコンポーネントとそのステータスの全リストを表示するメインページを開きます。
- **脅威** - サーバー上で検出されたすべての脅威を表示するページを開きます。このセクションでは、これらの検出された脅威を管理できます(感染したオブジェクトの隔離、検出された悪意のあるオブジェクトの再スキャン、修復、削除など)。
- **設定** - サーバーにインストールされているDr.Web for UNIX Mail Serversのコンポーネント設定のページを開きます。
- **情報** - このWebインターフェースのバージョンとウイルスデータベースの状態に関する簡単な情報を表示するページを開きます。
- **ヘルプ** - Dr.Web for UNIX製品のヘルプ情報を含む新しいブラウザタブを開きます。
- **脅威を含んだ添付アーカイブのパスワード** - スパムの兆候がある不審なメールメッセージの添付ファイル、感染した添付ファイル、不審なURL付きのメールメッセージの一部を含むアーカイブのパスワードを復元するためのパネルを表示します。
- **ファイルをスキャン** - ファイルをすばやくスキャンするためのパネルを表示します。このパネルは、閉じられるまでWebインターフェースの開いているページの上部に表示されます。
- **ログアウト** - 現在のWebインターフェースセッションを終了します(ユーザーの個人証明書による認証には使用できません)。

コンポーネントを管理する

メインページでは、Dr.Web for UNIX Mail Serversに含まれるコンポーネントのリストを表示したり、それらの動作を管理したりすることができます。

リスト上にあるDr.Web for UNIX Mail Serversのコンポーネントは、脅威を監視するメインコンポーネントと、Dr.Web for UNIX Mail Serversが正常に動作するよう全体的に管理するサービスコンポーネントの2つのグループに分けられます。メインコンポーネントのリストは、ページ上部に表形式で表示されます（コンポーネントのリストは、ディストリビューション範囲によって異なります）。コンポーネントごとに、以下の情報が表示されます。

1. **名前**。表示されているコンポーネント名をクリックすると、そのコンポーネントの設定ページが開きます。
2. **状態**。コンポーネントの状態は、スイッチアイコンとコンポーネントの現在の状態に関するメモによって示されます。スイッチをクリックするだけで、コンポーネントを開始または中断できます。スイッチの状態には次のようなものがあります。

	- コンポーネントが無効になっているため、使用されていません。
	- コンポーネントが有効になっており、正しく動作しています。
	- コンポーネントは有効になっていますが、エラーにより動作していません。

   コンポーネントの動作中にエラーが発生した場合は、コンポーネントの状態に関するメモではなく、エラーメッセージが表示されます。  アイコンをクリックすると、発生したエラーに関する詳細情報とエラーを解決するための推奨事項がウィンドウに表示されます。
3. **読み込み**。過去1分間、5分間、15分間に、コンポーネントによって1秒間に処理されたファイルの平均数がそれぞれ表示されます（3つの数字がスラッシュ「/」で区切られて表示されます）。
4. **エラー**。過去1分間、5分間、15分間にコンポーネントで1秒間に検出されたエラーの平均数がそれぞれ表示されます（3つの数字がスラッシュ「/」で区切られて表示されます）。

ツールチップを表示するには、  アイコンの上にカーソルを置きます。

メインコンポーネントに関する情報がまとめられた表の下に、Dr.Web for UNIX Mail Serversのサービスコンポーネント（スキャンエンジン、ファイルスキャンコンポーネントなど）がタイル表示されます。サービスコンポーネントごとに、その状態と動作統計も表示されます。これらのコンポーネントの設定ページを開くには、目的のコンポーネントの名前をクリックします。通常、これらのコンポーネントは必要に応じて自動的に起動または停止されます。ユーザーによって手動で起動および停止される可能性があるサービスコンポーネントについては、名前と動作統計の他に、コンポーネントを起動および停止するためのスイッチも、該当するサービスコンポーネントのタイルに表示されます。

ページの下部には、ウイルスデータベースが最新かどうかという情報と、ライセンス情報が表示されます。ウイルスデータベースを強制的に更新するには、**更新**をクリックします。**更新**ボタン（またはライセンスの現在の状態によっては**ライセンスを有効化**ボタン）をクリックすると、Dr.Web for UNIX Mail Serversに対して有効なキーファイルをライセンスサーバーにアップロードしてライセンスを更新または有効化できます。

脅威の管理

脅威ページで、検出された脅威の一覧を表示し、それらに対する対応を管理できます。

このページには、ファイルシステムを監視およびスキャンするDr.Web for UNIX Mail Serversのコンポーネントによって検出された脅威のすべてのリストが含まれています。ページ上部には、カテゴリー別に脅威にフィルターを適用できるメニューがあります。

- **全て** - 検出されたすべての脅威（アクティブな脅威と隔離された脅威の両方を含む）を表示します。
- **アクティブ** - アクティブな（検出されたがまだ駆除されていない）脅威のみを表示します。
- **ブロック済** - ブロックされているすべての脅威、つまり駆除されていないが、それを含む感染オブジェクトがブロックされている脅威をすべて表示します。
- **隔離済** - 隔離に移された脅威を表示します。
- **エラー** - エラーのために処理されなかった脅威を示します。

上部メニューの脅威カテゴリーの名称の隣（右側）には、このカテゴリーに分類される検出済みの脅威の数が表示されます。そのカテゴリーに属する脅威が現在表示され、選択されているカテゴリーは、暗い色のフォントで強調されています。必要なカテゴリーの脅威を表示するには、メニューのカテゴリーの名前をクリックしてください。

ネットワークトラフィックをスキャンするコンポーネント（SpIDer Gate、Dr.Web MailD）、およびDr.Web ClamDによって検出された脅威は**脅威**ページに表示されません。これらのコンポーネントによって検出された脅威を追跡するために、SNMPを介して使用可能な脅威カウンターと追跡通知を制御できます（Dr.Web SNMPDはMIB Dr.Web構造に従って脅威カウンターと通知へのアクセスを付与します）。

脅威ごとに次の情報が一覧表示されています。

- **ファイル** - 悪意のあるオブジェクトを含むファイルの名前（ファイルパスは指定されていません）。
- **所有者** - 感染ファイルを所有するユーザーの名前。
- **コンポーネント** - 脅威を検出したDr.Web for UNIX Mail Serversのコンポーネントの名前。
- **脅威** -（Doctor Web社の使用する分類に従い）ファイルで検出された脅威の名前。

リストで選択されているオブジェクトについては、以下の情報が表示されます。

- 脅威の名前（Dr.Webウイルス情報ライブラリのページを開くと、その脅威の説明が表示されます）。
- ファイルのサイズ、Byte単位。
- 脅威を検出したコンポーネントの名前。
- 脅威が検出された日時。
- 脅威が最後に変更された日時。
- 感染ファイルを所有しているユーザーの名前。
- ファイルの所有者を含むグループの名前。
- 脅威を含む隔離ファイルに割り当てられた識別子（ファイルが隔離された場合）。
- ファイルの元の場所（脅威の検出時にファイルが存在していた場所）を指すフルパス。

リスト内の任意のオブジェクトをクリックして選択できます。複数のオブジェクトを選択するには、対応するオブジェクトのチェックボックスを選択します。すべてのオブジェクトを選択するか選択をキャンセルするには、脅威リストのヘッダーの**ファイル**フィールドのチェックボックスをオンにします。

リストで選択したオブジェクトにアクションを適用するには、脅威リストの真上にあるツールバーの対応するボタンをクリックします。ツールバーには、次のボタンがあります（選択した脅威の種類によっては、使用できないボタンがあります）。

	- 選択したファイルを（永久に）削除します。
	- 選択したファイルを隔離から元の場所に復元します。
	- 選択したファイルに追加のアクションを適用します（使用可能なアクションはドロップダウンリストで指定します）。 • **隔離** - 脅威を含む選択したファイルを隔離します。 • **修復** - 脅威の修復を試みます。 • **無視** - 選択したファイルで検出された脅威を無視し、リストからその脅威を削除します。

検索クエリーに基づいて表示された脅威にフィルターを適用することもできます。フィルターを適用して不要な脅威を除外してクエリーに対応するものだけを表示するには、検索ボックスを使用します。検索ボックスはツールバーの右側に  マークと一緒に表示されています。脅威リストにフィルターを適用するには、検索ボックスに単語を入力します。名前や説明に入力した単語が含まれていない脅威はすべて非表示になります（このフィルターの適用では大文字と小文字が区別されません）。検索結果を消去してフィルターを適用していないリストを表示するには、検索ボックスの  をクリックするか、単語を消去します。

設定を管理する

Dr.Web for UNIX Mail Servers に含まれ、メインページに一覧表示されているコンポーネントの現在の設定パラメータを表示、変更できます。そのためには、**設定**ページを開きます。このページでは、Dr.Web for UNIX Mail Servers を*集中管理*モードまたは*スタンドアロン*モードに切り替えることもできます（これらのモードの詳細については、動作モードを参照してください）。

ページの左側に表示されるメニューには、設定を表示、調整できるすべてのDr.Web for UNIX Mail Servers のコンポーネント名が含まれています。任意のコンポーネントの設定を表示、調整するには、まずこのメニューで該当するコンポーネント名をクリックします。現在、設定の表示および編集を行っているコンポーネントの名前は左側のメニューに強調表示されます。

- メニューの**集中管理**の項目を選択すると、集中管理モードを管理するページに移動します。
- メニューの**全般設定**の項目は、Dr.Web ConfigDコンポーネントの設定に対応しており、Dr.Web for UNIX Mail Servers の全体的な機能を担っています。

コンポーネントにメイン設定のセクションとは別の追加設定のセクションがある場合（たとえば、ClamAV®アンチウイルスのインターフェースをエミュレートし、異なる接続アドレスを使用するクライアントごとに個別のスキャンパラメータを保持するためにそれらの追加セクションを使用する、Dr.Web ClamDコンポーネントで使用できるセクション）、追加セクションの展開や折りたたみができることを示すアイコンがコンポーネント名の左側に表示されます。このアイコンが▸の場合、追加セクションは非表示になります。アイコンが▾の場合は、追加セクションが1行に1つずつメニューに表示されます。追加セクションのリストを展開したり折りたたんだりするには、目的のコンポーネントの名前の横にある展開・折りたたみアイコンをクリックします。

- 設定を含む追加のセクションは、インデントラインとして表示されます。追加セクションのパラメータを表示または編集するには、その名前をクリックします。

- コンポーネントの設定を含む下位セクションの追加が許可されている場合、コンポーネント名の右側にある＋をクリックして追加します。次に、新しいサブセクションに一意の名前（タグ）を指定して、**OK**をクリックします。サブセクションを作成せずにウィンドウを閉じるには、**キャンセル**をクリックします。
- コンポーネントのサブセクションを削除するには、コンポーネント名にカーソルを合わせると表示されるサブセクション名（タグ）の右側の✕をクリックします。次に、サブセクションを削除することを確認して**はい**をクリックするか、**いいえ**をクリックしてサブセクションを削除せずにウィンドウを閉じます。

設定ページの上部には、表示モードを変更できるメニューがあります。以下のモードが利用可能です。

- **全て** - 表示および調整可能なすべてのコンポーネントの設定パラメータを含むテーブルを表示します。
- **変更** - デフォルトの値とは異なる値を持つコンポーネントの設定パラメータを含むテーブルを表示します。
- **Ini エディタ** - テキストエディターを表示し、デフォルトの値とは異なる値を持つこのコンポーネントの設定パラメータを示します。表示されるテキストは、設定ファイルと同じ形式です（`parameter = value`のペアを含みます）。

検索クエリーに基づいて表示されたパラメータにフィルターを適用することもできます。フィルターを適用して不要なパラメータを除外し、クエリーに対応するものだけを表示するには、検索ボックスを使用します。検索ボックスは表示モードメニューの右側に🔍マークと一緒に表示されています。パラメータリストにフィルターを適用するには、検索ボックスに任意の単語を入力します。説明に入力した単語が含まれていないパラメータはすべて非表示になります（このフィルターの適用では大文字と小文字が区別されません）。検索結果を消去してフィルターを適用していないリストを表示するには、検索ボックスの⊗をクリックするか、その中の単語を消去します。

パラメータが表形式で表示されている場合（つまり、**全て**表示モードと**変更**表示モード）にのみ、フィルターを適用してパラメータを除外できます。

表形式でコンポーネント設定を表示、編集する

表形式でパラメータを表示する場合（**全て**表示モードと**変更**表示モード）、各表の行にはパラメータの名前と説明（左側）およびその現在値（右側）が含まれます。Boolean値パラメータ（使用可能な値が2つのみの場合、「はい」と「いいえ」）では、値の代わりにチェックボックスが表示されます（チェックを入れると「はい」を、チェックしないと「いいえ」を意味します）。

> （変更されたものだけでなく）すべてのパラメータを表示するように選択した場合、変更された（デフォルト以外の）値は太字で示されます。

完全なパラメータリストはグループに分割されます（**全般**、**アドバンス**など）。グループを折りたたむまたは展開するには、その見出し（名前）をクリックします。グループが折りたたまれていて、そのパラメータが表に表示されていない場合は、グループ名の左側に›のアイコンが表示されます。グループが展開されてテーブルにパラメータが表示されると、⌄のアイコンがグループ名の左側に表示されます。

パラメータを調整するには、表内の現在の値をクリックします（Boolean値パラメータの場合は、対応するチェックボックスのチェックマークを設定または削除します）。パラメータに一連の定義済みの値がある場合は、現在の値をクリックした後にすべてドロップダウンリストとして表示されます。パラメータに数値がある場合は、現在の値をクリックした後に編集ボックスが表示されます。必要な値を指定して、ENTERを押してください。以下の図は、パラメータ値を変更する方法の例を示しています（図に示されているコンポーネントのセットは、提供されているものとは異なる可能性があることに注意してください）。パラメータ値に対するすべての変更は、対応するコンポーネントの設定にすぐに適用されます。

図3. 表形式のコンポーネント設定

パラメータがその値として文字列を要求しているか、任意の値のリストを受け付ける場合は、パラメータの現在の値をクリックして編集すると、ポップアップウィンドウが表示されます。パラメータが値のリストを受け付ける場合は、次の図に示すように複数行の編集ボックスに表示されます（1行に1つの値）。一覧表示されている値を編集するには、編集ボックスで必要な行を変更、削除、または追加する必要があります。

図4. 値リストの編集

パラメータの値を編集したら、**保存**をクリックして変更内容を適用し、ウィンドウを閉じます。変更を適用せずにウィンドウを閉じるには、**キャンセル**をクリックするか、ポップアップウィンドウの右上隅にある×のアイコンをクリックします。

テキストエディターでコンポーネントの設定を表示、編集する

Ini エディタモードでパラメータを表示すると、それらは製品の設定ファイルと同じ形式（`parameter = value`のペア）で表示されます。ここでのパラメータは、設定ファイル（対応するコンポーネントの設定セクション）に直接書き込まれるパラメータの名前です。このモードでは、デフォルト値とは異なる値のパ

ラメータのみが表示されます（つまり、全て表示モードで値が太字で強調されているパラメータ）。以下の図は、このシンプルビューのテキストエディターでパラメータがどのように表示されるかを示しています。

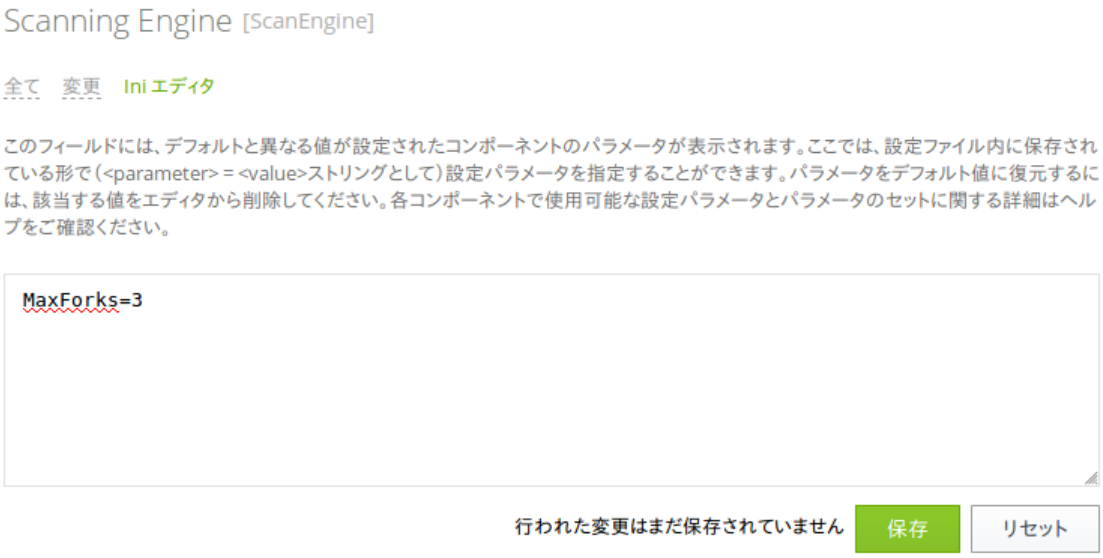

図5. 設定用の組み込みエディター

必要な変更を加えるには、設定ファイルの編集について説明したのと同じ規則に従って、このテキストエディターでテキストを編集します（これにより、左側で強調表示されているコンポーネントの設定を含むセクションのみが変更されます）。必要に応じて、コンポーネントで使用可能な任意のパラメータに新しい値を指定できます。この場合、このパラメータの値はデフォルト設定からエディターに入力した値に変わります。パラメータをデフォルト値にリセットする場合は、このテキストエディターでこのパラメータを含む行を消去してください。変更した場合は、変更を保存するとパラメータはデフォルト値に戻ります。

パラメータ値の編集が終了したら、**保存**をクリックして変更を適用するか、**キャンセル**をクリックして変更をキャンセルします。

保存をクリックすると、テキストが検証されます。プログラムは、すべてのパラメータが存在し、それらの設定値が有効であるかどうかを確認します。エラーが発生した場合は、適切なメッセージが表示されます。

パラメータ値を指定するために重要な設定ファイルとその機能の詳細については、付録D. Dr.Web for UNIX Mail Servers設定ファイルのセクションを参照してください。

追加情報

- Dr.Web ConfigDの設定パラメータ（共通設定）
- SpIDer Gateの設定パラメータ
- Dr.Web Firewall for Linuxの設定パラメータ
- Dr.Web MailDコンポーネントの設定パラメータ
- Dr.Web ES Agentの設定パラメータ
- Dr.Web Updaterの設定パラメータ
- Dr.Web ClamDの設定パラメータ

- Dr.Web File Checkerの設定パラメータ
- Dr.Web Scanning Engineの設定パラメータ
- Dr.Web Network Checkerの設定パラメータ
- Dr.Web SNMPDの設定パラメータ
- Dr.Web CloudDの設定パラメータ
- Dr.Web LookupDの設定パラメータ
- Dr.Web StatDの設定パラメータ
- 集中管理モードの管理

集中管理モードの管理

Dr.Web for UNIX Mail Serversは、集中管理サーバーに接続したり、スタンドアロンモードに戻して、製品を集中管理サーバーから切断したりすることができます。集中管理モードを管理できるページを開くには、**設定**ページの設定メニューから**集中管理**という項目を選択します。

Dr.Web for UNIX Mail Serversを集中管理サーバーに接続したり、集中管理サーバーとの接続を切断したりするには、このページの該当するチェックボックスを使用します。

集中管理サーバーとの接続

集中管理サーバーへの接続を試みると、画面にポップアップウィンドウが表示されます。このウィンドウでは、集中管理サーバーに接続するためのパラメータを指定する必要があります。

図6. 集中管理サーバーとの接続

ウィンドウ上部のドロップダウンリストから、集中管理サーバーとの接続方法を1つ選択します。3つの方法があります。

- *ファイルから読み込む*
- *手動で設定*
- *自動で検出*

*ファイルから読み込む*オプションを選択した場合は、このウィンドウの該当するフィールドで、接続設定を含むファイルへのパスも指定する必要があります。このファイルはアンチウイルスネットワーク管理者によって提供されます。*手動で設定*オプションを選択した場合は、集中管理サーバーのアドレスとポートを指定する必要があります。*手動で設定*または*自動で検出*オプションでは、サーバーのパブリックキー(ネットワーク管理者またはインターネットサービスプロバイダーによって提供されたもの)を含むファイルへのパスを指定することもできます。

また、集中管理サーバーでの認証用のワークステーション識別子(ID)とパスワードを知っている場合は、**認証(任意)**セクションでそれらを指定できます。これらのフィールドに入力すると、集中管理サーバーへの接続は、正しいIDとパスワードのペアが入力された場合にのみ成功します。これらのフィールドを空白のままにすると、集中管理サーバーへの接続は、集中管理サーバーによって(サーバーの設定に応じて自動的に、またはアンチウイルスネットワーク管理者によって)承認された場合にのみ確立します。

さらに、**ワークステーションを「新規端末」として接続**オプションを使用することもできます(新規ユーザーとして接続する場合)。この場合、ワークステーションからの接続に対して集中管理サーバーで新規端末モードが許可されていると、集中管理サーバーはこの接続を承認した後、自動的に一意のIDとパスワードのペアを生成します。その後、コンピューターをサーバーに接続する際はこのペアが使用されます。このモードでは、すでにサーバー上にワークステーションの別のアカウントがある場合でも、集中管理サーバーはワークステーション用に新しいアカウントを生成します。

接続パラメータはアンチウイルスネットワーク管理者またはサービスプロバイダーによって提供された指示に厳密に従って指定してください。

集中管理サーバーに接続するには、すべてのパラメータを指定し、**接続**をクリックして、接続が確立されるまで待ちます。サーバー接続を確立せずにウィンドウを閉じるには、**キャンセル**をクリックします。

Dr.Web for UNIX Mail Serversを集中管理サーバーに接続すると、スタンドアロンモードに戻すまで、その動作は集中管理サーバーによって管理されます。Dr.Web for UNIX Mail Serversが起動されるたびに、集中管理サーバーへの接続が自動的に確立されます。

ローカルファイルをスキャンする

Webインターフェースには、ローカルコンピューター(現在Webインターフェースにアクセスしているコンピューター)に保存されているファイルをスキャンして、ファイルに悪意のあるコンテンツが含まれているかどうか確認する機能があります。スキャンにはDr.Web for UNIX Mail Serversに含まれているスキャンエンジンが使用されます。スキャン対象として選択されたファイルはDr.Web for UNIX Mail Serversが動作しているサーバーに(HTTPプロトコル経由で)アップロードされますが、脅威が見つかった場合でも、スキャン後にファイルがサーバーに保存されることも、隔離されることもありません。スキャンするファイルを送信したユーザーには、スキャンの結果についてのみ通知されます。

この機能は、Dr.Web for UNIX Mail Serversディストリビューションに Dr.Web Network Checkerコンポーネントが含まれている場合にのみ使用できます。

ローカルファイルをスキャンするパネルを開いて、スキャン用のパラメータを設定する

Webインターフェースのメインメニューで**ファイルをスキャン**項目を選択したときに表示されるローカルファイルのスキャンパネルを使用して、スキャンするファイルを選択してアップロードできます。起動したパネルは、Webインターフェースの右下隅に表示されます。ローカルファイルのスキャンパネルは以下のようになります。

図7. ローカルファイルのスキャンパネル

このパネルを閉じるには、パネルの右上隅にある ✕ をクリックします。⚙アイコンをクリックすると、ローカルファイルのスキャン設定を表示できます。これには、ファイルをスキャンする最大時間（ローカルコンピューターからサーバーにファイルをアップロードするのにかかる時間は含まれません）、ヒューリスティック解析の使用、圧縮されたオブジェクトの最大圧縮率、コンテナ（アーカイブなど）に圧縮されたオブジェクトの最大ネスティングレベルなどが含まれます。

図8. ローカルファイルをスキャンするためのパラメータを設定する

変更した設定を適用して、スキャンするファイルを選択できるファイル選択モードに戻るには、**適用**ボタンを押します。設定に変更を適用せずにファイル選択に戻るには、**キャンセル**ボタンを押します。

ローカルファイルのスキャンを開始する

スキャンするファイルを選択してスキャンを開始するには、**ファイルをここにドラッグするか、クリックして選択します**と表示されているターゲット領域を左クリックします。そこをクリックすると、各OSのファイルマネージャーの標準的なファイル選択ウィンドウが開きます。スキャン対象として一度に複数のファイルを選択できます。スキャン対象としてディレクトリを選択することはできませんのでご注意ください。ファイルマネージャーウィンドウで選択したファイルをマウスで直接ファイルスキャンパネルのターゲットエリアにドラッグすることもできます。スキャンするファイルを指定すると、Dr.Web for UNIX Mail Serversがインストールされているサーバーへのアップロードが開始されます。ファイルがアップロードされると、スキャンが開始されます。ファイルのアップロードおよびスキャン中に、ファイルスキャンパネルにスキャン手順の全体的な進捗状況が表示されます。

図9. ローカルファイルのスキャンの進捗状況

必要に応じて、**停止**ボタンを押してスキャンを中止できます。スキャンが完了すると、アップロードされたファイルのスキャンに関するレポートがファイルスキャンパネルに表示されます。

図10. スキャンしたローカルファイルの結果

複数のファイルがアップロードされると、スキャンに関する詳細なレポートが利用可能になります。拡張レポートを表示するには、**すべてのファイルについてレポートを見る**というリンクをクリックします。

図11. スキャンしたローカルファイルに関する詳細レポート

レポートを閉じて、パネルでスキャン対象の新しいファイルを選択できる状態に戻るには、**OK**を押します。

ファイルスキャンパネルを閉じているときでも、(スキャンの現在の設定を使用して)ローカルファイルのスキャンを開始することができます。ローカルファイルのアップロードとスキャンを開始するには、ファイルマネージャーウィンドウからブラウザで開いているWebインターフェースのページにドラッグアンドドロップします。

メールアーカイブのパスワードを復元する

Webインターフェースでは、メールユーザーが受信した脅威を含んだ、保護されたアーカイブのパスワードをすぐに復元できます。このようなアーカイブは、アクション*Pass*がメールメッセージに適用された場合に、スキャンされたメールメッセージの悪意のある部分や望ましくない部分を保存するために、Dr.Web for UNIX Mail Serversによって使用されます。設定パラメータ`RepackPassword`の値に応じて、アーカイブは以下のようになります。

- パスワードで保護されていません(*None*)。
- パラメータに示されているのと同じパスワードで保護されています(*Plain*)。
- 秘密のワードと一意のメールメッセージ識別子に基づいて各アーカイブ用に生成された一意のパスワードで保護されています(*HMAC*)。

ユーザーが一意のメールメッセージ識別子を共有し、管理者がパスワード生成に使用される秘密のワードを知っている場合、パスワード復元のためのインターフェースでは、メールシステムの管理者は*HMAC*方式で保護されたアーカイブのパスワードを(ユーザーのリクエストにより)復元できます(デフォルトでは、モード*HMAC*が設定されている場合、秘密のワードはパラメータ`RepackPassword`の値からの現在有効な秘密のワードとなります)。

パスワード生成方式が変更された場合、パスワード復号が正常に実行されるためには、脅威を含んだ保護されたアーカイブのメールメッセージの解凍とパスワード生成の実行時に使用されていた秘密のワードが必要となります。

ユーザーのメールメッセージに一意の識別子がない場合、アーカイブのパスワード生成が*Plain*方式を使用して実行されたことを意味し、パスワード復元インターフェースではパスワードを復元できません。

パスワードを復元する

脅威のある保護されたアーカイブのパスワードの復元は、Webインターフェースのメインメニューで**脅威を含んだ添付アーカイブのパスワード**項目を選択すると表示されるパネルを介して実行されます。起動したパネルは、Webインターフェースの右下隅に表示されます。パスワード復元パネルの外観は以下のようになります。

図12. メールアーカイブのパスワード復元パネル

*HMAC*方式で生成されたアーカイブのパスワードを復元するには、以下のように指示します。

- パスワードで保護されたアーカイブを含むメールメッセージの受信者によって提供された**メッセージID**。
- メール処理時にDr.Web MailDの設定で使用された**秘密のフレーズ**（デフォルトでは、パスワード生成の*HMAC*方式がDr.Web MailDの設定で指示されている場合、このフィールドには現在有効な秘密のワードを入力します）。

アーカイブのパスワードを復元するには、**パスワードを取得**をクリックします。生成されたパスワードは、**アーカイブのパスワード**フィールドに表示されます。

パネルを閉じるには、パネルの右上隅にある×をクリックします。

Dr.Web MailD

Dr.Web MailDは、メールの直接スキャン、悪意のあるコンテンツの検出（添付ファイルだけでなく望ましくないWebサイトへのリンクも含む）、およびメッセージにスパムの兆候があるかどうかに関する解析と、メッセージがメールシステム管理者によって指定されたセキュリティ基準に準拠しているかどうかに関する解析（管理者によって指定された正規表現を使用したメールメッセージの本文とヘッダーのスキャン）を行えるように設計されています。

このコンポーネントは、標準のインターフェースである*Milter*、*Spamd*、*Rspamd*（これらのインターフェースは通常、SpamAssassinフィルターで使用されます）を介してメールサーバー（MTA）に統合したり、送信側と受信側（MTAとMTA、MDAとMUA）にとって透過的にメールプロトコル（SMTP、POP3、IMAP）に統合したりできます。2番目の方法は、SpIDer Gateコンポーネントのネットワークトラフィックのスキャン機能がDr.Web MailDコンポーネントによって使用されることを意味します。外部フィルターモードでは、Dr.Web vxCubeの統合が有効な場合、Dr.Web vxCube Webサービスによってメールの添付ファイルを解析できます。

SpIDer Gateモニターは GNU/Linux環境でのみ動作するため、透過的な統合方法(「プロキシ」モード)は、GNU/Linux環境で動作するメールサーバーでのみ使用できます。

メールメッセージのスキャンの負荷が高まると、Dr.Web Network Checkerコンポーネントが利用できるファイル記述子の数が減少するため、スキャンに問題が生じる場合があります。この場合、Dr.Web for UNIX Mail Serversに利用できるファイル記述子の制限数を増やす必要があります。

動作原理

このコンポーネントは、次の3つの方法でメールを保護できます。

1. *外部のメールフィルター*として(メールサーバーでサポートされている*Milter*、*Spamd*、または*Rspamd*のいずれかの拡張機能を使用)、またはSMTP(Postfix)モードでメールサーバー(Sendmail、Postfix、Eximなど)へ接続する。
2. BCCモードでDr.Web vxCubeと統合した場合に、メール添付ファイルスキャン用のメールサーバーへ接続する(メールのブラインドカーボンコピーの送信をサポートする任意のMTAを使用)。
3. SMTP、POP3、IMAP4のプロトコルを介してメールサーバーに対して透過的に転送されたメールのスキャンを実行する、*プロキシ*を設定する。このスキャン方法の設定にはSpIDer GateとDr.Web Firewall for Linuxが使用されます。これらのコンポーネントはGNU/Linuxでしか動作しないことから、この方法はGNU/Linuxファミリーの OSでのみ使用できます。

スキャンされたメールメッセージは、コンポーネント設定で設定されているルールに従って処理されます。メールサーバーとのインタラクションに使用されるインターフェースごとに、メールメッセージ処理のための独自のルールセットを指定できます。プロキシメカニズムを使用する場合(プロトコルSMTP、POP3、IMAPを介して直接受信したメールメッセージをスキャンする際など)、コンポーネントはDr.Web Firewall for Linuxの設定で決定された処理ルールを使用します。

メールメッセージのURLのスキャンには、SpIDer Gateコンポーネントと同じWebリソースカテゴリーのデータベースが使用されます。Dr.Web CloudDコンポーネントは、Dr.Web Cloudサービスを参照するために使用されます(クラウドサービスの使用はDr.Web for UNIX Mail Serversの共通設定で設定され、必要に応じて無効化できます)。送信されたデータを確認するために、Dr.Web MailDはDr.Web Network Checkerコンポーネントを使用します。後者では、Dr.Web Scanning Engineスキャンエンジンを介してスキャンが開始されます。

Dr.Web MailDまたはDr.Web vxCubeが統合されている場合、メール添付ファイル内の悪意のあるコードの有無を確認するためのスキャンは、直接実行できます。

MTAから*Milter*、*Spamd*、*Rspamd*のインターフェースを介して(フィルターモードで)受信されたメールメッセージの処理は、Luaで記述された特別な処理手順(*hook*)を呼び出すことによって実装されます。この手順の間に、メッセージに関して利用可能なすべての情報(送信者、受信者、内部構造、ヘッダー値、スパムスコア)を分析した結果に従って、メッセージが拒否されるか通過されるかが決定されます。*Milter*インターフェースの場合、MTAがメッセージに適用するアクション(「pass」、「reject」、「return an error to sender」など)が返されます。適用されるアクションが「pass」の場合、ヘッダーの追加/変更など、メッセージに変更が加えられる場合があります。メッセージの悪意のある部分はパスワードで保護されたアーカイブに保存(つまり再圧縮)されます。スキャン対象のメッセージの変更をサポートしていない*Spamd*や*Rspamd*のインターフェースの場合、判定は、そのメッセージに割り当てられた「スパムレート」およびメッセージをスパムとして認識するためのしきい値の形でMTAに返されます(MTAからのレターを拒否するには、レートがしきい値を超過している必要があります)。レートに加えて、テキストによる判定(プロトコルによって`report`または`action`)がMTAに返されます。これはMTA設定で分析できます。

Luaの柔軟性と処理手順から得られる多数のメッセージ情報により、Dr.Web Anti-Spamから受け取ったスコアによる典型的なスパムチェックや、添付された脅威または悪意のあるURLの検索だけでなく、メールサーバーによるメールメッセージ処理に必要な判定と連携した、任意の条件のチェックも実装できます。検証手順の詳細と手順の例については、Luaでのメール処理のセクションを参照してください。

スパムの兆候の存在に関するメッセージ解析では、Dr.Web MailDは特別なコンポーネントDr.Web Anti-Spamを使用します。

ディストリビューションによっては、Dr.Web for UNIX Mail ServersでDr.Web Anti-Spamを利用できない場合があります。そのような場合、スパムの兆候に関するメールメッセージのスキャンは実行されません。

コマンドライン引数

Dr.Web MailDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-maild [<parameters>]
```

Dr.Web MailDは次のオプションを処理できます。

パラメータ	説明
`--help`	機能：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 短縮形：`-h` 引数：なし
`--version`	機能：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 短縮形：`-v` 引数：なし

例：

```
$ /opt/drweb.com/bin/drweb-maild --help
```

このコマンドは、Dr.Web MailDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。Dr.Web for UNIX Mail Serversの他のコンポーネントからメールオブジェクトスキャンのリクエストを受信するときに、Dr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理し、必要に応じてメールオブジェクトをスキャンするには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（`drweb-ctl`コマンドを使用して起動されます）。

Dr.Web MailDコンポーネントによる任意のメールメッセージの処理をスキャンするには、Dr.Web Ctlツールの`checkmail`コマンドを使用します。これを行うには、スキャンしたメールメッセージをドライブに保存して(`.eml`形式など)、次のコマンドを使用します。

```
$ drweb-ctl checkmail <path to file .eml>
```

コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-maild`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[MailD]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータ値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。 デフォルト値:`Notice`
`Log` *{log type}*	コンポーネントのロギング方式。 デフォルト値:`Auto`
`ExePath` *{path to file}*	コンポーネントの実行パス。 デフォルト値:*<opt_dir>*`/bin/drweb-maild` • GNU/Linuxの場合:`/opt/drweb.com/bin/drweb-maild` • FreeBSDの場合:`/usr/local/libexec/drweb.com/bin/drweb-maild`
`RunAsUser` *{UID / user name}*	その権限によりコンポーネントを実行するユーザーの名前。このユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合(つまりUIDに似ている場合)は、「`name:`」というプレフィックスを付けて指定します。たとえば、`RunAsUser = name:123456`です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値:`drweb`
`FixedSocketPath` *{path to file}*	固定コンポーネントコピーのUNIXソケットへのパス。 このパラメータが指定されている場合、Dr.Web ConfigD設定デーモンは、このソケットを介してクライアントが使用可能な実行中のコンポーネントのコピーが常に存在することを確認します。

パラメータ	説明
	デフォルト値：*（未設定）*
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 FixedSocketPath, MilterSocket, SpamdSocket, RspamdHttpSocket, RspamdSocket, SmtpSocket, BccSocketのうち、いずれかのパラメータの値が設定されている場合、IdleTimeLimitの設定は無視されます（指定した時間を経過しても、コンポーネントはその動作を終了しません）。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s
DnsResolverConfPath *{path to file}*	ドメイン名パーミッション（DNSリゾルバ）のサブシステム設定ファイルへのパス。 デフォルト値：/etc/resolv.conf
TemplatesDir *{path to directory}*	メールがブロックされた場合にユーザーに返されるメールのテンプレートを含むディレクトリへのパス。 デフォルト値：*<var_dir>*/templates/maild • GNU/Linuxの場合：/var/opt/drweb.com/templates/maild • FreeBSDの場合：/var/drweb.com/templates/maild
TemplateContacts *{string}*	脅威に関するメッセージに挿入されるDr.Web for UNIX Mail Serversの管理者の連絡先（メッセージテンプレートで使用されます）。 連絡先情報は、最初のメッセージから脅威やその他の望ましくないオブジェクトが削除されている、パスワードで保護されたアーカイブを含む添付ファイルを取得した場合にのみ、再圧縮されたメッセージに追加されます。RepackPasswordパラメータの現在の値（以下参照）に従って添付アーカイブがパスワードで保護されていない場合、連絡先情報は変更されたメッセージに追加されません。 デフォルト値：*（未設定）*
ReportLanguages *{string}*	サービスメールメッセージ（メールがブロックされた場合に送信者に返されるメールメッセージなど）の生成に使用される言語。各言語は2文字の表記（en、ruなど）で識別されます。 リストをパラメータ値として指定できます。リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：言語ruおよびdeをリストに追加します。 1. 設定ファイルに値を追加します。

パラメータ	説明
	• 1行に2つの値： `[MailD]` ` ReportLanguages = "ru", "de"` • 2行（1行に1つの値）： `[MailD]` ` ReportLanguages = ru` ` ReportLanguages = de` 2. `drweb-ctl cfset` コマンドを使用して値を追加します。 `# drweb-ctl cfset MailD.ReportLanguages -a ru` `# drweb-ctl cfset MailD.ReportLanguages -a de` デフォルト値：`en`
`RepackPassword` *{None / Plain(<password>) / HMAC(<secret>)}*	メッセージに添付され受信者に送信された悪意のあるオブジェクトのアーカイブのパスワードを生成する方法。以下の方法を使用できます。 • `None` - アーカイブはパスワードで保護されません（非推奨）。 • `Plain(`*<password>*`)` - すべてのアーカイブが同じパスワード *<password>* で保護されます。 • `HMAC(`*<secret>*`)` - *<secret>* と *<message identifier>* のペアに基づいてアーカイブごとに一意のパスワードが生成されます。 メッセージIDと既知のシークレットを使用してアーカイブを保護するパスワードを復元するには、`drweb-ctl idpass` コマンドを使用します。 ⚠ デフォルトでは、このパラメータには値 `None` が設定されています。これはDr.Web for UNIX Mail Serversの設定中に変更することを推奨します。 デフォルト値：`None`
`ScanTimeout` *{time interval}*	Dr.Web MailDによって開始された1つのメールメッセージのスキャンのタイムアウト。 指定可能な値：1秒（`1s`）から1時間（`1h`）まで。 デフォルト値：`3m`
`HeuristicAnalysis` *{On / Off}*	未知の脅威を検出するためのヒューリスティック解析を有効／無効にします。 ヒューリスティック解析における検出の信頼性は高いのですが、ウイルススキャンに時間がかかります。

パラメータ	説明
	使用可能な値： • On - スキャン時にヒューリスティック解析を使用します。 • Off - ヒューリスティック解析を使用しません。 デフォルト値：On
PackerMaxLevel *{integer}*	圧縮されたオブジェクトの最大ネスティングレベル。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含む他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値はネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
ArchiveMaxLevel *{integer}*	他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）の最大ネスティングレベル（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値はネスティングの上限を指定します。この上限を超えると、他のアーカイブに含まれるアーカイブはスキャンされません。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
MailMaxLevel *{integer}*	他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）の最大ネスティングレベル（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
ContainerMaxLevel *{integer}*	他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）の最大ネスティングレベル。このパラメータの値はネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
MaxSizeToExtract *{size}*	アーカイブに含まれるファイルの最大サイズ。このパラメータの値よりサイズが大きいファイルは、スキャン時にスキップされます。デフォルトでは、アーカイブ内のファイルのサイズ制限はありません。

パラメータ	説明
	このパラメータの値は、サフィックス（b、kb、mb、gb）を付けた数値で指定します。サフィックスが指定されていない場合、値はバイト単位のサイズとして扱われます。 値を0に設定すると、アーカイブ内のファイルは全くチェックされません。 デフォルト値：None
MaxCompressionRatio *{integer}*	圧縮されたオブジェクトの最大圧縮率（非圧縮サイズと圧縮サイズの比率）。この比率が制限を超えると、そのオブジェクトはDr.Web MailDによって開始されたスキャン中にスキップされます。 圧縮率には2よりも小さい値は指定できません。 デフォルト値：500
MilterSocket *{path to file / IP address:port}*	メールの*Milter*フィルターとしてMTAに接続するためのソケット（Dr.Web MailDを対応するフィルターとして使用しているときに、MTAはこのソケットに接続します）。UNIXソケットまたはネットワークソケットを使用できます。 *Milter*経由で送信されるメッセージの処理ルールは、MilterHookパラメータで指定します（以下を参照）。 デフォルト値：*（未設定）*
MilterDebugIpc *{Boolean}*	デバッグログ（LogLevel = Debug）に*Milter*プロトコルメッセージを保存するかどうかを示します。 デフォルト値：No
MilterTraceContent *{Boolean}*	*Milter*プロトコルインターフェースを介してスキャン用に受信したメールメッセージの本文をデバッグログ（LogLevel = Debug）に出力します。 デフォルト値：No
MilterHook *{path to file / Lua function}*	*Milter*インターフェース経由で受信したメールメッセージを処理するLuaスクリプト、またはスクリプトを含むファイルへのパス（Luaでのメール処理セクションを参照）。 ファイルへのパスが正しくない場合、コンポーネントの起動時にエラーが返されます。 デフォルト値： `local dw = require "drweb"` `local dwcfg = require "drweb.config"` `function milter_hook(ctx)` `  -- Reject the message if it is likely spam` `  if ctx.message.spam.score >= 100 then` `    dw.notice("Spam score: " ..` `ctx.message.spam.score)` `    return {action = "reject"}` `  else`

パラメータ	説明
	`-- Assign X-Drweb-Spam headers in accordance with spam report` `ctx.modifier.add_header_field("X-DrWeb-SpamScore", ctx.message.spam.score)` `ctx.modifier.add_header_field("X-DrWeb-SpamState", ctx.message.spam.type)` `ctx.modifier.add_header_field("X-DrWeb-SpamDetail", ctx.message.spam.reason)` `ctx.modifier.add_header_field("X-DrWeb-SpamVersion", ctx.message.spam.version)` `end` `-- Check if the message contains viruses, repack if so` `for threat, path in ctx.message.threats{category = {"known_virus", "virus_modification", "unknown_virus", "adware", "dialer"}} do` `ctx.modifier.repack()` `dw.notice(threat.name .. " found in " .. (ctx.message.part_at(path).name or path))` `end` `-- Repack if unwanted URL has been found` `for url in ctx.message.urls{category = {"infection_source", "not_recommended", "owners_notice"}} do` `ctx.modifier.repack()` `dw.notice("URL found: " .. url .. "(" .. url.categories[1] .. ")")` `end` `-- Assign X-AntiVirus header` `ctx.modifier.add_header_field("X-AntiVirus", "Checked by Dr.Web [MailD version: " .. dwcfg.maild.version .. "]")` `-- Accept the message with all scheduled transformations applied` `return {action = 'accept'}` `end`
`SpamdSocket` *{path to file | IP address:port}*	メールメッセージの*Spamd*フィルターとしてMTAに接続するためのソケット（Dr.Web MailDを対応するフィルターとして使用しているときに、MTAはこのソケットに接続します）。UNIXソケットまたはネットワークソケットを使用できます。 *Spamd*経由で送信されるメッセージの処理ルールは、`SpamdReportHook`パラメータで指定します（以下を参照）。 デフォルト値：*（未設定）*
`SpamdDebugIpc` *{Boolean}*	*Spamd*プロトコルメッセージをデバッグログ（`LogLevel = Debug`）に出力します。

パラメータ	説明
	デフォルト値：No
SpamdReportHook *{path to file / Lua function}*	*Spamd* インターフェース経由で受信したメールメッセージを処理するLuaスクリプト、またはスクリプトを含むファイルへのパス（Luaでのメール処理セクションを参照）。 使用できないファイルを指定すると、コンポーネントを読み込む際にエラーが表示されます。 デフォルト値： `local dw = require "drweb"` `function spamd_report_hook(ctx)` `  local score = 0` `  local report = ""` `  -- Add 1000 to the score for each threat found in the message` `  for threat, path in ctx.message.threats{category = {"known_virus", "virus_modification", "unknown_virus", "adware", "dialer"}} do` `  score = score + 1000` `  report = report .. "Threat found: " .. threat.name .. "\n"` `  dw.notice(threat.name .. " found in " .. (ctx.message.part_at(path).name or path))` `  end` `  -- Add 100 to the score for each unwanted URL found in the message` `  for url in ctx.message.urls{category = {"infection_source", "not_recommended", "owners_notice"}} do` `  score = score + 100` `  report = report .. "Url found: " .. url .. "\n"` `  dw.notice("URL found: " .. url .. "(" .. url.categories[1] .. ")")` `  end` `  -- Add the spam score` `  score = score + ctx.message.spam.score` `  report = report .. "Spam score: " .. ctx.message.spam.score .. "\n"` `  if ctx.message.spam.score >= 100 then` `  dw.notice("Spam score: " .. ctx.message.spam.score)` `  end` `  -- Return the check result` `  return {` `  score = score,` `  threshold = 100,` `  report = report`

パラメータ	説明
	} end
SpoolDir *{path to directory}*	スキャンされたメールメッセージの一時ストレージディレクトリ。 デフォルト値：/tmp/com.drweb.maild
RspamdHttpSocket *{path to file / IP address:port}*	メールの*Rspamd*フィルターとしてMTAに接続するためのソケット（*Rspamd*プロトコルのHTTPオプションでDr.Web MailDを対応するフィルターとして使用しているときに、MTAはこのソケットを使用します）。UNIXソケットまたはネットワークソケットを使用できます。 *Rspamd*経由で送信されるメッセージの処理ルールは、RspamdHookパラメータで指定します（以下を参照）。 デフォルト値：*（未設定）*
RspamdSocket *{path to file / IP address:port}*	メールの*Rspamd*フィルターとしてMTAに接続するためのソケット（*Rspamd*プロトコルの*レガシー*オプションでDr.Web MailDを対応するフィルターとして使用しているときに、MTAはこのソケットを使用します）。UNIXソケットまたはネットワークソケットを使用できます。 デフォルト値：*（未設定）*
RspamdDebugIpc *{Boolean}*	*Rspamd*プロトコルメッセージをデバッグログ（LogLevel = Debug）に出力します。 デフォルト値：No
RspamdHook *{path to file / Lua function}*	*Rspamd*インターフェース経由で受信したメールメッセージを処理するLuaスクリプト、またはスクリプトを含むファイルへのパス（Luaでのメール処理セクションを参照）。 使用できないファイルを指定すると、コンポーネントを読み込む際にエラーが表示されます。 デフォルト値： `local dw = require "drweb"` `function rspamd_hook(ctx)` `  local score = 0` `  local symbols = {}` `  -- Add 1000 to the score for each threat found in the message` `  for threat, path in ctx.message.threats{category = {"known_virus", "virus_modification", "unknown_virus", "adware", "dialer"}} do` `  score = score + 1000` `  table.insert(symbols, {name = threat.name, score = 1000})` `  dw.notice(threat.name .. " found in " .. (ctx.message.part_at(path).name or path))` `  end`

パラメータ	説明
	`-- Add 100 to the score for each unwanted URL found in the message` `for url in ctx.message.urls{category = {"infection_source", "not_recommended", "owners_notice"}} do` `score = score + 100` `table.insert(symbols, {name = "URL " .. url, score = 100})` `dw.notice("URL found: " .. url .. "(" .. url.categories[1] .. ")")` `end` `-- Add the spam score` `score = score + ctx.message.spam.score` `table.insert(symbols, {name = "Spam score", score = ctx.message.spam.score})` `if ctx.message.spam.score >= 100 then` `dw.notice("Spam score: " .. ctx.message.spam.score)` `end` `-- Return the check result` `return {` `score = score,` `threshold = 100,` `symbols = symbols` `}` `end`
`SpfCheckTimeout` *{time interval}*	SPFチェックの最大合計時間。 デフォルト値：`20s`
`SpfVoidLimit` *{integer}*	SPFチェック中に許可される空の回答の最大数。 デフォルト値：`2`
`SmtpSocket` *{path to file / IP address:port}*	SMTPモードでメッセージのメールフィルターとしてMTAに接続するためのソケット（Dr.Web MailDを外部フィルターとして使用しているときに、MTAはこのソケットに接続します）。UNIXソケットまたはネットワークソケットを使用できます。 デフォルト値：*（未設定）*
`SmtpSenderRelay` *{path to file / IP address:port}*	SMTPモードでスキャンされるメッセージのメールフィルターとしてMTAに接続するためのソケット（Dr.Web MailDを外部フィルターとして使用しているときに、MTAはこのソケットに接続します）。UNIXソケットまたはネットワークソケットを使用できます。 デフォルト値：*（未設定）*
`BccSocket` *{path to file / IP address:port}*	BCCモードでメッセージのメールフィルターとしてMTAに接続するためのソケット（Dr.Web MailDを外部フィルターとして使用しているときに、MTAはこのソケットに接続します）。UNIXソケットまたはネットワークソケットを使用できます。 デフォルト値：*（未設定）*

パラメータ	説明
BccReporterAddress *{string}*	BCCモードでメールの添付ファイルをスキャンした後にDr.Web MailDレポートを送信する送信元メールアドレス。 デフォルト値：*（未設定）*
BccReporterPassword *{None | Plain(<password>)}*	BCCモードでメールの添付ファイルをスキャンした後にDr.Web MailDレポートを送信する送信元メールアドレスのパスワード。 使用可能な値： • None - メールはパスワードで保護されません。 • Plain(*<password>*) - 指定されたパスワードでメールが保護されます。 デフォルト値：None
BccReportRecipientAddress *{string}*	BCCモードでメールの添付ファイルをスキャンした後にDr.Web MailDレポートを送信する送信先メールアドレス。 デフォルト値：*（未設定）*
BccSmtpServer *{string}*	SMTPおよびBCCモードでメールメッセージを送信するためのMTAアドレス。ドメイン、IPアドレス、またはUNIXソケットを使用できます。 デフォルト値：*（未設定）*
VxcubePlatforms *{platform, … | All}*	Dr.Web vxCubeを外部フィルターモード（SMTPまたはBCC）でメールメッセージスキャンツールとして使用する場合に、メール添付ファイルを実行するためのOSプラットフォームのリスト。 リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 使用可能な値： • *<platform>* - Dr.Web vxCubeのplatforms API呼び出しのos_codeフィールドの値（ビット数指定のOS名）（詳細は、**Dr.Web vxCube**のユーザーマニュアルにあるプラットフォームのセクションを参照）。 • All - 利用可能なすべてのプラットフォーム。 デフォルト値：All
VxcubeFileFormats *{format, … | All}*	Dr.Web vxCubeを外部フィルターモード（SMTPまたはBCC）でメールメッセージスキャンツールとして使用する場合に解析のために送信される、メール添付ファイルを実行するためのOSプラットフォームのリスト。 リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 使用可能な値： • *<format>* - Dr.Web vxCubeのformats API呼び出しのnameフィールドの値（ファイルフォーマット名）（詳細は、**Dr.Web vxCube**のユーザーマニュアルにあるフォーマットのセクションを参照）。

パラメータ	説明
	• `All` - 利用可能なすべてのフォーマット。 デフォルト値：`All`
`VxcubeSampleRunTime` *{time interval}*	Dr.Web vxCubeを外部フィルターモード（SMTPまたはBCC）でメールメッセージスキャンツールとして使用する場合に解析のために送信される、メール添付ファイルを実行する時間。 デフォルト値：*（未設定）*
`SmtpHook` *{path to file | Lua function}*	SMTPモードで受信したメールメッセージを処理するLuaスクリプト、またはスクリプトを含むファイルへのパス（Luaでのメール処理セクションを参照）。 設定ファイルの`[Root]`セクションで`UseVxcube=Yes`パラメータの値が指定されている場合、Dr.Web vxCubeでメール添付ファイルをスキャンするアクションがデフォルトでLuaスクリプトに追加されます。 デフォルト値： `local dw = require "drweb"` `function smtp_hook(ctx)` `  -- Reject the message if it is likely spam` `  if ctx.message.spam.score >= 100 then` `  dw.notice("Spam score: " .. ctx.message.spam.score)` `  return {action = "discard"}` `  else` `  -- Добавить заголовки X-Drweb-Spam с отчетом о спаме` `  ctx.modifier.add_header_field("X-DrWeb-SpamScore", ctx.message.spam.score)` `  ctx.modifier.add_header_field("X-DrWeb-SpamState", ctx.message.spam.type)` `  ctx.modifier.add_header_field("X-DrWeb-SpamDetail", ctx.message.spam.reason)` `  ctx.modifier.add_header_field("X-DrWeb-SpamVersion", ctx.message.spam.version)` `  end` `  -- Check if the message contains viruses, repack if so` `  threat_categories = {"known_virus", "virus_modification", "unknown_virus", "adware", "dialer"}` `  if ctx.message.has_threat({category = threat_categories}) then` `  for threat, path in ctx.message.threats({category = threat_categories}) do` `  dw.notice(threat.name .. " found in " .. (ctx.message.part_at(path).name or path))` `  end` `  ctx.modifier.repack()`

パラメータ	説明
	`return {action = "accept"}` `end` `-- Repack if unwanted URL has been found` `url_categories = {"infection_source", "not_recommended", "owners_notice"}` `if ctx.message.has_url({category = url_categories}) then` `for url in ctx.message.urls({category = url_categories}) do` `dw.notice("URL found: " .. url .. " (" .. url.categories[1] .. ")")` `end` `ctx.modifier.repack()` `return {action = "accept"}` `end` `-- Accept the message with all scheduled transformations applied` `return {action = 'accept'}` `end`
`SmtpRetryInterval` *{time interval}*	SMTPモードで動作する場合に、エラー発生時のメッセージのスキャンや送信の再試行のタイムアウト。 デフォルト値：`5m`
`SmtpRequireTls` *{Always | IfSupported | Never}*	SMTPモードでSTARTTLS拡張機能を使用して動作するときのSMTPプロトコルポリシーを定義します。 使用可能な値： • `Always` - 常に保護された接続を使用します。サーバー自体が保護をサポートしていない場合は、接続を中断します。 • `IfSupported` - サーバーがサポートしている場合は、保護された接続を優先します。それ以外の場合は、保護されていないチャネルでメッセージを送信します。 • `Never` - 保護されていない接続を使用しません。 デフォルト値：`Always`
`SmtpDebugIpc` *{Boolean}*	デバッグログ（`LogLevel = Debug`）にSMTPモードのSMTPコマンドを保存するかどうかを示します。 デフォルト値：`No`
`SmtpTraceContent` *{Boolean}*	デバッグログ（`LogLevel = DEBUG`）にSMTPモードのメールコンテンツを保存するかどうかを示します。 デフォルト値：`No`
`CaPath` *{path to file or directory}*	信頼できるルート証明書のリストを含むディレクトリまたはファイルへのパス。 デフォルト値：*信頼できる証明書のリストへのパス*。パスはGNU/Linuxディストリビューションに依存します。

パラメータ	説明
	• Astra Linux、Debian、Linux Mint、SUSE Linux、Ubuntuの場合、通常はパス/etc/ssl/certs/です。 • CentOSとFedoraの場合はパス/etc/pki/tls/certs/ca-bundle.crtです。 • 他のディストリビューションでは、コマンドopenssl version -dの実行結果によってパスを定義できます。 • コマンドが使用できない場合、またはOSディストリビューションを特定できない場合は、値/etc/ssl/certs/が使用されます。
Hostname *{string}*	送信者のホスト名（FQDN）。SMTPクライアントから受信したウェルカム文字列HELO/EHLOの他、タイトルAuthentication-Resultsのsrvnameのデフォルト値にも表示されます。 デフォルト値：*現在のホスト名*

メールシステムとの統合

Dr.Web MailDとメールシステムとの統合については、以下のセクションで説明します。

- フィルターとしてのMTAとの統合 - メールスキャン用の外部フィルターとしてDr.Web MailDをメールサーバー（Exim、Sendmail、Postfix）に接続します。
- Dr.Web vxCubeとの統合 - メール添付ファイルを解析するためのDr.Web vxCube Webサービスを利用した外部フィルターとして、Dr.Web MailDをメールサーバーに接続します。
- SMTPプロキシモードでDr.Web for UNIX Mail Serversを使用する - メールスキャンの外部フィルターとして、メールメッセージの転送を実行するメールサーバーにDr.Web MailDを接続します。
- 透過プロキシモードでDr.Web for UNIX Mail Serversを使用する - Dr.Web MailDをメールプロトコル（SMTP、POP3、IMAP）に直接統合します。MTA/MDAおよびMUAに対して透過します。

それ以外にも、MTAをDr.Web ClamDコンポーネントに直接接続して、スパムの兆候やその他の脅威に対してメールスキャンを実行できます。

Luaでのメール処理

このセクションの内容

- 概要
- Milter用のスクリプト
- Spamd用のスクリプト
- Rspamd用のスクリプト
- SMTP用のスクリプト
- メッセージ構造を指定するテーブル
- 利用可能な補助モジュール

概要

Dr.Web MailDコンポーネントはLuaプログラムインタプリタを介したインタラクションをサポートします（バージョン5.3.4を使用。Dr.Web for UNIX Mail Serversに同梱）。Luaで記述されたスクリプトをコンポーネントで使用すると、メールメッセージの解析と処理を実行できます。

Milter、Spamd、Rspamd、またはSMTPモードで受信したメールメッセージは、Dr.Web MailD設定で`MilterHook`、`SpamdHook`、`RspamdHook`、または`Smtphook`パラメータの値としてそれぞれ指定したLuaスクリプトを使用して解析されます。これらのパラメータの値はスクリプトの全文またはスクリプトへのパスとして指定できます。

その他のメール処理用Luaスクリプトの例は、次の場所にあります。
https://github.com/DoctorWebLtd/drweb-lua-examples/tree/master/maild

Milterインターフェースのメッセージ処理のためのスクリプト

スクリプトの必要条件

スクリプトには、メッセージスキャンモジュールのエントリポイントとなるグローバル関数が含まれている必要があります（Dr.Web MailDは、すべての受信メッセージを処理する際にこの関数を呼び出します）。処理関数は、次の呼び出し規則を満たす必要があります。

1. *関数名*は`milter_hook`です。
2. *引数*はMilterContextテーブルのみです（関数から処理されたメールメッセージに関する情報へのアクセス権限を付与します）。
3. *返り値*はMilterResult完了テーブルのみです。返り値は、スキャンされたメッセージについての判定（承認："accept"、拒否："reject"、変更："change"、破棄："discard"）の他、メッセージが承認された場合にメッセージに適用される（可能性がある）アクションを定義します。

以下に示すのは、*Milter*インターフェースを介してスキャンするために、受信したすべてのメッセージの`accept`判定を常にDr.Web MailDに返す関数を定義した例です（以降、`ctx`引数は*MilterContext*テーブルのインスタンスです）。

```
function milter_hook(ctx)
  return {action = "accept"}
end
```

例

1. 脅威を検出し、スパムの兆候をチェックする

次のスクリプトは以下のように動作します。

- メールメッセージで検出された脅威の名前をヘッダー`X-Found`の値として追加する。
- スパムスコアが100ポイントを超える場合は、メールの件名（ヘッダー`Subject`の値）に`[SPAM]`プレフィックスを追加する。
- 処理後のメッセージを受信者に送信する。

```
function milter_hook (ctx)

 -- Add the detected threats' names to the header
 for threat in ctx.message.threats() do
   ctx.modifier.add_header_field("X-Found", threat.name)
 end

 -- Change the value of the Subject header, if a message has more than 100
points of spam scoring
 if ctx.message.spam.score > 100 then
   local old_value = ctx.message.header.value("Subject") or ""
   local new_value = "[SPAM] " .. old_value
   ctx.modifier.change_header_field("Subject", new_value)
 end

 -- Send a message to a recipient, applying the pending changes
 return {
   action = "accept",
   modifications = ctx.modifier.modifications()
 }

end
```

2. 検出されたすべての脅威を保護されたアーカイブに配置し、メッセージをアーカイブする（スパムスコアを超えた場合）

次のスクリプトは以下のように動作します。

- 検出された脅威を保護されたアーカイブに移動する。
- スパムスコアが100ポイントを超える場合は、保護されたアーカイブにメールメッセージを移動する。

```
function milter_hook(ctx)

 ctx.modifier.repack_password = "xxx"
 ctx.modifier.repack_message = ""

 -- Move all message parts where threats were found
 -- to a password protected archive
 for threat, path in ctx.message.threats() do
   ctx.modifier.repack(path)
   local msg = " Threat found: " .. threat.name
   ctx.modifier.repack_message = ctx.modifier.repack_message .. msg
 end

 -- Repack the whole email message if it has
 -- more than 100 points of spam scoring
 if ctx.message.spam.score > 100 then
   ctx.modifier.repack()
   local msg = " Spam score: " .. ctx.message.spam.score
   ctx.modifier.repack_message = ctx.modifier.repack_message .. msg
 end

 -- Send a message to a recipient, applying the pending changes
 -- Note that if the modification table is not specified,
 -- it will be automatically returned
 return {action = "accept"}

end
```

チェックされているメッセージは修正されます。望ましくない部分はすべて削除され、アーカイブされて、添付ファイルとして受信者に送信されます。
メッセージの望ましくない要素が含まれるアーカイブは、`ctx.modifier.repack_password`変数の値として指定されたパスワードで保護されます。この場合、設定ファイルの`RepackPassword`パラメータで指定されているパスワードは無効になります。

スクリプトで使用されるテーブル

テーブル MilterContext

このテーブルは`milter_hook`関数の入力引数として使用されます。このテーブルには、処理されているメールメッセージに関する情報（フィールド、構造、ヘッダー、本文、送信者と受信者に関する情報、SMTPセッションに関する情報）が含まれます。

フィールド	説明	データタイプ
`session_id`	このクライアントからのメッセージが処理されている間のクライアントセッションの識別子。	文字列
`sender`	メールメッセージの送信者に関する情報。	テーブル MilterSender
`helo`	メールメッセージを送信したSMTPクライアントから受信したウェルカム文字列 HELO/EHLO、または文字列が見つからない／不明の場合（MTAによって提供されない場合）は`nil`。	文字列

フィールド	説明	データタイプ
from	山括弧なしの送信者のメールアドレス（例：user@domain.com）。	文字列
to	山括弧なしの受信者のメールアドレス	テーブル RcptTo
message	メールメッセージ（本文とすべてのヘッダーを含む）。	テーブル MimeMessage
modifier	MilterResultテーブルのスキャン結果に従って"accept"アクションが生成された場合にメッセージに適用される、すべての変更が含まれるMilterModifierテーブル。	テーブル MilterModifier
gen	X-Antivirusなどの特殊なヘッダーを生成するために使用されるMilterGeneratorテーブル	テーブル MilterGenerator
spf	送信者のSPFチェックの実行に使用されるSPFテーブル	テーブルSPF
無効になったメタメソッド：*なし*		

テーブルMilterSender

このテーブルはMilterContextテーブルのsenderフィールドとして使用されます。このテーブルには、メール送信者に関する情報が含まれます。

フィールド	説明	データタイプ
hostname	送信者のホスト名（FQDN）	文字列
family	文字列で示した接続タイプ • "U" - 不明なタイプ • "L" - UNIXソケット経由の接続 • "4" - IPv4経由の接続 • "6" - IPv6経由の接続	文字列
port	ポート番号	整数
ip	送信者のホストのIPアドレス。IPアドレスがないか不明の場合（MTAから提供されない場合）はnil	テーブルIpAddress
無効になったメタメソッド：*なし*		

テーブルMilterResult

このテーブルは、milter_hook関数が返す結果を表しています。

フィールド	説明	データタイプ
action	メッセージに適用するアクションの説明を含む文字列： • "accept" - 承認（MTAから受信者へのメッセージ送信を許可します）。 • "discard" - 送信者に通知せずにメッセージを破棄します。 • "reject" - メッセージを拒否し、SMTP 5**コードを送信者に返します。 • "tempfail" - メッセージを拒否し、SMTP 4**コードを送信者に返します。 • "replycode" - codeフィールドで指定されたSMTPレスポンスを送信者に送信します。 必須フィールド。	文字列
code	送信者に送信される3桁のSMTPレスポンスコード（例："541"）。 オプションのフィールドです。action = "replycode"の場合にのみ使用します。	文字列
text	送信者に送信される応答テキストを含む文字列（例："User not local or invalid address – Relay denied"）。 オプションのフィールドです。action = "replycode"の場合にのみ使用します。	文字列
message	送信者に送信された応答テキストを含む文字列（例："Message rejected as spam"）。 オプションのフィールドです。action = "reject"の場合にのみ使用します。	文字列
modifications	受信者に送信する前に、メールメッセージに適用する変更を指定するテーブル。 オプションのフィールドです。action = "accept"の場合にのみ使用します。	テーブル MilterModifications
added_recipients	追加のメッセージ受信者のメールアドレスリスト。 オプションのフィールドです。action = "accept"の場合にのみ使用します。	文字列の配列
deleted_recipients	メッセージ受信者のリストから除外するメールアドレスのリスト。 オプションのフィールドです。action = "accept"の場合にのみ使用します。	文字列の配列
incident	Mailタイプの登録済みイベントでのインシデントの説明。 • このフィールドが文字列の場合、この文字列の内容がMailイベントの*incident_text*フィールドに送信され、イベントが登録されます。	文字列またはブール値

フィールド	説明	データタイプ
	• このフィールドがブール値で false に等しい場合、Mail イベントの *incident_text* フィールドは存在せず、イベントは登録されません。 • このフィールドがブール値で true に等しい場合、Mail イベントの *incident_text* フィールドは自動的に入力され、*incident_text* の値が空でない場合はイベントが登録されます。	
無効になったメタメソッド: *なし*		

テーブル MilterModifications

このテーブルは、受信者への配信が決定された場合（accept アクションが選択された場合）に、メールメッセージに対して行われるすべての変更の指定に使用されます。このテーブルのフィールドはすべてオプションです。フィールド値がない場合、このフィールドで指定されたアクションはメッセージに適用されません。

フィールド	説明	データタイプ
new_body	処理されているメッセージの本文を置き換えるために使用される新しいメッセージ本文（ヘッダーなし）	文字列
added_fields	処理されているメッセージに追加されるヘッダー	MilterAddedField テーブルの配列
changed_fields	処理されているメッセージから変更される、または削除されるヘッダー	MilterChangedField テーブルの配列
無効になったメタメソッド: *なし*		

テーブル MilterAddedField

このテーブルには、処理されているメッセージに追加されるヘッダーの指定が含まれます。

フィールド	説明	データタイプ
name	ヘッダー名	文字列
value	ヘッダー値	文字列
無効になったメタメソッド: *なし*		

テーブル MilterChangedField

このテーブルには、処理されているメッセージで変更される（またはメッセージから削除される）ヘッダーの指定が含まれます。

フィールド	説明	データタイプ
name	ヘッダー名	文字列
index	変更するメッセージ内で name の名前を持つヘッダーの序数（1からカウント）	番号

フィールド	説明	データタイプ
value	ヘッダーの新しい値（ヘッダーを削除する場合、値は空の文字列""）	文字列
無効になったメタメソッド: *なし*		

処理後にメッセージに加えられる変更の説明については、MilterModificationsテーブルに直接入力しないことをお勧めします。代わりに、コンテキストから受け取ることができるMilterModifierテーブルのメソッドを使用してください。

テーブルMilterGenerator

このテーブルには、X-AntivirusおよびX-Authentication-Resultsの標準ヘッダーを生成するための二次的な方法が含まれます。

フィールド	説明	データタイプ
x_antivirus_header_field	この関数は、メッセージのスキャンに関与したアンチウイルスコンポーネントに関する情報を含むX-Antivirusヘッダーを生成するために使用され、HeaderFieldテーブルまたはnil（メッセージがまだスキャンされていない場合）を返します。 *HeaderField*のnameフィールドは固定値（X-Antivirus）です。	機能
authentication_results_header_field	この関数は、DKIMおよびSPFチェックの結果に関する情報を含むAuthentication-Resultsヘッダーを生成するために使用されます。オプションの引数として、生成されるヘッダー内のサーバーのIDであるauthserv_id文字列を取ります。authserv_idの値はデフォルトで、Dr.Web MailDが実行されているホストの名前と一致します。 この関数はHeaderFieldテーブルを返します。 *HeaderField*のnameフィールドは固定値（Authentication-Results）です。	機能
無効になったメタメソッド: *なし*		

テーブルMilterModifier

このテーブルは、メッセージの処理後（受信者に送信される場合）に行う、メッセージに対する変更内容の指定に使用されます。

フィールド	説明	データタイプ
add_header_field	メールメッセージに新しいヘッダーを追加するアクションを指定する関数。 次の2つの必須の引数を受け取ります。 • nameはヘッダー名（文字列）です。 • valueはヘッダー値（文字列）です。 値はRFC 2047に従ってエンコードされます。	機能

フィールド	説明	データタイプ
change_header_field	指定されたヘッダーを変更（または削除）するアクションを指定する関数。 次の2つの必須の引数を受け取ります。 • nameはヘッダー名（文字列）です。 • valueはヘッダー値（文字列）です。 メッセージに指定された名前を持つ複数のヘッダーがある場合、この関数はその名前を持つ最初のヘッダーの値を変更します。同じヘッダーの値が複数回変更された場合は、最後の値だけが保持されます。valueが空の文字列""の場合、ヘッダーのnameはメッセージから削除されます。 値はRFC 2047に従ってエンコードされます。	機能
modifications	メールメッセージへの実行が指定されている変更の全リストを含むMilterModificationsテーブルを返す関数。引数はありません。	機能
repack	指定したメッセージ部分（部分が指定されていないか、指定された部分がない場合はメッセージ全体）の再圧縮を指定する関数。再圧縮中に、指定した部分がパスワード付きでアーカイブに追加されます。 オプションの引数pathまたはiteratorを受け入れます。 • pathは、アーカイブされるスキャンされたメールの添付ファイルへのパスです。パスが指定されていない場合や、指定されたパスが無効な場合は、メッセージ全体がアーカイブされます。 • iteratorはメッセージ部分の反復子であり、関数によって、MimePartテーブルのthreats、urls、attachments、files、parts、leaf_parts、text_partsが返されます。この場合、反復子から返されたメッセージのすべての部分が再圧縮されるように指定されます。 関数の引数が指定されていない場合、または指定されたパスが無効な場合は、メッセージ全体がアーカイブされます。	機能

フィールド	説明	データタイプ
	例： `-- Schedule the entire message to be repacked` `-- to an archive with a password` `ctx.modifier.repack()` `-- Schedule some parts of a message at the specified path` `-- (or the entire message if the part does not exist)` `-- to be repacked to an archive with a password` `ctx.modifier.repack('/1/2/3')` `-- Schedule all message parts that contain executables` `-- to be repacked to an archive with a password` `ctx.modifier.repack(ctx.message.files{name='*.exe'})` `-- Schedule all ZIP attachements to be repacked` `-- to an archive with a password` `ctx.modifier.repack(ctx.message.attachments{name='*.zip'})`	
`repack_archive_name`	メッセージの悪意のあるアイテムまたは不要なアイテムを圧縮するためのアーカイブの名前。デフォルトでは、`"quarantine.zip"`。	文字列
`repack_password`	アーカイブ保護のためのパスワード。指定されていない場合は、構成ファイルに指定されたパスワードが使用されます（`RepackPassword`パラメータ）。	文字列
`repack_message`	メッセージ（またはその部分）の再圧縮の理由を表す任意のメッセージ。最終メッセージに追加されます（省略することもできます）。	文字列
`templates_dir`	再圧縮のためのテンプレートが保存されているディレクトリへのパス。パスは、設定ファイル（`TemplatesDir`パラメータ）に指定したパスを基準にした相対パスになります。デフォルト値は`"milter"`です（つまり、`milter`サブディレクトリのテンプレートが使用されます）。	文字列
`cure`	添付ファイルの修復を指定する関数。 オプションの引数`path`または`iterator`を受け入れます。 • `path`は、スキャンされたメール添付ファイルへのパスです。添付ファイルが無害であるか、または修	機能

フィールド	説明	データタイプ
	復されている場合、cure(path)関数はtrueを返します。添付ファイルが修復できない場合、この関数はfalseを返します。 • iteratorはメッセージ部分の反復子であり、関数によってMimePartテーブルのthreats、urls、attachments、files、parts、leaf_parts、text_partsが返されます。この場合、反復子から返されたメッセージのすべての部分が修復されるように指定されます。すべての添付ファイルが無害であるか、または修復されている場合、cure(iterator)関数はtrueを返します。少なくとも1つの添付ファイルが修復できない場合、この関数はfalseを返します。 関数の引数が指定されていない場合、cure(ctx.message.leaf_parts())を呼ぶのと同じです。すべての添付ファイルが無害であるか、または修復されている場合、trueを返します。少なくとも1つの添付ファイルが修復できない場合、この関数はfalseを返します。	
cure_or_repack	添付ファイルの修復を指定する関数。修復できない場合、添付ファイルは再圧縮されます。再圧縮中に、指定した部分がパスワード付きでアーカイブに追加されます。 オプションの引数pathまたはiteratorを受け入れます。 • pathは、スキャンされたメール添付ファイルへのパスです。添付ファイルが無害であるか、または修復されている場合、cure_or_repack(path)関数はtrueを返します。添付ファイルが修復できない場合、この関数はfalseを返し、添付ファイルを再圧縮するよう指定します。 • iteratorはメッセージ部分の反復子であり、関数によってMimePartテーブルのthreats、urls、attachments、files、parts、leaf_parts、text_partsが返されます。この場合、反復子から返されたメッセージのすべての部分が修復されるように指定されます。すべての添付ファイルが無害であるか、または修復されている場合、cure_or_repack(iterator)関数はtrueを返します。少なくとも1つの添付ファイルが修復できない場合、この関数はfalseを返し、修復できないすべての添付ファイルを再圧縮するよう指定します。 関数の引数が指定されていない場合、cure_or_repack(ctx.message.leaf_parts())を呼ぶのと同じです。すべての添付ファイルが無害であるか、または修復されている場合、trueを返します。少なくとも1つの添付ファイルが修復できない場合、この関数はfalseを返し、修復できない	機能

フィールド	説明	データタイプ
	すべての添付ファイルを再圧縮するよう指定します。	
無効になったメタメソッド: *なし*		

このテーブルにアクセスするには、MilterContextテーブルの`modifier`フィールドを使用する必要があります。
例:

```
function milter_hook(ctx)

 -- Schedule adding a new header
 -- at the end of the list of headers
 ctx.modifier.add_header_field("X-Name", "Value")

 -- Schedule changing the "Subject" field value to "New value"
 ctx.modifier.change_header_field("Subject", "New value")

 -- Schedule repacking messages to an archive with a password
 ctx.modifier.repack()

 -- Return verdict via the milter protocol (table MilterResult)
 -- and apply all pending changes of the message
 return {action = "accept"}
end
```

以下に示すのは、MilterModifierテーブルは使用せずに、MilterResultテーブル(およびその`modifications`フィールド、つまりMilterModificationsテーブル)に直接入力する例です。

```
-- Enable message sending to recipients by adding
-- the "X-Checked: True" header

function milter_hook(ctx)
 return {
  action = "accept",
  modifications = {
    added_fields = {
      {
        name = "X-Checked",
        value = "True"
      }
    }
  }
 }
end
```

次に示すのは、MilterModifierテーブルに加えられた変更に関係なくacceptを返す例です。

```
function milter_hook(ctx)

 …

 -- Schedule adding the message header
 ctx.modifier.add_header_field('X-Header', 'some value')

 …

 -- Force return of an empty MilterModifications table
 return {action = "accept", modifications = {}}
end
```

Spamdインターフェースのメッセージ処理のためのスクリプト

スクリプトの必要条件

スクリプトには、メッセージスキャンモジュールのエントリポイントとなるグローバル関数が含まれている必要があります（Dr.Web MailDは、すべての受信メッセージを処理する際にこの関数を呼び出します）。処理関数は、次の呼び出し規則を満たす必要があります。

1. *関数名*はspamd_report_hookです。
2. *引数*はSpamdContextテーブルのみです（関数から処理されたメールメッセージに関する情報へのアクセス権限を付与します）。
3. *返り値*は入力済みのSpamdReportResultテーブルのみです。返り値は*Spamd*を介したレスポンスを定義します。

以下に示すのは、メッセージをスパムとしてマークする必要があるという判定を常にDr.Web MailDに返す関数を定義した例です（スパムスコア：200、スパム判定最低スコア：100、メッセージ：The message was recognized as spam。以降、ctx引数は*SpamdContext*テーブルのインスタンスです）。

```
-- An example of a trivial realization

function spamd_report_hook(ctx)
 return {
  score = 200,
  threshold = 100,
  report = "The message was recognized as spam"
 }
end
```

Luaスクリプトで使用されるテーブル

テーブルSpamdContext

このテーブルはspamd_report_hook関数の入力引数として使用されます。このテーブルには、処理されているメールメッセージに関する情報（構造、ヘッダー、本文、送信者と受信者に関する情報、SMTPセッションに関する情報）が含まれます。

フィールド	説明	データタイプ
session_id	このクライアントからのメッセージが処理されている間のクライアントセッションの識別子。	文字列
message	メールメッセージ	テーブル MimeMessage
無効になったメタメソッド: *なし*		

テーブルSpamdReportResult

このテーブルは、spamd_report_hook関数が返す結果を表します。Dr.Web MailDにスパムのスキャン結果を渡し、結果はMTAに返されます。

フィールド	説明	データタイプ
score	メッセージに割り当てられたスパム評価（Exim MTAの場合、$spam_scoreおよび$spam_score_int変数が入力されます）	番号
threshold	メッセージがスパムと判定されるしきい値	番号
report	メッセージスキャンのテキスト結果（Exim MTAの場合、$spam_report変数が入力されます）	文字列
incident	Mailタイプの登録済みイベントでのインシデントの説明。 • このフィールドが文字列の場合、この文字列の内容がMailイベントの*incident_text*フィールドに送信され、イベントが登録されます。 • このフィールドがブール値でfalseに等しい場合、Mailイベントの*incident_text*フィールドは存在せず、イベントは登録されません。 • このフィールドがブール値でtrueに等しい場合、Mailイベントの*incident_text*フィールドは自動的に入力され、*incident_text*の値が空でない場合はイベントが登録されます。	文字列またはブール値
無効になったメタメソッド: *なし*		

Rspamdインターフェースのメッセージ処理のためのスクリプト

スクリプトの必要条件

このスクリプトには、メッセージスキャンモジュールのエントリポイントとして機能するグローバル関数が含まれている必要があります（Dr.Web MailDは、新たに受信したメッセージを処理する際にこの関数を呼び出します）。処理関数は、次の呼び出し規則を満たす必要があります。

1. *関数名*はrspamd_hookです。
2. *引数*はRspamdContextテーブル*のみ*です（関数から処理されたメールメッセージに関する情報へのアクセス権限を付与します。下記のテーブルの説明を参照）。
3. *返り値*は入力済みのRspamdResultテーブルのみです（下記のテーブルの説明を参照）。返り値は*RspamD*を介したレスポンスを定義します。

スクリプトの定義例（ctx引数は*RspamdContext*テーブルのインスタンスになります）：

```
-- An example of a trivial realization

function rspamd_hook(ctx)
 return {
  score = 200,
  threshold = 100
 }
end
```

例

以下のスクリプトは、簡潔なコメントとともにスパムスコアを含むRspamdSymbol（メッセージで検出されたスパムの兆候と、それぞれの兆候に対応するポイント数）の他に、MTAの推奨されるアクションを返します。

```
function rspamd_hook(ctx)
 return {
  score = 1080,
  threshold = 100,
  action = "REJECT:Malicious message"
  symbols = {
   {
     name = "Threat found",
     score = 1000
   },
   {
     name = "Spam score by the third-party anti-spam library",
     score = 80
   }
  }
 }
end
```

Luaスクリプトで使用されるテーブル

テーブルRspamdContext

このテーブルはrspamd_hook関数の入力引数として使用され、処理されているメールメッセージに関する情報が含まれます。

フィールド	説明	データタイプ
session_id	このクライアントからのメッセージが処理されている間のクライアントセッションの識別子。	文字列
sender	メッセージ送信者に関する情報	テーブル RspamdSender
helo	SMTPクライアントから受信した文字列HELO/EHLO、または文字列が見つからない／不明の場合（MTAによって提供されない場合）は nil	文字列

フィールド	説明	データタイプ
from	送信者のメールアドレス（山括弧なし、例：user@domain.com）。アドレスが見つからない／不明な場合（MTAから提供されない場合）はnil	文字列
to	山括弧なしの受信者のメールアドレス	テーブルRcptTo
message	メールメッセージ	テーブルMimeMessage
spf	送信者のSPFチェックの実行に使用されるSPFテーブル	テーブルSPF
無効になったメタメソッド：*なし*		

テーブルRspamdSender

このテーブルには、メッセージ送信者に関する情報が含まれます。

フィールド	説明	データタイプ
hostname	送信者のホストの名前（FQDN）、または名前が見つからないか不明の場合（MTAによって提供されない場合）はnil	文字列
ip	送信者のホストのIPアドレス。アドレスが見つからないか不明の場合（MTAから提供されない場合）はnil	テーブルIpAddress
無効になったメタメソッド：*なし*		

テーブルRspamdResult

このテーブルは、rspamd_hook関数が返す結果を表しており、メッセージのスキャンレポートが含まれます。

フィールド	説明	データタイプ
score	スキャン後にメッセージに割り当てられたスパムスコア（Exim MTAの場合、$spam_scoreおよび$spam_score_int変数が入力されます）	番号
threshold	メッセージがスパムとして扱われる最小のスパムスコア	番号
action	オプションのフィールドで、メッセージスキャンの結果としてMTAに推奨されるアクションです（Exim MTAの場合、$spam_action変数が入力されます）。	文字列
symbols	オプションのフィールドです。scoreフィールドで指定されたスパムポイント数を割り当てる理由を特定するためのRspamdSymbolテーブルの配列	RspamdSymbolテーブルの配列
incident	Mailタイプの登録済みイベントでのインシデントの説明。 • このフィールドが文字列の場合、この文字列の内容がMailイベントの*incident_text*フィールドに送信され、イベントが登録されます。	文字列またはブール値

フィールド	説明	データタイプ
	• このフィールドがブール値で `false` に等しい場合、`Mail` イベントの *incident_text* フィールドは存在せず、イベントは登録されません。 • このフィールドがブール値で `true` に等しい場合、`Mail` イベントの *incident_text* フィールドは自動的に入力され、*incident_text* の値が空でない場合はイベントが登録されます。	
無効になったメタメソッド: *なし*		

テーブル RspamdSymbol

このテーブルでは、メッセージ内で検出されたスパムの兆候（脅威のあるファイル、望ましくないURLなど）のうち、スパムスコアに加点されるものを指定します。

フィールド	説明	データタイプ
`name`	検出されたスパムの兆候の名前	文字列
`score`	この兆候が検出されるとスパムスコアに加算されるポイント	番号
`description`	検出されたスパムの兆候の簡単な説明（オプションのフィールド）	文字列
無効になったメタメソッド: *なし*		

SMTPモードのメッセージ処理のためのスクリプト

スクリプトの必要条件

このスクリプトには、メッセージスキャンモジュールのエントリポイントとなるグローバル関数が含まれている必要があります（Dr.Web MailDは、新たに受信したメッセージを処理する際にこの関数を呼び出します）。処理関数は、次の呼び出し規則を満たす必要があります。

1. *関数名*は `smtp_hook` です。
2. *引数*はSmtpContextテーブルのみです（関数から処理されたメールメッセージに関する情報へのアクセス権限を付与します）。
3. *返り値*はSmtpResult完了テーブルのみです。返り値は、スキャンされたメッセージについての判定（承認: "accept"、拒否: "reject"、変更: "change"、破棄: "discard"）の他、メッセージが承認された場合にメッセージに適用される（可能性がある）アクションを定義します。

以下に示すのは、SMTPモードを介してスキャンするために、受信したすべてのメッセージの `Accept` 判定を常にDr.Web MailDに返す関数を定義した例です（以降、*ctx* 引数は *SmtpContext* テーブルのインスタンスです）。

```
function smtp_hook(ctx)
 return {action = "accept"}
end
```

SMTPモードでは、新しいヘッダーの追加やヘッダーの変更、メール受信者の追加または削除、メール本文の変更など、メールの変更を行えます。また、SMTPモードはDr.Web vxCube Webサービスとの統合をサポートしているので、メール添付ファイルの解析も行えます。Dr.Web vxCubeから受け取った判定結果は、メールに適用するアクションの決定に使用できます。

例

以下のスクリプトは、すべてのメールメッセージについて*Accept*の判定をDr.Web MailDに返し、すべてのメッセージに"X-Checked: True"ヘッダーフィールドを追加します。

```
function smtp_hook(ctx)
 return {
  action = "accept",
  modifications = {
   added_fields = {
     {
      name = "X-Checked",
      value = "True"
     }
   }
  }
 }
end
```

modificationsテーブルを形成するために、次の例のように、SmtpContextテーブルの補助的なMilterModifierオブジェクトを使用できます。

```
function smtp_hook(ctx)
 local modifier = ctx.modifier

 -- Schedule appending a new field to the end of the header
 modifier.add_header_field("X-Name", "Value")

 -- Schedule changing the "Subject" field value to "New value"
 modifier.change_header_field("Subject", "New value")

 -- Schedule repacking messages to an archive with a password
 modifier.repack()

 -- Apply all pending changes to the message and send it
 -- modifications do not have to be specified, the changes will be taken
directly from modifier
 return { action = "accept", modifications = modifier.modifications() }
end
```

スクリプトで使用されるテーブル

テーブルSmtpContext

このテーブルはsmtp_hook関数の入力引数として使用され、処理されているメールメッセージに関する情報が含まれます。

フィールド	説明	データタイプ
session_id	このクライアントからのメッセージが処理されている間のクライアントセッションの識別子。	文字列

フィールド	説明	データタイプ
sender	メールメッセージの送信者に関する情報。	テーブル MilterSender
helo	SMTPクライアントから受信した文字列HELO/EHLO、または文字列が見つからない／不明の場合（MTAによって提供されない場合）は nil	文字列
from	送信者のアドレス（山括弧なし。例："user@domain.com"）	文字列
to	山括弧なしの受信者のメールアドレス	テーブルRcptTo
message	メールメッセージ	テーブル MimeMessage
modifier	MilterModifierテーブルには、MilterResultテーブルのスキャン結果に従って"accept"アクションが生成された場合にメッセージに行われるすべての変更が含まれます。	テーブル MilterModifier
gen	X-Antivirusなどの特殊なヘッダーを生成するために使用されるMilterGeneratorテーブル	テーブル MilterGenerator
spf	送信者のSPFチェックの実行に使用されるSPFテーブル	テーブルSPF
無効になったメタメソッド：*なし*		

テーブルSmtpResult

このテーブルは、smtp_hook関数が返す結果を表しており、チェックされているメッセージに適用されるアクションの指定が含まれます。

フィールド	説明	データタイプ
action	メッセージに適用するアクションの説明を含む文字列： • "accept" - 承認（MTAから受信者へのメッセージ送信を許可します）。 • "discard" - 送信者に通知せずにメッセージを破棄します。 • "tempfail" - メッセージを拒否し、SMTP 4**コードを送信者に返します。 必須フィールド。	文字列
modifications	受信者に送信する前に、メールメッセージに適用する変更を指定するテーブル。 オプションのフィールドです。action = "accept"の場合にのみ使用します。	テーブル MilterModifications
added_recipients	追加のメッセージ受信者のメールアドレスリスト。 オプションのフィールドです。action = "accept"の場合にのみ使用します。	文字列の配列

フィールド	説明	データタイプ
deleted_recipients	メッセージ受信者のリストから除外するメールアドレスのリスト。 オプションのフィールドです。action = "accept"の場合にのみ使用します。	文字列の配列
message	拒否されているメッセージについて送信者に送信される応答テキストを含む文字列。同期的にメッセージがチェックされている場合は、541コードとともに送信者に返されます。 オプションのフィールドです。action = "reject"の場合にのみ使用します。	文字列
incident	Mailタイプの登録済みイベントでのインシデントの説明。 • このフィールドが文字列の場合、この文字列の内容がMailイベントの*incident_text*フィールドに送信され、イベントが登録されます。 • このフィールドがブール値でfalseに等しい場合、Mailイベントの*incident_text*フィールドは存在せず、イベントは登録されません。 • このフィールドがブール値でtrueに等しい場合、Mailイベントの*incident_text*フィールドは自動的に入力され、*incident_text*の値が空でない場合はイベントが登録されます。	文字列またはブール値
無効になったメタメソッド: *なし*		

メッセージ構造を指定するテーブル

テーブルRcptTo

このテーブルには、SMTPプロトコルのRCPT TOコマンドで発生したメッセージ受信者のメールアドレス(山括弧なし)の配列の他、次の追加情報が含まれます。

フィールド	説明	データタイプ
search	アドレス配列内の指定されたテンプレートの少なくとも1つに対応する少なくとも1つのアドレスの存在を確認する関数。 Perl構文(PCRE)の1つの必須patterns引数(検索パターン: 1つ(文字列)または複数(文字列の配列))の正規表現を受け入れます。 次のブール値を返します。 • true - 少なくとも1つのテンプレートに対応するアドレスが見つかった場合 • false - 少なくとも1つのテンプレートに対応するアドレスが見つからなかった場合 大文字と小文字を区別しません。	機能

フィールド	説明	データタイプ
all_match	すべてのアドレスがアドレス配列内の指定されたテンプレートの少なくとも1つに対応するかどうかをチェックする関数。 Perl構文(PCRE)の1つの必須patterns引数(検索パターン:1つ(文字列)または複数(文字列の配列))の正規表現を受け入れます。 次のブール値を返します。 • true - すべてのアドレスが少なくとも1つのテンプレートに完全に対応する場合 • false - いずれのテンプレートにも完全に対応するアドレスがない場合 大文字と小文字を区別しません。	機能
無効になったメタメソッド: *なし*		

テーブルMimeMessage

このテーブルでは、処理されているメールメッセージをまとめて指定します(MimePartテーブルと同じフィールドと、いくつかの追加情報が含まれます)。

フィールド	説明	データタイプ
dkim	メールメッセージのDKIM署名(RFC 6376を参照)	DKIMテーブル
raw	クライアントから受信したメールメッセージ	文字列
spam	スパムの兆候に対するメッセージスキャンの結果についてのレポート	テーブルSpam
from	Fromヘッダー値。メッセージにFromがない場合はnil	テーブルFrom
to	Toヘッダー値。メッセージにToがない場合はnil	テーブルTo
date	Dateヘッダー値。メッセージにDateがない場合はnil	文字列
message_id	Message-IDヘッダー値。メッセージにMessage-IDがない場合はnil	文字列
subject	Subjectヘッダー値。メッセージにSubjectがない場合はnil	文字列
user_agent	User-Agentヘッダー値。メッセージにUser-Agentがない場合はnil	文字列
vxcube_analysis	Dr.Web vxCubeからのメッセージ解析結果。このフィールドはSMTPモードのときにのみ存在します。	VxcubeAnalysisテーブルの配列
(次のフィールドはMimePartテーブルのフィールドと同様で、ルートMIME部分を指定します)		

フィールド	説明	データタイプ
無効になったメタメソッド: *なし*		

テーブル MimePart

このテーブルではメールメッセージの部分を指定します。

フィールド	説明	データタイプ
header	メッセージのヘッダー。	テーブル MimeHeader
body	該当部分の本体。添付された部品がある場合は nil。	テーブル MimeBody
part	テーブルの配列として(現在のメッセージ部分に)添付された部分。添付された部品がない場合、配列は空になります。	MimePart テーブルの配列
content_disposition	このヘッダーが該当部分にない場合、Content-Dispositionヘッダーのコンテンツ、または nil。	テーブル ContentDisposition
content_id	このヘッダーが該当部分にない場合、Content-IDヘッダーのコンテンツまたは nil。	文字列
content_type	このヘッダーが該当部分にない場合、Content-Typeヘッダーのコンテンツまたは nil。	テーブル ContentType
name	添付ファイル名。該当部分が添付ファイルでない場合は nil。	文字列
part_at	path引数(メッセージの子部分へのパス)を受け取る関数。指定されたパスにある添付されたメッセージ部分(テーブル MimePart)を返します。 pathは"/1/2/3"のような文字列となります。これは、root_part.part [1].part [2].part [3]を意味します。数字のない""、"/"、"//"のようなパスはメッセージ部分に対応し、この関数の呼び出し元になります(例:root_part)。指定したパスに子部分がない場合や、パスが正しくない場合は、関数は nilを返します。	機能
threats	オプションの引数として filterを取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分にある、指定された filter条件を満たすすべての脅威を調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • Virus テーブル。 • 検出された脅威を含むメッセージ部分への相対パス。 filter引数として、次のものを使用できます。 • ThreatFilter テーブル。	機能

フィールド	説明	データタイプ
	• Virus 引数のみを受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている（脅威である）場合 o `false` - 引数が条件（脅威である）を満たしていない場合	
`urls`	オプションの引数として `filter` を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分にある、指定された `filter` 条件を満たすすべてのURLを調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • Url テーブル。 • 見つかったURLを含むメッセージ部分への相対パス。 `filter` 引数として、次のものを使用できます。 • UrlFilter テーブル。 • Url 引数のみを受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている（不要である）場合 o `false` - 引数が条件（望ましくないものではない）を満たしていない場合	機能
`attachments`	オプションの引数として `filter` を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分にある、指定された `filter` 条件を満たすすべての添付ファイルを調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • MimePart テーブル。 • 見つかった添付ファイルを含むメッセージ部分への相対パス。 `filter` 引数として、次のものを使用できます。 • PartFilter テーブル。 • MimePart 引数のみを受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている場合 o `false` - 引数が条件を満たしていない場合	機能
`files`	オプションの引数として `filter` を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分（アーカイブを含む）にある、指定された `filter` 条件を満たすすべてのファイルを調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • 文字列で示したファイル名。 • 見つかったファイルを含むメッセージ部分への相対パス。 `filter` 引数として、次のものを使用できます。 • FileFilter テーブル。 • ファイル名を文字列として受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている場合	機能

フィールド	説明	データタイプ
	o `false` - 引数が条件を満たしていない場合	
`parts`	オプションの引数として`filter`を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分にある、指定された`filter`条件を満たすすべてのメッセージ部分を調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • MimePartテーブル。 • メッセージ部分への相対パス。 `filter`引数として、次のものを使用できます。 • PartFilterテーブル。 • MimePartテーブルを受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている場合 o `false` - 引数が条件を満たしていない場合	機能
`leaf_parts`	オプションの引数として`filter`を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分にある、指定された`filter`条件を満たすすべてのリーフ部分を調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • MimePartテーブル。 • メッセージ部分への相対パス。 `filter`引数として、次のものを使用できます。 • PartFilterテーブル。 • MimePartテーブルを受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている場合 o `false` - 引数が条件を満たしていない場合	機能
`text_parts`	オプションの引数として`filter`を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、メッセージのこの部分とその添付部分にある、指定された`filter`条件を満たすすべてのテキスト部分を調べることができます。反復子関数は引数を持たず、次の2つの値を返します。 • MimePartテーブル。 • メッセージ部分への相対パス。 `filter`引数として、次のものを使用できます。 • PartFilterテーブル。 • MimePartテーブルを受け取り、次のブール値を返す任意の叙述関数： o `true` - 引数が条件を満たしている場合 o `false` - 引数が条件を満たしていない場合	機能

フィールド	説明	データタイプ
scan_reports	オプションの引数として filter を取る関数。この関数は単一の値、つまり反復子関数を返します。この反復子を使用すると、指定された filter 条件を満たす、メッセージのこの部分とその添付部分のスキャンに関するすべてのレポートを調べることができます。 反復子関数には引数がなく、単一の値、つまり ScanReport テーブルを返します。 filter 引数として、次のものを使用できます。 • ScanReportFilter テーブル。 • ScanReport テーブルを受け取り、次のブール値を返す任意の叙述関数: o true - 引数が条件を満たしている場合 o false - 引数が条件を満たしていない場合	機能
has_url	オプションの引数として filter を取る関数(既述の urls 関数の説明を参照)。 次のブール値を返します。 • true - メッセージの該当部分とその添付された部分に条件 filter を満たす URL が含まれる場合。 • false - メッセージの該当部分と添付された部分のいずれにも、条件 filter を満たす URL が含まれていない場合。	機能

フィールド	説明	データタイプ
	例： `if ctx.message.has_url() then` `  -- at least one URL has been found in the email message` `end` `if ctx.message.has_url{category = "adult_content"} then` `  -- an adult content link has been found` `end` `if ctx.message.has_url{category = {"adult_content", "social_networks"}} then` `  -- an "adult_content" or a "social_networks" link has been found` `end` `if ctx.message.has_url{category = "black_list"} then` `  -- a link to a blacklisted resource been found` `end` `if ctx.message.has_url{host = "example.com"} then` `  end` `end` `if ctx.message.has_url{host_not = "*example.com"} then` `  -- a link detected with a host that does not correspond to the "*example.com" template` `end` `if ctx.message.has_url(function(url) return port > 80 end) then` `  -- a link whose port number is more than 80 has been found` `end`	
`has_threat`	オプションの引数として`filter`を取る関数（既述の`threats`関数の説明を参照）。 次のブール値を返します。 • `true` - メッセージの該当部分と添付ファイル部分に条件`filter`を満たす脅威が含まれる場合。 • `false` - メッセージの該当部分と添付された部分のいずれにも、条件`filter`を満たす脅威が含まれていない場合。	機能

フィールド	説明	データタイプ
	例： `if ctx.message.has_threat() then` `  -- the message contains at least one threat of any category` `end` `if ctx.message.has_threat({category = "known_virus"}) then` `  -- the message contains at least one threat of the "known_virus" category` `end` `if ctx.message.has_threat{category = "known_virus"} then` `  -- the same` `end` `if ctx.message.has_threat({category = {"known_virus", "joke"}}) then` `  -- the message contains at least one threat of the "known_virus" or "joke" category` `end` `if ctx.message.has_threat{category_not = "joke"} then` `  -- the message contains a threat of any category except "joke"` `end`	
has_file	オプションの引数として`filter`を取る関数（既述の`files`関数の説明を参照）。 次のブール値を返します。 • `true` - メッセージの該当部分とその添付ファイル部分に条件`filter`を満たすファイルが含まれる場合（アーカイブのファイルを含む）。 • `false` - メッセージの該当部分と添付された部分のいずれにも、条件`filter`を満たすURLが含まれていない場合。 例： `if ctx.message.has_file() then` `  -- at least one file has been found in the email message` `end` `if ctx.message.has_file{name = "*.exe"} then` `  -- at least one exe file has been found in the email message` `end`	機能

フィールド	説明	データタイプ
has_part	オプションの引数として filter を取る関数（既述の parts 関数の説明を参照）。 次のブール値を返します。 • true - メッセージの該当部分とその添付ファイル部分に条件 filter を満たす部分が含まれる場合。 • false - メッセージの該当部分と添付された部分のいずれにも、条件 filter を満たす部分が含まれていない場合。	機能
has_scan_report	オプションの引数として filter を取る関数（既述の scan_reports 関数の説明を参照）。 次のブール値を返します。 • true - メッセージの該当部分とその添付ファイル部分に条件 filter を満たすスキャンレポートが含まれる場合。 • false - メッセージの該当部分と添付された部分のいずれにも、条件 filter を満たすスキャンレポートが含まれていない場合。 使用例については、ScanReportFilter テーブルの説明を参照してください。	機能
search	正規表現（PCRE）を使用してこのメッセージセクション内のテキストを検索する関数。正規表現（文字列）を受け取ります。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。 次のブール値を返します。 • true - 指定した正規表現との一致がこのセクションまたは子部分で見つかった場合。 • false - 指定した正規表現との一致がこのセクションまたは子部分で見つからなかった場合。	機能
無効になったメタメソッド：*なし*		

テーブル From

このテーブルでは From メールメッセージヘッダーを指定します。ヘッダー（文字列の配列）から抽出されたメールアドレスのリストに加えて、次のフィールドも含まれます。

フィールド	説明	データタイプ
search	アドレス配列内の指定されたテンプレートの少なくとも1つに対応する少なくとも1つのアドレスの存在を確認する関数。 Perl構文（PCRE）の1つの必須 patterns 引数（検索パターン：1つ（文字列）または複数（文字列の配列））の正規表現を受け入れます。	機能

フィールド	説明	データタイプ
	次のブール値を返します。 • true - 少なくとも1つのテンプレートに完全に対応するアドレスが見つかった場合 • false - 少なくとも1つのテンプレートに完全に対応するアドレスが見つからなかった場合 大文字と小文字を区別しません。	
all_match	すべてのアドレスがアドレス配列内の指定されたテンプレートの少なくとも1つに対応するかどうかをチェックする関数。 Perl構文(PCRE)の1つの必須patterns引数(検索パターン:1つ(文字列)または複数(文字列の配列))の正規表現を受け入れます。 次のブール値を返します。 • true - すべてのアドレスが少なくとも1つのテンプレートに完全に対応する場合 • false - 少なくとも1つのテンプレートに完全に対応するアドレスがない場合 大文字と小文字を区別しません。	機能
無効になったメタメソッド: • __tostringは、デコードされたヘッダー値を返す関数です。 • __concatは、ヘッダーの復号化された値を文字列と連結する関数です。		

テーブルTo

このテーブルではToメールメッセージヘッダーを指定します。Fromテーブルと同じフィールドとメソッドが含まれています。

テーブルContentType

このテーブルではメッセージ部分のContent-Typeヘッダーを指定します。

フィールド	説明	データタイプ
type	メッセージ部分のMIMEタイプ	文字列
subtype	メッセージ部分のサブタイプ	文字列
param	次のフィールドを持つテーブル配列形式のヘッダーパラメータ: • nameはパラメータ名(文字列)です。 • valueはパラメータ値(文字列)です。	テーブル配列
無効になったメタメソッド: • __tostringは、デコードされたヘッダー値を返す関数です。		

フィールド	説明	データタイプ
• __concatは、ヘッダーの復号化された値を文字列と連結する関数です。		

テーブル ContentDisposition

このテーブルではメッセージ部分のContent-Dispositionヘッダーを指定します。

フィールド	説明	データタイプ
type	メッセージ部分のビュータイプ	文字列
param	次のフィールドを持つテーブル配列形式のヘッダーパラメータ: • nameはパラメータ名(文字列)です。 • valueはパラメータ値(文字列)です。	テーブル配列
無効になったメタメソッド: • __tostringは、デコードされたヘッダー値を返す関数です。 • __concatは、ヘッダーの復号化された値を文字列と連結する関数です。		

テーブル Spam

このテーブルには、指定されたメッセージのスパムチェックレポートが含まれます。

フィールド	説明	データタイプ
type	メッセージのタイプ(スパムのステータス)。以下の値を使用できます。 • "legit" - メッセージはスパムではありません。 • "spam" - メッセージはスパムです。 • "virus" - 他社製ヒューリスティックアナライザにより、メッセージ本文にウイルスが検出されました。 • "bounce" - メッセージには、元のメッセージの送信者に送信されたネガティブ配信確認(DSN)に関するレポートが含まれています。 • "suspicious" - 疑わしいメッセージ。 • "pce" - 有効なサブスクリプションサービスによって送信される「プロフェッショナルな」商用(広告)メッセージ。 • "mce" - 有効なサブスクリプションサービスでは送信されないが、サブスクリプションを解除する方法が含まれる商用(広告)メッセージ。 • "dce" - サブスクリプションを解除する方法のない「ダーティーな」コマーシャル(広告)メッセージ。 • "community" - ソーシャルネットワークからのメッセージ。 • "transactional" - トランザクション関連のメッセージ(登録、サービスまたは商品の購入)。 • "phishing" - 不正なメッセージ。 • "scam" - 不正なメッセージ(詐欺メッセージ)。	文字列
score	メッセージに割り当てられたスパム評価	番号

フィールド	説明	データタイプ
normalized_score	インターバル[0, 1]で正規化されたスパムスコア	番号
reason	メールメッセージがスパムであるとされた理由の説明を含む暗号化された文字列	文字列
version	サードパーティ製アンチスパムライブラリのバージョン	文字列
無効になったメタメソッド: *なし*		

テーブルVirus

このテーブルでは脅威を指定します。

フィールド	説明	データタイプ
type	脅威の種類（Doctor Webの分類による）: • "known_virus" - 既知の脅威（ウイルスデータベースに登録されている脅威）。 • "virus_modification" - 既知の脅威の亜種。 • "unknown_virus" - 未知の脅威、疑わしいオブジェクト。 • "adware" - 広告プログラム。 • "dialer" - ダイアラープログラム。 • "joke" - ジョークプログラム。 • "riskware" - 潜在的に危険なプログラム。 • "hacktool" - ハッキングツール。	文字列
name	脅威の種類（Doctor Webの分類による）	文字列
無効になったメタメソッド: *なし*		

テーブルUrl

URLを指定するテーブルです。

フィールド	説明	データタイプ
scheme	スキーム（プロトコル）プレフィックス。例: "http"	文字列
host	ホスト名またはIPアドレス。例: "example.com"	文字列
port	ポート番号。例: 80。URLに含まれていない場合、値はnilになります。	番号
path	リソースへのパス。例: "index.html"。URLに含まれていない場合、値はnilになります。	文字列

フィールド	説明	データタイプ
categories	スキャンの結果に基づいてURLを配置するカテゴリーの配列。以下の値を使用できます。 • "infection_source" - 感染源。 • "not_recommended" - 非推奨のWebサイト。 • "adult_content" - アダルトコンテンツ。 • "violence" - 暴力。 • "weapons" - 武器。 • "gambling" - ギャンブル。 • "drugs" - 薬物。 • "obscene_language" - 卑猥な表現。 • "chats" - チャット。 • "terrorism" - テロリズム。 • "free_email" - 無料メール。 • "social_networks" - ソーシャルネットワーク。 • "owners_notice" - 著作権者からの申し立てによってリストに登録されたWebサイト。 • "online_games" - オンラインゲーム。 • "anonymizers" - アノニマイザー。 • "cryptocurrency_mining_pools" - 仮想通貨マイニングプール。 • "jobs" - 求人検索サイト。 • "black_list" - ブラックリスト（メールサーバー管理者によって非推奨と見なされるリソース）。	文字列のテーブル
legal_url	URLがowners_noticeカテゴリーに属する場合、フィールドには所有者のWebサイトへのURLが含まれます。属さない場合はnilになります。	文字列
無効になったメタメソッド： • __toString - この関数は、Urlコンテンツを文字列（UTF-8）として返します。 • __concat - この関数は、URL文字列値と別の文字列を連結します。		

テーブルThreatFilter

このテーブルでは脅威のフィルターを指定します。フィールドはすべてオプションです。

フィールド	説明	データタイプ
category	脅威が該当すると予想されるカテゴリーのリスト（大文字と小文字は区別されません）。Virusテーブルのtypeフィールドの説明にあるカテゴリーのリストを参照してください。	文字列または文字列のテーブル
category_not	脅威が該当しないと予想されるカテゴリーのリスト（大文字と小文字は区別されません）。	文字列または文字列のテーブル

フィールド	説明	データタイプ
無効になったメタメソッド: *なし*		

フィルターフィールドが指定されていない（値が`nil`である）場合、脅威はフィルターと一致します。複数のフィルターフィールドが指定されている場合、条件は接続詞（論理積）によって結合されます。フィルターフィールドがテーブル（リスト）の場合、オブジェクトは少なくとも1つのテーブル（リスト）の項目と一致する必要があります。

使用例:

1. メッセージで検出された脅威の名前をすべてログに書き込む。

```
function milter_hook(ctx)

 …

 for virus in ctx.message.threats() do
  dw.notice("threat found: " .. virus.name)
 end

 …

end
```

2. カテゴリーフィルターに一致する脅威の名前と、脅威が検出されたメッセージ部分の名前をログに書き込む。

```
function milter_hook(ctx)

 …

 for v, p in ctx.message.threats({category = "known_virus"}) do
  dw.notice("found " .. v.name .. " in " ..
ctx.message.part_at(p).name(p))
 end

 …

end
```

3. 叙述関数に一致する脅威名と、脅威が検出されたメッセージ部分の名前をログに書き込む。

```
function milter_hook(ctx)

 …

 local function eicar_filter(v)
  return v.name == "EICAR Test File (NOT a Virus!)"
 end

 for v, p in ctx.message.threats(eicar_filter) do
  dw.notice("found " .. v.name .. " in " ..
ctx.message.part_at(p).name(p))
 end

 …

end
```

テーブルUrlFilter

このテーブルでは、URLに適用されるフィルターを指定します（既述のテーブルThreatFilterと同様）。フィールドはすべてオプションです。

フィールド	説明	データタイプ
category	URLが該当するカテゴリーのリスト（大文字と小文字は区別されません）。Urlテーブルのcategoriesフィールドの説明にあるカテゴリーのリストを参照してください。	文字列または文字列のテーブル
category_not	URLが該当しないカテゴリーのリスト（大文字と小文字は区別されません）。	文字列または文字列のテーブル
text	URLと一致しなければならないテキスト	文字列または文字列のテーブル
text_not	URLと一致できないテキスト	文字列または文字列のテーブル
host	URLに存在するホスト（ドメイン）	文字列または文字列のテーブル
host_not	URLにないホスト（ドメイン）	文字列または文字列のテーブル
無効になったメタメソッド：*なし*		

フィルターフィールドが指定されていない（値がnilである）場合、脅威はフィルターと一致します。複数のフィルターフィールドが指定されている場合、条件は接続詞（論理積）によって結合されます。フィルターフィールドがテーブル（リスト）の場合、オブジェクトは少なくとも1つのテーブル（リスト）の項目と一致する必要があります。

使用例:

1. メッセージ内に見つかったすべてのURLをログに書き込む。

```
function milter_hook(ctx)

 …

 for url in ctx.message.urls() do
  dw.notice("url found: " .. url)
 end

 …

end
```

2. カテゴリーに一致するURLと、URLが検出されたメッセージ部分の名前をログに書き込む。

```
function milter_hook(ctx)

 …

 for u, p in ctx.message.urls{category = "adult_content"} do
  dw.notice("found " .. u.text .. " in " ..
ctx.message.part_at(p).name(p))
 end

 …

end
```

テーブルFileFilter

このテーブルでは、ファイルに適用されるフィルターを指定します(既述のテーブルThreatFilterと同様)。フィールドはすべてオプションです。

フィールド	説明	データタイプ
name	ファイル名が一致すると予想される文字セットまたはワイルドカード(例:"*.exe"、"eicar.txt")。 大文字と小文字を区別しません。	文字列または文字列のテーブル
name_re	ファイル名が一致すると予想される正規表現(PCRE)(例:".*\\.zip"、[[.*\.zip]])。 大文字と小文字を区別しません。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。	文字列または文字列のテーブル
name_not	ファイル名が一致しないと予想される文字セットまたはワイルドカード。 大文字と小文字を区別しません。	文字列または文字列のテーブル

フィールド	説明	データタイプ
name_re_not	ファイル名が一致しないと予想される正規表現（PCRE）。 大文字と小文字を区別しません。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。	文字列または文字列のテーブル
無効になったメタメソッド：*なし*		

複数のフィルターフィールドが指定されている場合、条件は接続詞（論理積）によって結合されます。フィルターフィールドがテーブル（配列）の場合、オブジェクトは少なくとも1つのテーブル（配列）の項目と一致する必要があります。フィルターフィールドが指定されていない（値が`nil`である）場合、すべてのファイルがフィルターと一致します。

使用例：

`.exe`拡張子を持つファイルを含む、該当部分の名前をログに出力する。

```
function milter_hook(ctx)

 …

 for f, p in ctx.message.files{name = "*.exe"} do
  local where = ctx.message.part_at(p).name
  if not where or where == "" then where = p end
  dw.notice("EXE found in " .. where)
 end

 …

end
```

テーブルPartFilter

このテーブルではメッセージ部分のフィルターを指定します（既述のテーブルFileFilterと同様）。フィールドはすべてオプションです。

フィールド	説明	データタイプ
name	該当部分の名前が一致すると予想される文字セットまたはワイルドカード（例：`"*.exe"`、`"eicar.txt"`）。 大文字と小文字を区別しません。	文字列または文字列のテーブル
name_re	該当部分の名前が一致すると予想される正規表現（PCRE）（例：`".*\\.zip"`、`[[.*\.zip]]`）。 大文字と小文字を区別しません。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。	文字列または文字列のテーブル
content_type	該当部分の`Content-Type`の値が一致すると予想される文字セット（例：`"image/*"`）。 大文字と小文字を区別しません。	文字列または文字列のテーブル

フィールド	説明	データタイプ
content_disposition	該当部分（添付ファイル）のContent-Dispositionの値が一致すると予想される文字セット（例："inline"、"attachment"）。 大文字と小文字を区別しません。	文字列または文字列のテーブル
name_not	ファイル名が一致しないと予想される文字セット。 大文字と小文字を区別しません。	文字列または文字列のテーブル
name_re_not	該当部分の名前が一致しないと予想される正規表現。 大文字と小文字を区別しません。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。	文字列または文字列のテーブル
content_type_not	該当部分のContent-Typeの値が一致しないと予想される文字セット。 大文字と小文字を区別しません。	文字列または文字列のテーブル
content_disposition_not	該当部分のContent-Dispositionの値が一致しないと予想される文字セット。 大文字と小文字を区別しません。	文字列または文字列のテーブル
無効になったメタメソッド：*なし*		

フィルターフィールドが指定されていない（値がnilである）場合、任意の部分（添付ファイル）がフィルターと一致します。複数のフィルターフィールドが指定されている場合、条件は接続詞（論理積）によって結合されます。フィルターフィールドがテーブル（配列）の場合、オブジェクトは少なくとも1つのテーブル（配列）の項目と一致する必要があります。

使用例：

1. すべての添付ファイルとそのMD5ハッシュをログに書き込む。

```
function milter_hook(ctx)

 …

 for a, p in ctx.message.attachments() do
  -- the attachment name be an empty string
  -- if it is not specified in Content-Type and in Content-Disposition
  local name = a.name
  if name == "" then name = "at path " .. p end
  dw.notice("Attachment: " .. name .. "; md5=" .. a.body.md5)
 end

 …

end
```

2. .exe拡張子を持つすべての添付ファイルをログに書き込む。

```
function milter_hook(ctx)

 …

 for a in ctx.message.attachments{name = "*.exe"} do
  dw.notice("EXE attachment: " .. a.name)
 end

 …

end
```

3. メールメッセージにある画像をカウントし、種類別にソートする。

```
function milter_hook(ctx)

 …

 local images = {}
 for part, path in ctx.message.parts{content_type = "image/*"} do
  local subtype = part.content_type.subtype
  images[subtype] = (images[subtype] or 0) + 1
 end

 for t, c in pairs(images) do
  dw.notice("Found " .. t .. " images: " .. c)
 end

 …

end
```

4. 添付ファイルにあるオーディオファイルのリストをログに書き込む。

```
function milter_hook(ctx)

 …

 for p, path in ctx.message.parts{
  content_type = "audio/*",
  content_disposition = {"inline", "attachment"}
 } do
    local name = p.name
    if name == "" then name = "<unnamed>" end
    dw.notice("Audio file: " .. name)
 end

 …

end
```

テーブル ScanReportFilter

このテーブルでは、脅威のメッセージ部分をスキャンするレポートのフィルターを指定します（既述のテーブル FileFilter と同様）。次のフィールドが含まれます（フィールドはすべてオプションです）。

フィールド	説明	データタイプ
`error`	ScanReport に追加されるエラーの名前（例：`"password_protected"`、`"scan_timeout"`）。 大文字と小文字を区別しません。	文字列または文字列のテーブル
`error_not`	ScanReport に追加されてはならないエラーの名前（例：`"password_protected"`、`"scan_timeout"`）。 大文字と小文字を区別しません。	文字列または文字列のテーブル
無効になったメタメソッド：*なし*		

複数のフィルターフィールドが指定されている場合、条件は接続詞（論理積）によって結合されます。フィルターフィールドがテーブル（配列）の場合、スキャンレポートは少なくとも1つのテーブル（配列）の項目と一致する必要があります。フィルターフィールドが指定されていない（値が`nil`である）場合、スキャンレポートはすべてフィルターに一致します。

使用例：

1. パスワードで保護されたアーカイブのスキャンに失敗した場合に、メッセージを隔離する。

```
function milter_hook(ctx)

 …

 if ctx.message.has_scan_report{error = 'password_protected'} then
   return
    {
      action = 'accept', deleted_recipients = ctx.to,
      added_recipients = {'quarantine@mail.domain.com'}
    }
 end

 …

end
```

2. スキャン制限を超えている場合にメッセージを隔離する。

```
function milter_hook(ctx)

 …

 local limit_errors = {
  'archive_level_limit', 'compression_limit',
  'container_level_limit', 'mail_level_limit',
  'packer_level_limit', 'report_size_limit'
 }
```

```
 if ctx.message.has_scan_report{error = limit_errors} then
   return
    {
      action = 'accept', deleted_recipients = ctx.to,
      added_recipients = {'quarantine@mail.domain.com'}
    }
 end

 …

end
```

3. スキャンエラーがある場合にメッセージを拒否する。

```
function milter_hook(ctx)

 …

 if ctx.message.has_scan_report{error = '*'} then
   return {action = 'reject'}
 end

 …

end
```

テーブルMimeHeader

このテーブルではメッセージ部分のヘッダーを指定します。

フィールド	説明	データタイプ
`field`	ヘッダーとその値のリスト	HeaderFieldテーブルの配列
`search`	正規表現(PCRE)でヘッダーを検索する関数。引数として正規表現(文字列)を取ります。検索は、メッセージ部分のすべてのヘッダーで行われます。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。 次のブール値を返します。 • `true` - `field.name .. ": " .. field.value.decoded`文字列が、少なくとも1つのヘッダーに対して指定した正規表現と一致する場合 • `false` - `field.name .. ": " .. field.value.decoded`文字列が、少なくとも1つのヘッダーに対して指定した正規表現と一致しない場合	機能
`value`	指定したヘッダーの値を返す関数。引数としてヘッダーの名前(文字列)を取ります。 この関数は、指定した名前を持つ最初に見つかったヘッダーに対応するHeaderFieldValueテーブルを返します。ヘッダーが見つからなかった場合は、`nil`を返します。	機能

フィールド	説明	データタイプ
無効になったメタメソッド: *なし*		

テーブル HeaderField

このテーブルではメッセージ部分のヘッダーを指定します。

フィールド	説明	データタイプ
name	ヘッダー名	文字列
value	ヘッダー値	テーブル HeaderFieldValue
url	ヘッダー値で見つかったURLのリスト（Subjectヘッダーのみ）。それ以外のヘッダーではnil	Urlテーブルの配列
無効になったメタメソッド: *なし*		

テーブル HeaderFieldValue

このテーブルではメールメッセージヘッダーの値を指定します。

フィールド	説明	データタイプ
raw	未加工の（デコードされていない）ヘッダー値	文字列
decoded	デコードされたヘッダー値	文字列
無効になったメタメソッド: • __toString - この関数は、HeaderFieldValueの内容（decodedフィールドの値）を文字列として返します。 • __concat - この関数は、HeaderFieldValue（decodedフィールドの値）と別の文字列を連結します。		

テーブル MimeBody

このテーブルではメッセージ部分の本文を指定します。

フィールド	説明	データタイプ
raw	メッセージ部分の未加工の（デコードされていない）本文	文字列
decoded	メッセージ部分本体のデコードされた値（Content-Transfer-Encodingヘッダーと Content-Typeヘッダーの値に応じたもの）	文字列
text	Content-Typeヘッダーの charsetパラメータに従って、UTF-8でデコードされたメッセージ部分本体の値。 "Content-Type: text/*"を持つ部分または Content-Typeが空の部分にのみ存在します。それ以外の場合は、nilになります。	文字列

フィールド	説明	データタイプ
scan_report	脅威のスキャンレポート	テーブル ScanReport
url	部分テキストに見つかったURL。Urlテーブルの配列。 textフィールドがない場合（nil）、フィールドは空になります。	Urlテーブルの配列
search	正規表現（PCRE）を使用してこの本文内のテキストを検索する関数。引数として正規表現（文字列）を取ります。引用符内で文字列を使用する場合は、スラッシュ文字をエスケープする必要があります。 次のブール値を返します。 • true - 本文がテキストであり、一致が見つかった場合 • false - 本文に一致するものが見つからなかった場合	機能
md5	メールメッセージ本文のMD5ハッシュ。	文字列
sha1	メールメッセージ本文のSHA1ハッシュ。	文字列
sha256	メールメッセージ本文のSHA256ハッシュ。	文字列
vxcube_analysis	Dr.Web vxCubeからのメッセージ解析結果。このフィールドはSMTPモードのときにのみ存在します。	VxcubeAnalysisテーブルの配列
無効になったメタメソッド: *なし*		

テーブルScanReport

このテーブルには、脅威のスキャンについてのレポートが含まれます。

フィールド	説明	データタイプ
object	スキャンされたオブジェクトの名前	文字列
archive	スキャンされたオブジェクトがコンテナの場合は、コンテナに関する情報。オブジェクトがコンテナではない場合、nil	Archiveテーブル
virus	検出された脅威のリスト	Virusテーブルの配列
error	エラーが発生した場合は、スキャンエラーの文字列が含まれます。発生していない場合は、nilになります。使用可能な値: • "path_not_absolute" - 指定されたパスは絶対パスではありません。 • "file_not_found" - ファイルが見つかりませんでした。 • "file_not_regular" - ファイルは通常のファイルではありません。 • "file_not_block_device" - ブロックデバイスではありません。 • "name_too_long" - 名前が長すぎます。 • "no_access" - アクセスが拒否されました。	文字列

フィールド	説明	データタイプ
	• "read_error" - 読み取りエラーが発生しました。 • "write_error" - 書き込みエラー。 • "file_too_large" - ファイルが大きすぎます。 • "file_busy" - ファイルは使用中です。 • "unpacking_error" - アンパックエラー。 • "password_protected" - アーカイブはパスワードで保護されています。 • "arch_crc_error" - CRCアーカイブエラー。 • "arch_invalid_header" - 無効なアーカイブヘッダー。 • "arch_no_memory" - アーカイブを解凍するための十分なメモリがありません。 • "arch_incomplete" - 不完全なアーカイブ。 • "can_not_be_cured" - ファイルを修復できません。 • "packer_level_limit" - 圧縮されたオブジェクトのネスティングレベルの上限を超えました。 • "archive_level_limit" - アーカイブのネスティングレベルの上限を超えました。 • "mail_level_limit" - メールファイルのネスティングレベルの上限を超えました。 • "container_level_limit" - コンテナのネスティングレベルの上限を超えました。 • "compression_limit" - 圧縮率の上限を超えました。 • "report_size_limit" - レポートサイズの上限を超えました。 • "scan_timeout" - スキャンタイムアウトの上限を超えました。 • "engine_crash" - スキャンエンジンの障害。 • "engine_hangup" - スキャンエンジンのハングアップ。 • "engine_error" - スキャンエンジンエラー。 • "no_license" - アクティブなライセンスが見つかりません。 • "multiscan_too_late" - マルチスキャンエラー。 • "curing_limit_reached" - 修復試行の上限を超えました。 • "non_supported_disk" - サポートされていないディスクタイプ。 • "unexpected_error" - 予期しないエラー。	
item	コンテナ添付ファイルのスキャンに関するレポート（オブジェクトがコンテナである場合、すなわちアーカイブ、添付されたMIMEオブジェクトなど）	ScanReportテーブルの配列
無効になったメタメソッド: *なし*		

Archiveテーブル

このテーブルではアーカイブとその他の複合オブジェクトを指定します。

フィールド	説明	データタイプ
type	アーカイブの種類: • "archive" - アーカイブ。 • "mail"- メールファイル。 • "container" - その他のコンテナ。	文字列
name	アーカイブ名。例: ZIP	文字列
無効になったメタメソッド: *なし*		

テーブルDKIM

このテーブルでは、メッセージ内のすべてのDKIM署名を指定します。

フィールド	説明	データタイプ
signature	メッセージに存在するDKIM署名のリスト	DKIMSignature テーブルの配列
has_valid_signature	引数としてfilterを取り、以下を返す関数 • DKIMSignatureテーブルの形式で"pass"に等しいスキャン結果を持つ、最初に見つかったDKIM署名。 • 検証で署名が検出されなかった場合、nilが表示されます。 filter引数として、次のものを使用できます。 • DKIMSignatureFilterテーブル。 • DKIMSignature引数のみを受け取り、ブール値を返す任意の叙述関数: o true - 引数が条件を満たしている場合 o false - 引数が条件を満たしていない場合	機能
無効になったメタメソッド: *なし*		

テーブルDKIMSignature

このテーブルでは、メッセージ内の各DKIM署名のプロパティを指定します。

フィールド	説明	データタイプ
auid	DKIM署名の"i"タグから取得される*エージェントまたはユーザー識別子(AUID)*の値は、デフォルト値を考慮します	文字列
data	DKIM署名の"b"タグから抽出された署名のテキスト(*base64*)値	文字列
result	DKIM署名検証結果	DKIMResultテーブル
sdid	DKIM署名の"d"タグから取得した*署名ドメイン識別子(SDID)*の値	文字列
selector	DKIM署名の"s"タグから取得した*セレクター*	文字列

フィールド	説明	データタイプ
無効になったメタメソッド: *なし*		

テーブルDKIMResult

このテーブルでは、メッセージのすべてのDKIM署名の結果を指定します。

フィールド	説明	データタイプ
`type`	テキスト形式のDKIM署名検証結果。次の値が含まれる場合があります。 • `pass`は、署名の検証が成功したことを示します。 • `fail`は、署名の検証が失敗したことを示します(つまり、メール本文のハッシュと一致しないか、署名を検証できませんでした)。 • `neutral`は、DKIM署名の構文エラーです。 • `temperror`は、ドメインキーの取得に失敗したことを示します(DNSエラー)。 • `permerror`は、他のタイプのエラー(署名形式、キー形式、キーと署名の間の不整合など)を示します。	文字列
`comment`	スキャン結果のコメント(`Authentication-Results`のコメントとして使用できます)	文字列
`key_size`	スキャン中に使用されるキーサイズは`nil`か、キーまたはスキャン結果の取得に失敗した場合は`neutral`	番号
無効になったメタメソッド: *なし*		

テーブルDKIMSignatureFilter

このテーブルではDKIMメッセージ署名のフィルターを指定します(既述のテーブルFileFilterと同様)。フィールドはすべてオプションです。

フィールド	説明	データタイプ
`domain`	DKIM署名のSDIDフィールドのドメインが一致すると予想される文字セットまたはワイルドカード	文字列または文字列のテーブル
`domain_not`	DKIM署名のSDIDフィールドのドメインが一致しないと予想される文字セットまたはワイルドカード	文字列または文字列のテーブル
無効になったメタメソッド: *なし*		

フィルターフィールドが指定されていない場合(つまり、`nil`値が含まれている場合)、このメッセージのDKIM署名はすべてフィルターと一致します。複数のフィルターフィールドが指定されている場合、条件は接続詞(論理積)によって結合されます。フィルターフィールドタイプがテーブル(配列)の場合、フィルターされたオブジェクトは、テーブル(配列)要素の少なくとも1つと一致する必要があります。

テーブル SPF

このテーブルには、SPFをチェックするために必要なすべてのデータが含まれます。

フィールド	説明	データタイプ
`helo`	HELOチェックの結果	テーブル SPFResult
`from`	MAIL FROMチェックの結果	テーブル SPFResult
`check()`	補助関数。最初にMAIL FROMチェックを実行し、(判定が受信されなかった場合は)次にHELOチェックを実行します。この関数は文字列としてチェックの結果を返します。これは、SPFResult***テーブルの`status`フィールドの値と同じ値になる場合があります。 この関数は、生成された`Authentication-Results`ヘッダーを介して結果をOpenDMARCに送信するように設計されています。	機能
無効になったメタメソッド: *なし*		

テーブル SPFResult

SPFのチェック結果を含むテーブル。

フィールド	説明	データタイプ
`status`	文字列として表されるチェックの結果。`none`、`neutral`、`pass`、`fail`、`softfail`、`temperror`、`permerror`のいずれかの値を取ることができます(RFC 7208に準拠します)。	文字列
`explanation`	`fail`のレスポンスを受け取った場合の結果の説明。	文字列、または説明を受け取らなかった場合や説明が得られなかった場合は`nil`
無効になったメタメソッド: *なし*		

テーブル VxcubeAnalysis

このテーブルにはDr.Web vxCubeのオブジェクト解析の結果が含まれます。

フィールド	説明	データタイプ
`filename`	解析された添付ファイルの名前	文字列
`id`	解析ID	文字列
`sample_id`	解析されたファイルのID	文字列

フィールド	説明	データタイプ
format_name	解析されたファイルのフォーマット	文字列
tasks	解析結果	VxcubeTaskテーブルの配列
max_maliciousness	VxcubeTaskテーブルの配列からのmaliciousnessフィールドの最大値。0から100までの浮動小数で、エラーが発生した場合はnil	数字またはnil
無効になったメタメソッド: *なし*		

テーブルVxcubeTask

このテーブルには、Dr.Web vxCubeの特定のプラットフォームで実行されたオブジェクト解析の結果が含まれます。テーブルの構造は、Dr.Web vxCube APIを介して受け取ったTaskFinishedオブジェクトとほぼ同等です。

フィールド	説明	データタイプ
id	タスクID。	文字列
status	"failed"や"finished"などの解析ステータス。	文字列
platform_code	解析が行われたプラットフォームのコード。エラーがあった場合はnil。	文字列またはnil
maliciousness	オブジェクトの悪質性。0から100の浮動小数、またはエラーがあった場合はnil。	数字またはnil
verdict	*<category><degree>*の形式で表される、3つのカテゴリーのいずれかに対応するファイルの総合的な悪意性スコア。*<category>*は"neutral"、"suspicious"、"malware"の値のいずれかであり、*<degree>*は悪質性の程度(1～3)を表す整数です。判定の値は"malware1"、"suspicious3"などになります。	文字列
無効になったメタメソッド: *なし*		

利用可能な補助モジュール

LuaのプログラムスペースでDr.Web for UNIX Mail Serversとやり取りするために、次の特定のモジュールをインポートできます。

モジュール名	機能
drweb	Luaプログラムを起動したDr.Web for UNIX Mail ServersコンポーネントとLuaプロシージャの非同期実行の手段のログに、Luaプログラムからのメッセージを記録する機能を提供します。
drweb.lookup	Dr.Web LookupDモジュールを呼び出して、外部ソースからデータを要求するためのツールを提供します。
drweb.dnsxl	ホストのアドレスがDNSxLブラックリストにあるかどうかを確認するためのツールを提供します。
drweb.regex	文字列と正規表現を一致させるためのインターフェースを提供します。
drweb.subprocess	外部アプリケーション(プロセス)を実行するためのインターフェースを提供します。
drweb.config	Dr.Web MailD設定パラメータ値を持つテーブルを提供します。
drweb.store	MailDが実行される合間にデータを保存するための機能を提供します。

drwebモジュールの内容

1. 機能

このモジュールには、次のような機能があります。

- Luaプログラムからのメッセージを Dr.Web for UNIX Mail Serversコンポーネントログに保存する:
  - `log(`*<level>*`,` *<message>*`)` は *<message>* 文字列をDr.Web for UNIX Mail Serversログに *<level>* レベル(必要なレベルは、*「debug」*、*「info」*、*「notice」*、*「warning」*、*「error」*を使用して定義します)で書き込みます。
  - `debug(`*<message>*`)` は *<message>* 文字列をDr.Web for UNIX Mail Serversログに *DEBUG* レベルで書き込みます。
  - `info(`*<message>*`)` は *<message>* 文字列をDr.Web for UNIX Mail Serversログに *INFO* レベルで書き込みます。
  - `notice(`*<message>*`)` は *<message>* 文字列をDr.Web for UNIX Mail Serversログに *NOTICE* レベルで書き込みます。
  - `warning(`*<message>*`)` は *<message>* 文字列をDr.Web for UNIX Mail Serversログに *WARNING* レベルで書き込みます。
  - `error(`*<message>*`)` は *<message>* 文字列をDr.Web for UNIX Mail Serversログに *ERROR* レベルで書き込みます。
- Luaプロシージャの同期を管理する:
  - `sleep(`*<sec.>*`)` はこのLuaプロシージャインスタンスの実行を指定された秒数で一時停止します。
  - `async(`*<Lua function>*`[,` *<argument list>*`])` は、指定された関数を非同期的に起動し、指定され

た引数リストに渡します。async関数呼び出しはすぐに完了し、戻り値（Futureテーブル）を使用すると、*<Lua function>*の結果を取得できます。

- IpAddressテーブルにIPアドレスを追加する：
  - ip(*<address>*）は、IpAddressテーブルの形式で*<address>*文字列として送信される、IPアドレスを指定します。IPv4またはIPv6アドレスのいずれかを使用できます。
- テキストファイルから外部データをアップロードする：
  - load_set(*<file path>*）は、指定されたテキストファイルのコンテンツからtrue値を含むテーブルを生成します。ファイルから読み取られた文字列はキーとして使用されます。空の文字列と空白を含む文字列は無視されます。
  - load_array(*<file path>*）は、指定されたテキストファイルのコンテンツから文字列の配列を生成します。空の文字列と空白文字のみで構成される文字列は無視され、配列には含まれません。

2. テーブル

- Futureテーブルは、async関数を使用して関数を実行した後の保留中の結果を表します。

フィールド	説明	データタイプ
wait	async関数を使用して開始した関数の結果を返す関数。関数がまだ実行を完了していない場合は、完了を待って結果を返します。waitが呼び出される前に関数が完了した場合、結果はすぐに返されます。開始された関数が失敗した場合、wait呼び出しは同じエラーを生成します。	機能
無効になったメタメソッド：*なし*		

- IpAddressテーブルはIPアドレスを表します。

フィールド	説明	データタイプ
belongs	IpAddressテーブルに保存されているIPアドレスが、指定されたサブネット（IPアドレス範囲）に所属しているかどうかを確認する関数 "*<IP address>*"または"*<IP address>*/*<mask>*"のような文字列を唯一の引数として受け取ります。ここで、*<IP address>*はホストアドレスまたはネットワークアドレス（"127.0.0.1"など）、*<mask>*はサブネットワークマスク（"255.0.0.0"などのIPアドレスとして、または"8"などの数値形式で指定できます）です。 次のブール値を返します。 • trueは、アドレスが指定されたアドレスの少なくとも1つと等しいか、指定されたサブネット（IPアドレスの範囲）の少なくとも1つに属していることを示します。 • false - それ以外の場合。	機能
無効になったメタメソッド： • __tostringは、文字列内のIpAddressを変更する関数（例："127.0.0.1"（IPv4）または"::1"（IPv6））です。 • __concatは、IpAddressを文字列に結合する関数です。 • __eqは、2つのIpAddressが等しいことを確認する関数です。		

フィールド	説明	データタイプ
• __bandは、マスクを適用するための関数（例：dw.ip('192.168.1.2') & dw.ip('255.255.254.0')）です。		

3. 例

- 非同期的に開始される手順によって生成されるメッセージをログへ書き込む：

```
local dw = require "drweb"

-- This function waits two seconds and returns a string,
-- received as an argument
function out_msg(message)
 dw.sleep(2)
 return message
end

-- "Main" function
function intercept(ctx)
 -- Output of a string at the NOTICE level to the Dr.Web for UNIX Mail
Servers log
 dw.notice("Intercept function started.")

 -- An asynchronous start of two copies of the out_msg function
 local f1 = dw.async(out_msg, "Hello,")
 local f2 = dw.async(out_msg, " world!")

 -- Waiting for the completion of the copies of the function
 -- out_msg and output its results to log
 -- the Dr.Web for UNIX Mail Servers log at the DEBUG level
 dw.log("debug", f1.wait() .. f2.wait())
end
```

- スケジュールされた手順を作成する：

```
local dw = require "drweb"

-- Save the table Future in the future global variable in order
-- to preven the removal by the garbage collector
future = dw.async(function()
   while true do
     -- Everyday, the following message is displayed in the log
     dw.sleep(60 * 60 * 24)
     dw.notice("A brand new day began")
   end
end)
```

- 文字列で表現されたIPアドレスをIpAddressテーブルに変更する：

```
local dw = require "drweb"

local ipv4 = dw.ip("127.0.0.1")
local ipv6 = dw.ip("::1")
local mapped = dw.ip("::ffff:127.0.0.1")
```

drweb.lookupモジュールの内容

1. 機能

このモジュールは次の機能を提供します。

- `lookup(<request>, <parameters>)`は、Dr.Web LookupDモジュールを介して利用できる外部ストレージからデータを要求します。*<request>*引数は、Dr.Web LookupD設定内のセクション(文字列*<type>@<tag>*)に対応している必要があります。*<parameters>*引数は任意で、リクエストを生成するために使用される置換を表します。以下の自動的に許可されるマーカーを使用できます。
  - `$u`、`$U`は、クライアントコンポーネントによって送信されたユーザー名(`user`)に自動的に置き換えられます。
  - `$d`、`$D`は、クライアントコンポーネントによって送信されたドメイン(`domain`)に自動的に置き換えられます。

  これらの引数はテーブルとして設定されます。このテーブルのキーと値は文字列でなければなりません。この関数は、リクエストの結果である文字列の配列を返します。
- `check(<checked string>、<request>、<parameters>)`は、Dr.Web LookupDモジュールを介して利用できる外部リポジトリで*<checked string>*が見つかった場合に`true`を返します。引数*<request>*および*<parameters>*は`lookup`関数の引数と同じです(上記を参照)。*<checked string>*引数は、文字列または`__tostring`メタメソッドを持つテーブル(つまり、文字列にフォーマットできる)であると想定されます。

2. 例

- `LookupD.LDAP.users`データソースから取得したユーザーのログリストへ書き込む:

```
local dw = require "drweb"
local dwl = require "drweb.lookup"

-- "Main" function
function intercept(ctx)
 -- Writing the string at the NOTICE level to the Dr.Web for UNIX Mail
Servers log
 dw.notice("Intercept function started.")

 -- Writing the request results to the Dr.Web for UNIX Mail Servers log
 -- to the 'ldap@users' data source
 for _, s in ipairs(dwl.lookup("ldap@users", {user="username"})) do
   dw.notice("Result for request to 'ldap@users': " .. s)
 end

end
```

drweb.dnsxlモジュールの内容

1. 機能

このモジュールは次の機能を提供します。

- `ip(<IP address>, <DNSxL server>)`は指定されたIPアドレス*<IP address>*に対応するDNSxLサーバー*<DNSxL server>*からAタイプのDNSレコードをリクエストします。

検査されているIPアドレスがDNSxLサーバーのリストに登録されている場合、結果は架空のIPアドレスのリストになります。さらに、返された架空のIPアドレスのそれぞれに、検査済みの *<IP address>* がこのサーバーのリストに表示されている理由が含まれていることがあります（通常、理由タイプは返された架空のIPアドレスの最後のオクテット値によって決まります）。DNSxLサーバーにIPアドレス *<IP address>* に対応するAタイプのDNSレコードが含まれていない場合、関数は `nil` を返します。

- `url(`*<URL>, <SURBL server>*`)` は *<URL>* ドメイン部分に対応する *<SURBL server>* サーバーにAタイプのDNSレコードをリクエストします（HTTPリダイレクトは処理されません）。

  <URL> から取得された、検査されているドメインがSURBLサーバーのサーバーリストに登録されている場合、結果は架空のIPアドレスのリストになります。さらに、返された架空のIPアドレスのそれぞれに、検査済みのドメインがこのサーバーのリストに表示されている理由が含まれていることがあります（通常、理由タイプは返された架空のIPアドレスの最後のオクテット値によって決まります）。SURBLサーバーに *<URL>* からのドメインに対応するAタイプのDNSレコードが含まれていない場合、この関数は `nil` を返します。

関数の引数は、文字列または文字列にキャストされるオブジェクトです（たとえば、*<IP address>* としては IpAddress テーブルを使用でき、*<URL>* としては `Url` テーブルを使用できます）。IPアドレスは、IpAddress テーブルの配列として返されます。

2. テーブル

- IpAddressテーブルではIPアドレスを指定します。このテーブルの説明については上記を参照してください。

3. 例

- DNSxLサーバーによるIPアドレスのスキャン結果をログへ出力する。

```
local dw = require "drweb"
local dwxl = require "drweb.dnsxl"

-- "Main" function
function intercept(ctx)
 -- Output of a string at the NOTICE level to the Dr.Web for UNIX Mail
Servers log
 dw.notice("Intercept function started.")

 -- Output of the scanning results to the Dr.Web for UNIX Mail Servers log
 -- 10.20.30.40 IP addresses are in the DNSxL server black list
 -- dnsxl.server1.org
 local records = dwxl.ip("10.20.30.40", "dnsxl.server1.org")
 if records then
   for _, ip in ipairs(records) do
    dw.notice("DNSxL A record for 10.20.30.40: " .. ip)
   end
 end

end
```

drweb.regexモジュールの内容

1. 機能

このモジュールは次の機能を提供します。

- `search(<template>, <text>[, <flags>])` - <text>文字列に<template>正規表現と一致するサブストリングが含まれている場合は`true`を返します。オプションの<flags>パラメータ（整数）は、関数の動作に影響を与える一連のフラグであり、論理和でつなげられます。
- `match(<template>, <text>[, <flags>])` - <template>正規表現がそのサブストリングだけでなく<text>ストリング全体と一致しなければならない点を除いて`search`と同じです。

2. 利用可能なフラグ

- `ignore_case`はテキストの大文字と小文字を区別しません。

3. 例

```
local rx = require "drweb.regex"

rx.search("te.?t", "some TexT") -- false
rx.search("te.?t", "some TexT", rx.ignore_case) -- true

rx.match("some.+", "some TexT") -- true
```

drweb.subprocessモジュールの内容

1. 機能

このモジュールは次の機能を提供します。

- `run(<parameters>)`は指定されたプロセス（アプリケーション）を同期モードで実行します（この関数はプロセスが終了した後にのみコントロールを戻します）。<parameters>引数は、ファイルの実行パス、起動時にアプリケーションに渡されるすべての引数（配列*argv*）、アプリケーションの入出力ストリーム（*stdin*、*stdout*、*stderr*）に関連付けられたオプションのパラメータを含むテーブルです。オプションのパラメータは、アプリケーションの（現在の）作業ディレクトリと環境変数を指定します。

  関数の結果は、終了後のプロセス操作の結果（プロセスが終了した終了コードまたはシグナル番号）を含むテーブルになります。さらに、プロセス実行パラメータのテーブルで指定すると、返されるテーブルに、*stdout*および*stderr*出力ストリームから読み込まれるデータを含むフィールドを含めることができます。

2. テーブル

- 入力実行パラメータのテーブル

フィールド	説明	データタイプ
（名前なし）	ファイルパスとアプリケーション実行引数（配列*argv*）。 必須フィールド。フィールドはコマンドライン引数の数だけ繰り返され、最初の値は*argv[0]*、つまり実行パスに対応します。	文字列

フィールド	説明	データタイプ
stdin	実行後にアプリケーション（プロセス）が入力ストリーム（*stdin*）から受け取るテキスト。オプションのフィールド（指定しない場合、*stdin*は何も受け取りません）。	文字列
stdout	返されるテーブルのフィールドの名前。このフィールドは、プロセスが*stdout*ストリームに出力するテキストを受け取ります。オプションのフィールド（指定しない場合、出力は*stdout*に保存されません）。	文字列
stderr	返されるテーブルのフィールドの名前。このフィールドは、プロセスが*stderr*ストリームに出力するテキストを受け取ります。オプションのフィールド（指定しない場合、出力は*stderr*に保存されません）。	文字列
env	プロセス環境に送信される環境変数をフィールドに持つテーブル。環境変数は、"*<variable name>*"="*<value>*"のペアで指定されます。オプションのフィールド（指定しない場合、環境変数は設定されません）。	テーブル
workdir	実行中のプロセスの作業（現在の）ディレクトリ。オプションのフィールド（指定しない場合、作業ディレクトリは設定されません）。	文字列

- 実行結果のテーブル（run関数の戻り値）

フィールド	説明	データタイプ
exit_status	プロセスが正常に終了したときの返りコード。発生していない場合は、nilになります。	番号
exit_signal	プロセスが終了したときのシグナル番号。発生していない場合は、nilになります。	番号
*（入力パラメータテーブルのstdout*フィールドの値*）*	終了したプロセスの*stdout*ストリームから読み込まれたデータ。フィールドは、入力パラメータのテーブルでstdoutフィールドが指定されている場合にのみ使用可能です。	文字列
*（入力パラメータテーブルのstderr*フィールドの値*）*	終了したプロセスの*stderr*ストリームから読み込まれたデータ。フィールドは、入力パラメータのテーブルでstderrフィールドが指定されている場合にのみ使用可能です。	文字列

3. 例

- catユーティリティを引数なしで実行し、'some data'テキストをその入力ストリームに渡し、コマンド出力の結果を、返されるテーブルのstdout_fieldフィールドに渡す。

```
local sp = require 'drweb.subprocess'

local cat_result = sp.run({
   '/bin/cat',
   stdin = 'some data',
   stdout = 'stdout_field',
})
```

`cat_result`変数に格納される結果のテーブルには、以下のフィールドが含まれます。

フィールド	Valueexit_status
exit_status	0
exit_signal	nil
stdout_field	'some data'

- `-c`と`env | grep TESTVAR 1>&2`のパラメータを指定して`sh`コマンド(コマンドインタプリタ)を実行し、`VALUE`値を持つ`TESTVAR`変数を環境に追加し、コマンド*stderr*の出力結果を、返されるテーブルの`stderr_field`フィールドに渡す。

```
local sp = require 'drweb.subprocess'

local env_result = sp.run({
   '/bin/sh', '-c', 'env | grep TESTVAR 1>&2',
   env = { TESTVAR = 'VALUE' },
   stderr = 'stderr_field'
})
```

`env_result`変数に格納される結果のテーブルには、以下のフィールドが含まれます。

フィールド	Valueexit_status
exit_status	0
exit_signal	nil
stderr_field	'TESTVAR=VALUE\n'

- `pwd`コマンドを実行し、作業ディレクトリをシステムのルートディレクトリに設定し、コマンド*stdout*の出力結果を、返されるテーブルの`stdout_field`フィールドに渡す。

```
local sp = require 'drweb.subprocess'

local pwd_result = sp.run{
   '/bin/pwd',
   workdir = '/',
   stdout = 'stdout_field'
}
```

`pwd_result`変数に格納される結果のテーブルには、以下のフィールドが含まれます。

フィールド	Valueexit_status
exit_status	0
exit_signal	nil
stdout_field	'/\n'

- killコマンドをbashで実行し、*SIGKILL*信号をインタプリタに渡す。

```
local sp = require 'drweb.subprocess'

local kill_result = sp.run{'/bin/bash', '-c', 'kill -9 $$'}
```

kill_result変数に格納される結果のテーブルには、以下のフィールドが含まれます。

フィールド	**Valueexit_status**
exit_status	nil
exit_signal	9

プロセスを非同期的に実行するには、async関数呼び出し内でrun関数を実行します（上記を参照）。

drweb.configモジュールの内容

1. 機能

このモジュールには関数はありません。

2. 利用可能なテーブル

- このモジュールでは、次のフィールドでMailDConfigテーブルを提供します。

フィールド	説明	データタイプ
version	Dr.Web MailDのバージョン	文字列
無効になったメタメソッド: *なし*		

MailDConfigテーブルは、モジュールによってmaildフィールドとして提供されます。

3. 例

- Dr.Web MailDコンポーネントの現在のバージョンをログに出力する。

```
local dw  = require 'drweb'
local cfg = require 'drweb.config'

-- "Main" function
function milter_hook(ctx)

 -- Output of a string at the NOTICE level to the Dr.Web for UNIX Mail
Servers log
 dw.notice(cfg.maild.version)

end
```

drweb.storeモジュールの内容

1. 機能

このモジュールは次の機能を提供します。

- `exists`(*<name>*、*<key>*)は、選択したリポジトリに指定したキーを持つエントリがあるかどうかをチェックします。2つの引数、すなわちリポジトリの名前である*<name>*(文字列)と、エントリのキーである*<key>*(文字列)を取ります。エントリがあれば`true`を返し、なければ`false`を返します。
- `get`(*<name>*、*<key>*)は、選択したストレージから、指定したキーを持つエントリの値を取得します。2つの引数、すなわちリポジトリの名前である*<name>*(文字列)と、エントリのキーである*<key>*(文字列)を取ります。`value`パラメータと`ctime`パラメータのペアを返します。エントリがない場合は`nil`を返します。
  `value parameter`(文字列)は、指定されたキーを持つエントリの値で、`ctime`(整数)はレコードの変更タイムスタンプです。
- `put`(*<name>*、*<key>*、*<value>*)は、指定したキーを持つエントリを選択したストレージに追加します。3つの引数、すなわちリポジトリの名前である*<name>*(文字列)、エントリのキーである*<key>*(文字列)、エントリの値である*<value>*(文字列)を取ります。
- `remove`(*<name>*、*<key>*)は、指定したキーを持つエントリを選択したストレージから削除します。2つの引数、すなわちリポジトリの名前である*<name>*(文字列)と、エントリのキーである*<key>*(文字列)を取ります。
- `count`(*<name>*)は、選択したストレージ内のレコード数を返します。1つの引数、すなわちリポジトリの名前である*<name>*(文字列)を取ります。整数を返します。
- `drop`(*<name>*、*<ctime>*)は、指定したタイムスタンプ以前に変更されたすべてのエントリを、選択されたストレージから削除します。2つの引数、すなわちリポジトリの名前である*<name>*(文字列)と、エントリ変更のタイムスタンプである`ctime`(整数)を取ります。

2. 例

- アンチスパムスキャン: local store = require "drweb.store"のホワイトリストを作成する。

```
local store = require "drweb.store"

local antispam_whitelist = "antispam_whitelist"
local intra_domain_mask = ".*@test%.test$"

-- "Main" function
function milter_hook(ctx)

 -- Add all the outgoing mails recipients
 -- to the anti-spam white list
 if ctx.from:match(intra_domain_mask) then
   for _, to in ipairs(ctx.to) do
    store.put(antispam_whitelist, to, "")
   end
   return {action="accept"}
 end

 if not store.exists(antispam_whitelist, ctx.from) then
   if ctx.message.spam.score > 300 then
    return {action="reject"}
   end
 end

 return {action="accept"}

end
```

- ホワイトリストに一時的に追加する。

```
local store = require "drweb.store"

local antispam_whitelist = "antispam_whitelist"
local antispam_whitelist_timeout = 604800 -- 1 week
local intra_domain_mask = ".*@test%.test$"

-- "Main" function
function milter_hook(ctx)

 -- Add all the outgoing mails recipients
 -- to the anti-spam white list
 if ctx.from:match(intra_domain_mask) then
   for _, to in ipairs(ctx.to) do
    store.put(antispam_whitelist, to, "")
   end
   return {action="accept"}
 end

 local _, ctime = store.get(antispam_whitelist, ctx.from)
 -- Is on the list and was added within a week
 local in_whitelist = ctime and os.time() + ctime <
antispam_whitelist_timeout

 if not in_whitelist and ctime then
   store.remove(antispam_whitelist, ctx.from)
 end

 -- You can also update a non-expired entry:
 -- if in_whitelist then
 --  store.put(antispam_whitelist, ctx.from, "")
 -- end

 if not in_whitelist then
   if ctx.message.spam.score > 300 then
    return {action="reject"}
   end
 end

 return {action="accept"}

end
```

Dr.Web Anti-Spam

Dr.Web Anti-Spamは、メールメッセージをスキャンしてスパムを検出するためのコンポーネントです。このコンポーネントは、メールスキャンコンポーネントDr.Web MailDによって使用されます。パッケージによっては、Dr.Web for UNIX Mail ServersにDr.Web Anti-Spamがない場合があります（この場合、Dr.Web MailDはスパムスキャンを実行しません）。

ARM64およびE2Kのアーキテクチャでは、このコンポーネントはサポートされていません。

動作原理

Dr.Web MailD（またはその他の外部アプリケーション）から受信したメッセージのスパムの兆候についての解析は、アンチスパムライブラリとDr.Web Anti-Spamコンポーネントを使用して実行されます。メッセージの解析は、スパムに関する情報の外部ソースへの要求なしに、スタンドアロンモードで実行されます。このソリューションではメッセージのスパム分類ルールのデータベースが動的に更新されるため、メッセージ処理の高速化とメッセージ解析品質の継続的な向上も実現します（更新はDr.Web Updaterで自動的に行われます）。

ユーザーは、メールメッセージのアンチスパムスキャン用にDr.Web Anti-Spamを使用して独自のコンポーネント（外部アプリケーション）を作成できます。そのために、Dr.Web Anti-SpamにはGoogle Protobufをベースにした特別なAPIが含まれています。Dr.Web Anti-Spam APIのガイドとDr.Web Anti-Spamを使用したクライアントアプリケーションの例を入手するには、Doctor Webパートナーケア部門（https://partners.drweb.com/）までお問い合わせください。

Dr.Web for UNIX Mail Serversには、ARM64、E2K、IBM POWER（ppc64el）アーキテクチャ向けのDr.Web Anti-Spamは含まれていません。

Dr.Web Anti-Spamによって誤って検出されたメールメッセージがある場合は、分析のため、また、スパムフィルタの品質向上のためにそれらを専用のアドレスに転送していただけますようお願いいたします。これを行うには、各メッセージを別々の`.eml`ファイルに保存します。次に、保存したファイルをメールメッセージに添付して、該当するアドレスに転送してください。

- *誤ってスパムと判定されたメールメッセージは*nonspam@drweb.comに送信してください。
- *スパムとして検出されなかったメールメッセージは*spam@drweb.comに送信してください。

コマンドライン引数

OSのコマンドラインからDr.Web Anti-Spamを起動するには、次のコマンドを使用します。

```
$ <opt_dir>/bin/drweb-ase [<arguments>]
```

Dr.Web Anti-Spamでは、次のオプションを使用できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-ase --help
```

このコマンドは、Dr.Web Anti-Spamに関する簡単なヘルプ情報を出力します。

スタートアップノート

スタンドアロンモードでコマンドラインからコンポーネントを直接起動するオプションは提供されていません。メールのスパムスキャン中にDr.Web Anti-Spamコンポーネントによって自動的に実行されます。さらに、コンポーネント構成で`FixedSocket`パラメータ値が定義されている場合、1つのコンポーネントコピーがDr.Web ConfigD設定デーモンによって常に実行され、このUNIXソケットを介してユーザーが使用できるようになります。コンポーネントパラメータとメールオブジェクトチェックを管理するには、コマンドラインからDr.Web for UNIX Mail Serversを管理するためのDr.Web Ctlユーティリティを使用します（コマンド`drweb-ctl`によって実行されます）。

Dr.Web Anti-Spamコンポーネント（Dr.Web MailDコンポーネント呼び出し）による任意のメールメッセージのスパムスキャンを実行するには、ツールDr.Web Ctlのコマンド`checkmail`を使用します。これを行うには、スキャンしたメールメッセージをドライブに保存して（`.eml`形式など）、次のコマンドを使用します。

```
$ drweb-ctl checkmail <path to file .eml>
```

このコンポーネントに関するマニュアルをコマンドラインから要求するには、`man 1 drweb-ase`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの[Antispam]セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-ase • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-ase • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-ase
RunAsUser *{UID / user name}*	このパラメータは、コンポーネントを実行するユーザー名を決定します。ユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「name:」というプレフィックスを付けて指定します。たとえば、RunAsUser = name:123456です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：drweb
FixedSocket *{path to file}*	固定コンポーネントコピーのソケットファイルへのパス。 このパラメータが指定されている場合、Dr.Web ConfigD設定デーモンは、このソケットを介してクライアントが使用可能な実行中のコンポーネントのコピーが常に存在することを確認します。 デフォルト値：*（未設定）*
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 FixedSocket値が設定されている場合、この設定は無視されます（指定した時間を経過しても、コンポーネントはその動作を終了しません）。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能しま

パラメータ	説明
	す。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s
FullCheck *{Boolean}*	メッセージのフルスキャンを実行してスパムの兆候がないか調べます。Noの場合、スパムスコアの数値がFastCheckStopThresholdパラメータで指定された値を超えるとすぐに、スキャンが停止します。 デフォルト値：Yes
FastCheckStopThreshold *{integer}*	FullCheckパラメータの値がNoに設定されている場合、スパムスコアが上限に達すると、メッセージのスキャンが停止します。 デフォルト値：300
AllowCyrillicText *{Boolean }*	キリル文字のテキストを含むメールを許可します。Noの場合、そのようなメールのスパムスコアが増加します。 デフォルト値：Yes
AllowCjkText *{Boolean}*	アジア言語（中国語、日本語、韓国語）のテキストを含むメールを許可します。Noの場合、そのようなメールのスパムスコアが増加します。 デフォルト値：Yes
CheckCommercialEmails *{Boolean}*	広告メール（プロモーションメール、プロモーションやセールスの通知など）をチェック対象から除外します。Noの場合、そのようなメールはスパムと見なされます。 デフォルト値：No
CheckSuspiciousEmails *{Boolean}*	疑わしいメール（懸賞金の提供を申し出るメールなど）をチェック対象から除外します。Noの場合、そのようなメールはスパムと見なされます。 デフォルト値：No
CheckCommunityEmails *{Boolean}*	ソーシャルメディアメールをチェック対象から除外します。Noの場合、そのようなメールはスパムと見なされます。 デフォルト値：No
CheckTransactionalEmails *{Boolean}*	トランザクション通知（サービスや商品などの購入、登録）をチェック対象から除外します。Noの場合、そのようなメールはスパムと見なされます。 デフォルト値：No
DetectSpamType *{Boolean}*	スパムタイプ（不正行為、詐欺）のチェックを無効にします。Noの場合、そのようなメールはスパムと見なされます。 デフォルト値：No

Dr.Web Mail Quarantine

Dr.Web Mail Quarantineメッセージキューマネージャーはメッセージスキャン中にメールメッセージとそのメタデータをハードディスクに保存します。

SMTPまたはBCCモードで機能しているとき、Dr.Web Mail QuarantineはDr.Web MailDコンポーネントによって使用されます。Dr.Web Mail Quarantineが機能することで、Dr.Web MailDのエラーやMTAの接続損失が発生した場合でも、メッセージキューの保存や、メッセージの処理および再送信が中断なく行われます。

動作原理

Dr.Web Mail Quarantineコンポーネントは主に、次の2つの目的を果たします。

- Dr.Web MailDのメッセージチェック時に、メッセージキューとメタデータをハードディスクに保存する。メールメッセージはファイルとして保存され、そのメタデータはSQLiteリレーショナルデータベースに保存されます。メッセージの保存はSMTPおよびBCCモードで必要です。
- メッセージ処理中にエラー（メッセージを再送信できなかったなど）が発生した場合に、一定のタイムアウト後に、Dr.Web MailDコンポーネントにメッセージをリダイレクトする。このコンポーネントによって、処理されたメッセージが確実にMTAに配信されます。

ユーザーがDr.Web Mail Quarantineコンポーネントを起動することはできません。他のコンポーネントからリクエストを受信したときに、Dr.Web ConfigD設定デーモンによって自動的に起動されます。

コマンドライン引数

コマンドラインからDr.Web Mail Quarantineを起動するには、次のコマンドを入力します。

```
$ drweb-ctl mailquarantine [<parameters>]
```

Dr.Web Mail Quarantineでは次のパラメータを使用できます。

パラメータ	説明
`--help`	機能：既存のコマンドラインパラメータに関するヘルプをコンソールまたはターミナルエミュレーターに表示し、その後、対応するプロセスをシャットダウンします。 短縮コマンド：`-h` 引数：なし
`--version`	機能：コンポーネントのバージョン情報をコンソールまたはターミナルエミュレーターに表示し、その後、対応するプロセスをシャットダウンします。 短縮コマンド：`-v` 引数：なし

例：

```
$ drweb-ctl mailquarantine --help
```

このコマンドは、Dr.Web Mail Quarantineに関するヘルプを表示します。

スタートアップノート

スタンドアロンモードではコマンドラインからコンポーネントを起動できません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。環境設定を管理するには、コマンドラインからDr.Web for UNIX Mail Serversを管理できるように用意されているDr.Web Ctlツールを使用します（`drweb-ctl`コマンドを使用して起動します）。

このコンポーネントに関するヘルプをコマンドラインから表示するには、`man 1 drweb-mail-quarantine`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[MailQuarantine]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータ値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。 デフォルト値：`Notice`
`Log` *{log type}*	コンポーネントのロギング方式。 デフォルト値：`Auto`
`ExePath` *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*`/bin/drweb-mail-quarantine` • GNU/Linuxの場合：`/opt/drweb.com/bin/drweb-mail-quarantine` • FreeBSDの場合：`/usr/local/libexec/drweb.com/bin/drweb-mail-quarantine`
`RunAsUser` *{UID / user name}*	このパラメータは、コンポーネントを実行するユーザー名を決定します。ユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（UIDに似ている場合）は、「`name:`」というプレフィックスを付けて指定します。たとえば、`RunAsUser = name:123456`です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：`drweb`
`IdleTimeLimit` *{time interval}*	コンポーネントがアイドル状態を維持できる最大時間。指定された値を超えると、コンポーネントはシャットダウンします。

パラメータ	説明
	指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは無期限に機能します。コンポーネントがアイドル状態になると、SIGTERM信号は送信されません。 デフォルト値：30s
SpoolDir *{path to directory}*	メールメッセージとメタデータの保存に使用されるローカルファイルシステムのディレクトリ。 デフォルト値：*<var_dir>*/drweb.com/lib/mail-quarantine • GNU/Linuxの場合：/var/opt/drweb.com/lib/mail-quarantine • FreeBSDの場合：/var/drweb.com/lib/drweb-mail-quarantine

SpIDer Gate

このコンポーネントはGNU/Linux OSのディストリビューションにのみ含まれています。

ネットワークトラフィックとURLを監視するコンポーネントSpIDer Gateは、データ(ネットワークからローカルコンピューター、ローカルホストからネットワークにダウンロードされたデータ)の脅威を検査し、望ましくないカテゴリーに含まれるWebリソースや、管理者によって定義されたブラックリストに含まれるネットワークホストとの接続を防止します。

コンポーネント設定では、スキャン対象のプロトコルの種類を指定できます。このコンポーネントには、検査済みの接続を介してデータを送信するために使用されるプロトコルタイプのアナライザが含まれています。プロトコルがメールプロトコルであると判定された場合、脅威の解析と検索には、スキャン対象としてDr.Web MailDメールメッセージコンポーネントが使用されます。

このコンポーネントは、URLがいずれかのカテゴリー(HTTP/HTTPSプロトコルを利用する接続のスキャンに使用される)に属しているかどうかを確認するために、Doctor Webの更新サーバーから定期的に更新されるWebリソースカテゴリーのデータベースを使用するだけでなく、Dr.Web Cloudサービスの参照も行います。Doctor Webは、以下のWebリソースカテゴリーを追跡します。

- *InfectionSource* - 悪意のあるソフトウェアを含むWebサイト(「感染源」)。
- *NotRecommended* - アクセスすることが推奨されない不正なWebサイト(「ソーシャルエンジニアリング」を使用しているもの)。
- *AdultContent* - ポルノまたはエロティックなコンテンツ、出会い系サイトなどを含むWebサイト。
- *Violence* - 暴力行為を助長するWebサイトや、さまざまな死亡事故などに関するコンテンツを含むWebサイト。
- *Weapons* - 武器および爆発物に関するWebサイトや、それらの製造に関する情報を提供しているWebサイト。
- *Gambling* - 勝負事、カジノ、オークションなどのオンラインゲームへのアクセスを提供するWebサイト(賭けサイトなどを含む)。
- *Drugs* - 麻薬の使用、製造または流通などを促進するWebサイト。
- *ObsceneLanguage* - (タイトルや記事などに)卑猥な表現を含むWebサイト。
- *Chats* - テキストメッセージのリアルタイム送信を提供するWebサイト。
- *Terrorism* - 攻撃的なプロパガンダ、またはテロ攻撃などに関する内容を含むWebサイト。
- *FreeEmail* - メールの無料登録を提供するWebサイト。
- *SocialNetworks* - さまざまなソーシャルネットワークサービス: 一般、仕事、企業、興味、テーマ別出会い系サイト。
- *DueToCopyrightNotice* - 一部の著作物(映画、音楽など)の著作権者によって定義されるWebサイト。
- *OnlineGames* - インターネットへの常時接続を使用してゲームへのアクセスを提供するWebサイト。
- *Anonymizers* - ユーザーが個人情報を隠し、ブロックされたWebリソースにアクセスすることを可能にするWebサイト。
- *CryptocurrencyMiningPool* - 仮想通貨マイニングのための一般的なサービスへのアクセスを提供するWebサイト。

- *Jobs* - 求人検索Webサイト。

システム管理者は、ホストが属するカテゴリーに基づいて、望ましくないホストへのアクセスを設定できます。またユーザーは、特定のホストへのアクセスをブロックするために独自のブラックリストを設定したり、アクセスを許可するためにホワイトリストを設定したりすることもできます。ホワイトリストのホストへのアクセスは、それらが望ましくないカテゴリーに属する場合でも許可されます。ローカルのブラックリストとWebリソースのカテゴリーのデータベースにURLに関する情報がない場合、コンポーネントはDr.Web Cloudサービスを参照して、他のDr.Web製品から受信するこうしたURLに悪意があるかどうかの情報をリアルタイムで検査できます。

同一のWebサイトは、複数のカテゴリーに同時に所属することができます。そのようなWebサイトへのアクセスは、不要なカテゴリーのいずれに属する場合もブロックされます。

Webサイトがホワイトリストに含まれる場合でも、データ（Webサイトから送信およびダウンロードされたもの）に脅威が含まれているかどうかがスキャンされます。

HTTPプロトコルを介して転送されるファイルのスキャンの負荷が高まると、Dr.Web Network Checkerコンポーネントが利用できるファイル記述子の数が減少するため、スキャンに問題が生じる場合があります。この場合、Dr.Web for UNIX Mail Serversに利用できるファイル記述子の制限数を増やす必要があります。

動作原理

SpIDer Gateコンポーネントは、ユーザーアプリケーションによって確立されたネットワーク接続を監視します。このコンポーネントは、クライアントアプリケーションが接続しようとしているサーバーが、設定で望ましくないと指定されているWebリソースカテゴリーのいずれかに属するかどうかを検査します。さらにこのコンポーネントは、Dr.Web Cloudを参照してURLを検査できます。URLが望ましくないカテゴリー（Dr.Web Cloudのリクエストによって返されたものを含む）に属する場合や、システム管理者によって定義されたブラックリストに属する場合、接続は中断され、アクセスが許可されていないというメッセージが含まれるHTMLページが表示されます（HTTP/HTTPS接続の場合）。HTMLページは、コンポーネントに付属するテンプレートに従ってSpIDer Gateによって生成されます。このページには、リクエストされたリソースにアクセスできないという通知と、ブロックに関する詳細が表示されます。SpIDer Gateがブロックする必要のある脅威を見つけた場合にも同様のページが表示され、クライアントに返されます。接続にHTTP（S）とは異なるプロトコルが使用されている場合、コンポーネントはこのサーバーとの接続を確立するための許可のみをスキャンします。メールプロトコル（SMTP、POP3、またはIMAP）であると判断された場合、メールメッセージのスキャン用コンポーネントDr.Web MailDがデータの解析と脅威の検索に使用されます。このコンポーネントは独自にメールメッセージを分類し、本文に含まれるファイルやURLを抽出します。そのため、このコンポーネントはコンポーネントSpIDer Gateと共通のブロックパラメータを使用します。

補助コンポーネントDr.Web Firewall for Linuxは、クライアントアプリケーションによって確立されたリモートサーバーとの接続をリダイレクトします。このコンポーネントはGNU/Linuxシステムコンポーネントの NetFilterルールの動的管理を実行します。

Dr.Web Updaterコンポーネントは、Doctor Web更新サーバーからWebリソースカテゴリーのデータベースを定期的かつ自動的に更新するために使用されます。同じコンポーネントは、Dr.Web Scanning Engineスキャンエンジン用のウイルスデータベースの更新にも使用されます。Dr.Web CloudDコンポーネントはDr.Web Cloudサービスの参照に使用されます（クラウドサービスの使用は付録共通設定で設定され、必要に応じて無効にできます）。転送されたデータを検査するために、SpIDer GateはDr.Web Network Checkerコンポーネントを使用します。後者では、Dr.Web Scanning Engineスキャンエンジンを介してスキャンが開始されます。

コマンドライン引数

SpIDer Gateを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-gated [<parameters>]
```

SpIDer Gateは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-gated --help
```

このコマンドはSpIDer Gateに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-gated`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[GateD]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel`	コンポーネントのロギングレベル。

パラメータ	説明
{logging level}	パラメータの値が指定されていない場合は、[Root] セクションの DefaultLogLevel パラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-gated • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-gated • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-gated
RunAsUser *{UID / user name}*	その権限によりコンポーネントを実行するユーザーの名前。このユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「name:」というプレフィックスを付けて指定します。たとえば、RunAsUser = name:123456です。 ユーザー名が指定されていない場合、コンポーネント動作は、開始後に発生するエラーによって終了します。 デフォルト値：drweb
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s
TemplatesDir *{path to directory}*	Webリソースをブロックしたときに送信されるHTML通知のテンプレートを含むディレクトリへのパス。 デフォルト値：*<var_dir>*/templates/gated • GNU/Linuxの場合：/var/opt/drweb.com/templates/gated • FreeBSDの場合：/var/drweb.com/templates/gated
CaPath *{path}*	信頼できるルート証明書のシステムリストを含むディレクトリまたはファイルへのパス。 デフォルト値：*信頼できる証明書のリストへのパス*。パスは、お使いのGNU/Linuxディストリビューションに依存します。 • Astra Linux、Debian、Linux Mint、SUSE Linux、Ubuntuの場合、通常はパス/etc/ssl/certs/です。 • CentOSとFedoraの場合はパス/etc/pki/tls/certs/ca-bundle.crtです。 • 他のディストリビューションでは、コマンドopenssl version -dの実行結果によってパスを定義できます。

パラメータ	説明
	• コマンドが使用できない場合、またはOSディストリビューションを特定できない場合は、値 `/etc/ssl/certs/` が使用されます。

接続スキャンの設定を変更しても、変更を加える前にアプリケーションによってすでに確立されている接続のスキャンには影響しません。

補助コンポーネントDr.Web Firewall for Linuxの設定で、トラフィックモニタリングのより具体的なパラメータを指定します。

Dr.Web Firewall for Linux

このコンポーネントはGNU/Linux OSのディストリビューションにのみ含まれています。

コンポーネントを正しく動作させるため、以下のオプションを指定してOSカーネルが構築されている必要があります。

- *CONFIG_NETLINK_DIAG, CONFIG_INET_TCP_DIAG;*
- *CONFIG_NF_CONNTRACK_IPV4, CONFIG_NF_CONNTRACK_IPV6, CONFIG_NF_CONNTRACK_EVENTS;*
- *CONFIG_NETFILTER_NETLINK_QUEUE, CONFIG_NETFILTER_NETLINK_QUEUE_CT, CONFIG_NETFILTER_XT_MARK.*

必要なオプションの組み合わせは、使用するGNU/Linuxのディストリビューションキットによって異なります。

Dr.Web Firewall for Linuxは補助コンポーネントです。SpIDer Gateに対して接続マネージャーの役割を果たします。Dr.Web Firewall for Linuxが機能することで、ホスト接続がSpIDer Gateを通過し、接続トラフィックが監視されます。

動作原理

このセクションの内容:

- 概要
- 接続監視のメカニズム
- 接続監視の順序

概要

SpIDer GateはDr.Web Firewall for Linuxコンポーネントの働きにより正常に動作します。このコンポーネントは、NetFilter(GNU/Linux OSコンポーネント)用に調整されたルーティングルールを解析し、確立された接続が、クライアントアプリケーションとリモートサーバー間の中間の機能(プロキシ)を実行するSpIDer Gateにリダイレクトされるように修正します。

Dr.Web Firewall for Linuxは、トランジット接続だけでなく、送信と受信のリダイレクトのルールを別々に管理できます。バイパスまたはリダイレクトのルールを正確に設定するには、コンポーネントではLuaで書かれたスクリプトだけでなく設定に組み込まれたルールを使うことができます。

接続監視のメカニズム

接続を監視するために、Dr.Web Firewall for Linuxはルーティングポリシーのデータベース(`man ip`: `ip route`、`ip rule`参照)とNetFilterシステムコンポーネントの*nf_conntrack*インターフェースで指定されたルーティングテーブルを使用します。監視された接続と転送されたパケットには、Dr.Web Firewall for LinuxがNetFilterのチェーンのさまざまな段階で接続をリダイレクトし、転送されたパケットを正しく処理できるようにするビットマークが付けられます(詳細は`man iptables`を参照)。

iptablesルールのアクション

Dr.Web Firewall for Linuxは、`iptables`ルールで次のアクションを使用します。

- *MARK*: このアクションにより、Dr.Web Firewall for Linuxは指定された数字マークをパケットに設定できます。
- *CONNMARK*: このアクションにより、Dr.Web Firewall for Linuxは指定された数字マークを接続に設定できます。
- *TPROXY*: このアクションにより、Dr.Web Firewall for Linuxは、パケットのコンテンツを変更することなく、*PREROUTING* NetFilterチェーンから指定されたネットワークソケット（*<IP address>*: *<port>*）にパケットをリダイレクトできます。このアクションを使用すると、Dr.Web Firewall for Linuxは接続の最初の宛先アドレスを特定できます。
- *NFQUEUE*このアクションにより、エンジンのネットワークスタックから、スキャンのためにカーネル空間外で動作するプロセスにパケットを送信できます。したがって、Dr.Web Firewall for Linuxは、特殊な*Netlink*ソケットを介して、指定した番号のキュー*NFQUEUE*に接続し、処理で判定するために必要なパケットを入手します（Dr.Web Firewall for Linuxは、NetFilterに、*DROP*: *ドロップ*、*ACCEPT*: *許可*、*REPEAT*: *繰り返し*のいずれかの判定を伝える必要があります）。

パケットと接続のマーク

パケットをマークするため、Dr.Web Firewall for Linuxは、パケットおよび接続マークで使用可能な32ビットのうち次の3つを使用します。

- `LDM`ビット（*Local Delivery Mark*）。マークにこのビットがあるパケットは、使用しているルーティングルールに基づいてローカルホストに送信されます。
- `CPM`ビット（*Client Packets Mark*）。クライアント（接続開始側）とプロキシ（Dr.Web Firewall for Linuxなど）の間の接続を示します。
- `SPM`ビット（*Server Packets Mark*）。プロキシ（Dr.Web Firewall for Linuxなど）とサーバー（接続受信側）の間の接続を示します。

`LDM`、`CPM`、`SPM`ビットは、ルーティング接続を実行する他のアプリケーションがパケットをマークするために使用していない任意の*さまざまな*ビットにすることができます。デフォルトでは、Dr.Web Firewall for Linuxは適切な（他のアプリケーションでは使用されていない）ビットを自動的に選択します。

ルートとルーティングポリシー（ip rule、ip route）

Dr.Web Firewall for Linuxを（あらゆる接続スキャンモードで）正しく動作させるには、100番のルーティングテーブルを使用する`ip rule`ルーティングポリシーをシステムに設定する必要があります。

```
from all fwmark <LDM>/<LDM> lookup 100
```

次のルートをテーブルに追加する必要があります。

```
local default dev lo scope host
```

このルーティングポリシーは、マークに`LDM`ビットが入っているパケットが常にローカルホストに送信されることを保証します。

それ以降、XXXビットの *<XXX>*文字列は、2^N（*N*はパケットマーク内のXXXビットの序数）に等しい*16進*値になります。たとえば、最小（ゼロ）ビットがLDMビットとして選択されている場合は、*<LDM>* = 2^0 = *0x1*になります。

NetFilter（iptables）のルール

Dr.Web Firewall for Linuxを（あらゆる接続スキャンモードで）正しく動作させるには、次の6つのルール（iptables-save出力コマンド形式で表示）がNetFilterシステムコンポーネントの該当するチェーンのnatテーブルとmangleテーブルに存在している必要があります。

```
*nat

-A POSTROUTING -o lo -m comment --comment drweb-firewall -m mark --mark
<LDM>/<LDM> -j ACCEPT

*mangle

-A PREROUTING -m comment --comment drweb-firewall -m mark --mark
0x0/<CPM+SPM> -m connmark --mark <SPM>/<CPM+SPM> -j MARK --set-xmark
<LDM>/<LDM>
-A PREROUTING -p tcp -m comment --comment drweb-firewall -m mark ! --mark
<CPM+SPM>/<CPM+SPM> -m connmark --mark <CPM>/<CPM+SPM> -j TPROXY --on-port
<port> --on-ip <IP-address> --tproxy-mark <LDM>/<LDM>
-A OUTPUT -m comment --comment drweb-firewall -m mark --mark
<CPM>/<CPM+SPM> -j CONNMARK --set-xmark <CPM>/0xffffffff
-A OUTPUT -m comment --comment drweb-firewall -m mark --mark
<SPM>/<CPM+SPM> -j CONNMARK --set-xmark <SPM>/0xffffffff
-A OUTPUT -m comment --comment drweb-firewall -m mark --mark 0x0/<CPM+SPM>
-m connmark ! --mark 0x0/<CPM+SPM> -j MARK --set-xmark <LDM>/<LDM>
```

以下の説明では、0～5の番号がこれらのルールに割り当てられています（文書に記載されている順序で）。式 *<X+Y>*は、それぞれの数*X*と*Y*のビット数「OR」（合計）を意味します。

ルール番号2のパラメータ*<IP address>*と*<port>*は、Dr.Web Firewall for Linuxが監視された接続を管理するネットワークソケットを示します。

さらに、Dr.Web Firewall for Linux設定で監視接続モード（送信、受信、トランジット）を有効にする場合は、次の追加ルールが該当するチェーンのmangleテーブル（*OUTPUT*、*INPUT*、*FORWARD*）に存在する必要があります。

- *送信*（*OUTPUT*）接続を監視するには：

```
-A OUTPUT -p tcp -m comment --comment drweb-firewall -m tcp --tcp-flags
SYN,ACK SYN -m mark --mark 0x0/<CPM+SPM> -j NFQUEUE --queue-num <ONum> --
queue-bypass
```

- *受信*（*INPUT*）接続を監視するには：

```
-A INPUT -p tcp -m comment --comment drweb-firewall -m tcp --tcp-flags
SYN,ACK SYN -m mark --mark 0x0/<CPM+SPM> -j NFQUEUE --queue-num <INum> --
queue-bypass
```

- *トランジット(FORWARD)*接続を監視するには:

```
-A FORWARD -p tcp -m comment --comment drweb-firewall -m tcp --tcp-flags
SYN,ACK SYN -m mark --mark 0x0/<CPM+SPM> -j NFQUEUE --queue-num <FNum> --
queue-bypass
```

以下の説明では、6、7、8の番号がこれらのルールに割り当てられています(文書に記載されている順序で)。

ここで、*<ONum>*、*<INum>*、*<FNum>*は、Dr.Web Firewall for Linuxが対応する接続のインストールを示すパケット(SYNフラグが設定されているがACKフラグは含まれていないパケット)を待機している*NFQUEUE*内のキューの数です。

接続監視の順序

ルール6、7、8のいずれかに従って、対応する方向の新しいネットワーク接続を示すパケットは、CPMとSPMのどちらのビットでもマークされていない場合、NetFilterによって該当する*NFQUEUE*キューに入れられます。ここでパケットは、*nf_conntrack*インターフェースを介してDr.Web Firewall for Linuxによって読み取られます。ルール3と4は、接続を監視済みとしてマークします。つまり、接続マークに接続方向が設定されていることを示すビットがあります。このビット番号は、パケットマークのビット番号と一致します。その結果、ルール1、2、5に従って、この接続を介して送信されたパケットはDr.Web Firewall for Linuxによって配信されます。ルール0がnatテーブルの*POSTROUTING*チェーンの最上部に追加されるため、NATが設定されている場合、マークされたパケットのアドレスは送信されません(Dr.Web Firewall for Linuxの監視および接続処理ロジックに干渉するため)。

パケットがいずれかの*NFQUEUE*キューに現れると、Dr.Web Firewall for Linuxは、NetFilterで誤ったルールが設定されている場合に備えて、パケットの基本的な処理を実行します。次に、Dr.Web Firewall for Linuxは、ルール4に従い、それ自体の名前と、PSCとマークされたソケットを使用してサーバーへの接続を試みます。ローカル配信のルール5は適用されません。パケットはSPMでマークされており、このルールは*<CPM + SPM>*でマークされたパケットにのみ適用できるためです。

- *サーバーへの接続が失敗した場合*、Dr.Web Firewall for Linuxは、RSTビットを含むクライアントパケットを生成し、*<IP address>*:*<port>*のペアをリクエストされたサーバーのネットワークソケットのアドレスに置き換えます。DROP判定も*NFQUEUE*に送信されます。RSTビットを含むパケットの送信に使用されるソケットは*<CPM+SPM>*としてマークされているため、上記のルールはどれも適用されず、パケットは通常のルーティングルールに従ってクライアントに配信されます。
- *リモートサーバーへの接続が成功した場合*、Dr.Web Firewall for Linuxは、監視したSYNパケットをコピーし、*<LDM+CPM>*とマークされたソケットから再送信して、パケットをローカルネットワークソケットにリダイレクトします。LDMビットが設定されているため、指定したルーティングルールに従って出力インターフェースを選択すると、パケットは*loopback*インターフェースに追加されます。その後、NetFilter *PREROUTING*チェーンに追加され、そこでルール2が適用されます。したがって、パケットは変更されることなくネットワークソケットDr.Web Firewall for Linuxにリダイレクトされます。この機能により、Dr.Web Firewall for Linuxは接続アドレスの4つの要素すべて(パケット送信者のIPアドレスとポート、パケット受信者のIPアドレスとポート)を保存できます。

Dr.Web Firewall for Linuxがルール2に従って監視した接続を受信するネットワークソケットの場合、IP_TRANSPARENTオプションと*<LDM+CPM>*マークが設定され、このソケットからDr.Web Firewall for

Linuxによって送信されたパケットが*NFQUEUE*キューに分類されないようにします。クライアントが接続すると、保存されている4要素のアドレス(パケット送信者のIPアドレスとポート、パケット受信者のIPアドレスとポート)を使用して、ペアのソケットが検索されます。クライアントとサーバーの接続が確立されると、Luaの手順とDr.Web Firewall for Linuxの設定で指定されたスキャンルールに従ってスキャンされます。スキャンが成功し、接続が安定している場合は、クライアント側とサーバー側を接続する関連ソケットのペアが転送データの解析のためにSpIDer Gateコンポーネントに転送されます。それに続くクライアントとサーバー間の対話は、メディエーターのSpIDer Gateを介して確立されます。Dr.Web Firewall for Linuxは、クライアント側とサーバー側に関連付けられたソケットペアに加えて、確立された接続をスキャンするためのパラメータとルールをSpIDer Gateに送信します。

Dr.Web Firewall for Linux動作の概略図を以下に示します。

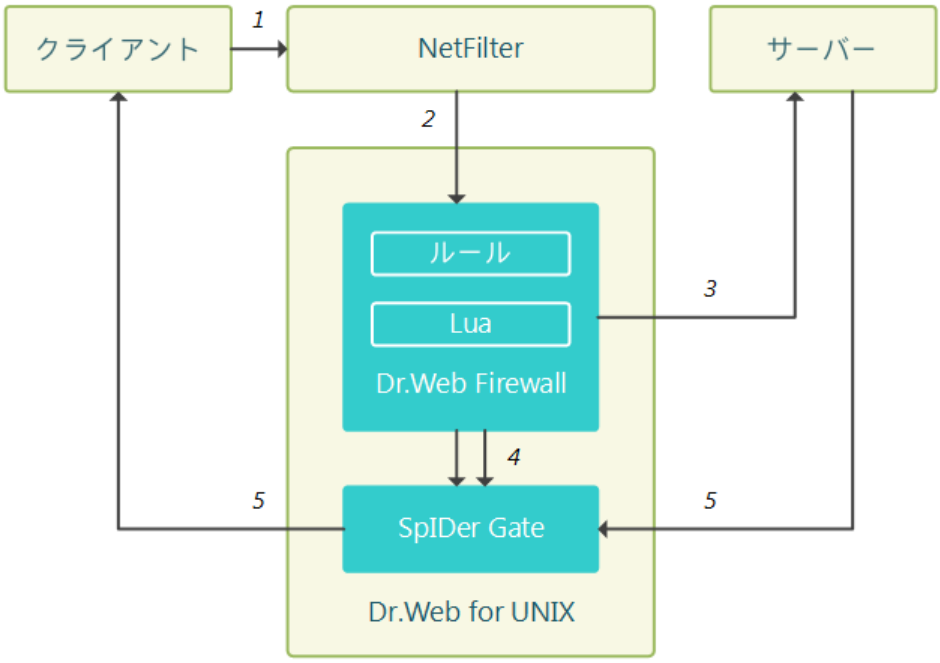

図13. コンポーネントの動作図

以下の接続処理のステップには番号が付いています。

1. クライアントがサーバーへの接続を試みます。
2. ルーティングルールに従って、NetFilterの接続をDr.Web Firewall for Linuxにリダイレクトします。
3. Dr.Web Firewall for Linuxは、クライアントの名前と接続スキャンを使用して、サーバーへの接続を試みます。
4. 接続のクライアント側とサーバー側、接続処理用のSpIDer Gate、およびスキャンの設定とルールに関連するソケットペアを送信します。
5. メディエーターとしてSpIDer Gateを介し、サーバーとクライアント間でデータ交換を行います。

Dr.Web Firewall for Linuxを正しく動作させるためには、ルーティングテーブルに正しい数のビットマーク、*NFQUEUE*キュー、接続監視用のネットワークソケットアドレスを使用するためのルールが必要です。デフォルト設定では、コンポーネントは必要なルール設定を自動的に実行します。設定で接続の自動設定が無効になっている場合、コンポーネントを起動するときに必要なルールを手動で入力する必要があります。

コマンドライン引数

Dr.Web Firewall for Linuxを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-firewall [<options>]
```

Dr.Web Firewall for Linuxは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-firewall --help
```

このコマンドはDr.Web Firewall for Linuxに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-firewall`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[LinuxFirewall]`セクションで指定されている設定パラメータを使用します。

- コンポーネントパラメータ
- トラフィックモニタリングとアクセスブロックのルール

コンポーネントパラメータ

セクションには以下のパラメータが含まれています。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-firewall • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-firewall • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-firewall
XtablesLockPath *{path to file}*	iptables（NetFilter）テーブルブロックファイルへのパス。パラメータ値が指定されていない場合は、/run/xtables.lockと/var/run/xtables.lockパスが検査されます。指定されたパスまたはデフォルトのパスにファイルが見つからない場合は、コンポーネントを起動したときにエラーが発生します。 デフォルト値：*（指定なし）*
InspectFtp *{On / Off}*	FTPプロトコルを介して転送されたデータをスキャンします。 データはルールに従ってスキャンされます（下記参照）。 デフォルト値：On
InspectHttp *{On / Off}*	HTTPプロトコルを介して転送されたデータをスキャンします。 データはルールに従ってスキャンされます（下記参照）。 デフォルト値：On
InspectSmtp *{On / Off}*	SMTPプロトコルを介して転送されたデータをスキャンします（インストールされている場合は、Dr.Web MailDが使用されます）。 実際のデータスキャンは、指定されたスキャンルールに従って実行されます（下記参照）。

パラメータ	説明
	デフォルト値：On
InspectPop3 *{On / Off}*	POP3プロトコルを介して転送されたデータをスキャンします（インストールされている場合は、Dr.Web MailDが使用されます）。 実際のデータスキャンは、指定されたスキャンルールに従って実行されます（下記参照）。 デフォルト値：On
InspectImap *{On / Off}*	IMAPプロトコルを介して転送されたデータをスキャンします（インストールされている場合は、Dr.Web MailDが使用されます）。 実際のデータスキャンは、指定されたスキャンルールに従って実行されます（下記参照）。 デフォルト値：On
AutoconfigureIptables *{Yes / No}*	iptablesインターフェースを介してNetFilterシステムコンポーネントを設定するためのルール。 使用可能な値： • Yes - コンポーネントを起動するときにNetFilterのルールを設定し、動作を自動的に終了するときにルールを削除します（*推奨*）。 • No - ルールを自動的に設定しません。必要なルールは、コンポーネントを起動する前に管理者が手動で追加し、動作が完了した後に削除する必要があります。 ⚠ iptablesのルールの自動設定が許可されていない場合は、iptablesに必要なルールがコンポーネント動作開始前に利用できるようになっている必要があります。 デフォルト値：Yes
AutoconfigureRouting *{Yes / No}*	ip routeとip ruleのルーティングルールとポリシーの設定モード。 使用可能な値： • Yes - コンポーネントを起動するときにip routeとip ruleのルーティングルールとポリシーを設定し、動作を自動的に終了するときにルーティングルールとポリシーを削除します（*推奨*）。 • No - ルールを自動的に設定しません。必要なルールは、コンポーネントを起動する前に管理者が手動で追加し、動作が完了した後に削除します。

パラメータ	説明
	 ルーティングルールとポリシーの自動設定が許可されていない場合は、`ip route`と`ip rule`に必要なルールがコンポーネント動作開始前に利用できるようになっている必要があります。 デフォルト値：`Yes`
`LocalDeliveryMark` *{integer / Auto}*	接続を監視するためにDr.Web Firewall for Linuxネットワークソケット（`TproxyListenAddress`パラメータで指定、下記参照）にリダイレクトされるパケットの *<LDM>*マーク。 使用可能な値： • *<integer>* - パケットの *<LDM>*マーク。2^Nに等しく、*N*がパケット内の`LDM`ビット数の場合、$0 \le N \le 31$。 • `Auto` - Dr.Web Firewall for Linuxはパケットマークの適切なビット数を自動的に選択できます（*推奨*）。  *<LDM>*番号を手動で割り当てるときは、（NetFilter経由を含む）ルート接続とパケットを管理する他のアプリケーションが、パケットマークで相応するビット数を使用していないことを確認してください。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`、`AutoconfigureRouting = No`の場合、指定された *<LDM>*番号は、手動で追加する必要があるルーティングルールで使用してください。 デフォルト値：`Auto`
`ClientPacketsMark` *{integer / Auto}*	接続を開始するクライアントとDr.Web Firewall for Linuxの間で転送されるパケットの *<CPM>*マーク。 使用可能な値： • *<integer>* - パケットの *<CPM>*マーク。2^Nに等しく、*N*がパケット内の`CPM`ビット数の場合、$0 \le N \le 31$。 • `Auto` - Dr.Web Firewall for Linuxはパケットマークの適切なビット数を自動的に選択できます（*推奨*）。

パラメータ	説明
	 *<CPM>*番号を手動で割り当てるときは、(NetFilter経由を含む)ルート接続とパケットを管理する他のアプリケーションが、パケットマークで、対応するビット数を使用していないことを確認してください。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`の場合、指定された*<CPM>*番号は、手動で追加する必要があるルーティングルールで使用する必要があります。 デフォルト値:`Auto`
`ServerPacketsMark` *{integer / Auto}*	Dr.Web Firewall for Linuxと接続を受信するサーバーの間で転送されるパケットの*<SPM>*マーク。 使用可能な値: • *<integer>* - パケットの*<SPM>*マーク。2^Nに等しく、*N*がパケット内の`SPM`ビット数の場合、$0 \le N \le 31$。 • `Auto` - Dr.Web Firewall for Linuxはパケットマークの適切なビット数を自動的に選択できます(*推奨*)。  *<SPM>*番号を手動で割り当てるときは、(NetFilter経由を含む)ルート接続とパケットを管理する他のアプリケーションが、パケットマークで相応するビット数を使用していないことを確認してください。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`、`AutoconfigureRouting = No`の場合、指定された*<SPM>*番号は、手動で追加する必要があるルーティングルールで使用してください。 デフォルト値:`Auto`
`TproxyListenAddress` *{network socket}*	Dr.Web Firewall for Linuxが監視した接続を受信するネットワークソケット(*<IP address>*:*<port>*)。ポート0を指定すると、システムによって自動的に選択されます。

パラメータ	説明
	 該当するソケットが他のアプリケーションによって使用されていないことを確認する必要があります。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`の場合、指定されたIPアドレスとポートは、手動で追加する必要があるルーティングルールで使用してください。 デフォルト値：`127.0.0.1:0`
`OutputDivertEnable` *{Yes / No}*	受信接続（つまりローカルホストに接続があるリモートホストのアプリケーションによって開始された接続）の監視モードを有効／無効にします。 使用可能な値： • `Yes` - 送信接続の監視と処理を行います。 • `No` - 送信接続の監視と処理は行いません。  この設定は、`AutoconfigureIptables = No`の場合に、手動で追加または削除されるルーティングルール番号5を追加または削除します。 デフォルト値：`No`
`OutputDivertNfqueueNumber` *{integer / Auto}*	Dr.Web Firewall for Linuxが、送信接続を開始するSYNパッケージを取得するキュー番号 *NFQUEUE* 使用可能な値： • `<integer>` - *NFQUEUE*で監視された送信接続のSYNパケットを監視するためのキュー番号 *<ONum>* • `Auto` - Dr.Web Firewall for Linuxは適切なキュー番号を自動的に選択できます（*推奨*）。

パラメータ	説明
	 *<ONum>*番号を手動で割り当てるときは、(NetFilterルール経由を含む)接続とパケットを管理する他のアプリケーションが、相応するキュー番号を使用していないことを確認してください。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`の場合、指定された*<ONum>*番号は、手動で追加する必要があるルーティングルール番号5で使用する必要があります。 デフォルト値:`Auto`
`OutputDivertConnectTransparently` *{Yes / No}*	送信接続のために監視されたパケットの送信者(クライアント)のIPアドレスを使用して、受信者(サーバー)に接続するためのエミュレーションモードを有効/無効にします。 使用可能な値: • `Yes` - 接続を監視するときに、自分のアドレスの代わりに接続をリクエストしたクライアントのアドレスを使用してサーバーに接続します。 • `No` - Dr.Web Firewall for Linuxアドレスからサーバーに接続します。 クライアントとDr.Web Firewall for Linuxアドレスは通常、送信接続監視モードでは同じであるため、デフォルト値は`No`になります。 デフォルト値:`No`
`InputDivertEnable` *{Yes / No}*	受信接続(つまりローカルホストに接続があるリモートホストのアプリケーションによって開始された接続)の監視を有効/無効にします。 使用可能な値: • `Yes` - 受信接続の監視と処理を有効にします。 • `No` - 受信接続の監視と処理を無効にします。  この設定は、`AutoconfigureIptables = No`の場合に、手動で追加または削除されるルーティングルール番号6を追加または削除します。無効な値を指定した場合、コンポーネントの起動は失敗します。 デフォルト値:`No`

パラメータ	説明
`InputDivertNfqueueNumber` *{integer / Auto}*	Dr.Web Firewall for Linuxが、受信接続を開始するSYNパッケージを取得するキュー番号 *NFQUEUE* 使用可能な値： • `<integer>` - *NFQUEUE*で監視された送信接続のSYNパケットを監視するためのキュー番号 *<INum>* • `Auto` - Dr.Web Firewall for Linuxは適切なキュー番号を自動的に選択できます（*推奨*）。 ⚠ *<INum>*番号を手動で割り当てるときは、（NetFilterルール経由を含む）接続とパックを管理する他のアプリケーションが、相応するキュー番号を使用していないことを確認してください。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`の場合、指定された *<INum>*番号は、手動で追加する必要があるルーティングルール番号6で使用する必要があります。 デフォルト値：`Auto`
`InputDivertConnectTransparently` *{Yes / No}*	受信接続のために監視されたパケットの送信者（クライアント）のIPアドレスを使用して、受信者（サーバー）に接続するためのエミュレーションモードを有効／無効にします。 使用可能な値： • `Yes` - 接続を監視するときに、自分のアドレスの代わりに接続をリクエストしたクライアントのアドレスを使用してサーバーに接続します。 • `No` - Dr.Web Firewall for Linuxアドレスからサーバーに接続します。 受信接続監視モードでは、すべてのトラフィックがDr.Web Firewall for Linuxを通過するため、不正なクライアントのアドレスを使用してサーバーに安全に接続する可能性があります。これがデフォルト値が`Yes`である理由です。 デフォルト値：`Yes`
`ForwardDivertEnable` *{Yes / No}*	トランジット接続（つまりその他のリモートホストに接続があるリモートホストのアプリケーションによって開始された接続）の監視を有効／無効にします。 使用可能な値： • `Yes` - トランジット接続の監視と処理を有効にします。 • `No` - トランジット接続の監視と処理を無効にします。

パラメータ	説明
	 この設定は、`AutoconfigureIptables = No`の場合に、手動で追加または削除されるルーティングルール番号7を追加または削除します。 デフォルト値：`No`
`ForwardDivertNfqueueNumber` *{integer / Auto}*	Dr.Web Firewall for Linuxが、トランジット接続を開始するSYNパッケージを取得するキュー番号 *NFQUEUE* 使用可能な値： • `<integer>` - *NFQUEUE*で監視されたトランジット接続のSYNパケットを監視するためのキュー番号 *<FNum>* • `Auto` - Dr.Web Firewall for Linuxは適切なキュー番号を自動的に選択できます（*推奨*）。  *<FNum>*番号を手動で割り当てるときは、（NetFilterルール経由を含む）接続とパックを管理する他のアプリケーションが、相応するキュー番号を使用していないことを確認してください。無効な値を指定した場合、コンポーネントの起動は失敗します。 `AutoconfigureIptables = No`の場合、指定された *<FNum>*番号は、手動で追加する必要があるルーティングルール番号7で使用してください。 デフォルト値：`Auto`
`ForwardDivertConnectTransparently` *{Yes / No}*	トランジット接続のために監視されたパケットの送信者（クライアント）のIPアドレスを使用して受信者（サーバー）に接続するためのエミュレーションモード。 使用可能な値： • `Yes` - 接続を監視するときに、自分のアドレスの代わりに接続をリクエストしたクライアントのアドレスを使用してサーバーに接続します。 • `No` - Dr.Web Firewall for Linuxアドレスからサーバーに接続します。 トランジット接続監視モードでは、すべてのトラフィックがDr.Web Firewall for Linuxがインストールされているのと同じホスト（ルーター）を通過するという保証はないため、正しい動作のデフォルト値は`No`になります。保護されたアプリケーションが同じルーターを使用することがネットワーク設定によって保証されている場合、このパラメータを`Yes`に設定できます。この場合は、Dr.Web Firewall for Linuxがサー

パラメータ	説明
	バーに接続するときは常にクライアントのアドレスへの接続を評価します。 デフォルト値：No
ExcludedProc *{path to file}*	プロセスのホワイトリスト（ネットワークアクティビティが監視されないプロセス） リストをパラメータ値として指定できます。リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：プロセスのリストにwgetとcurlを追加します。 1. 設定ファイルに値を追加します。 • 1行に2つの値： `[LinuxFirewall]` `ExcludedProc = "/usr/bin/wget", "/usr/bin/curl"` • 2行（1行に1つの値）： `[LinuxFirewall]` `ExcludedProc = /usr/bin/wget` `ExcludedProc = /usr/bin/curl` 2. `drweb-ctl cfset`コマンドを使用して値を追加します。 `# drweb-ctl cfset LinuxFirewall.ExcludedProc -a /usr/bin/wget` `# drweb-ctl cfset LinuxFirewall.ExcludedProc -a /usr/bin/curl` ⚠ このパラメータで示されるプロセスリストの実際の使用方法は、Dr.Web Firewall for Linuxに定義されているスキャンルールでの使用*方法*によって決まります。 デフォルトルールのリスト（以下参照）は、リストからの全プロセスのトラフィックが*スキャンせずに*許可されることを保証します。 デフォルト値：*（未設定）*
UnwrapSsl *{Boolean}*	SSL/TLS接続を介して転送された暗号化トラフィックをスキャンします。

パラメータ	説明
	 最近の実行例では、この変数の値は保護されたトラフィックの処理に影響しません。処理を管理するには、SET Unwrap_SSL = true/falseアクションを含むルールを作成する必要があります（以下参照）。 drweb-ctlユーティリティのcfsetコマンドまたはWebインターフェースを使用してこのパラメータの値を変更した場合、影響を受ける依存ルールが自動的に適応します。 デフォルト値：No
BlockInfectionSource *{Boolean}*	悪意のあるソフトウェア（*InfectionSource*カテゴリーに含まれる）を含むWebサイトへの接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：Yes
BlockNotRecommended *{Boolean}*	非推奨サイト（*NotRecommended*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：Yes
BlockAdultContent *{Boolean}*	アダルトコンテンツ（*AdultContent*カテゴリーに含まれる）を含むWebサイトへの接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No

パラメータ	説明
BlockViolence *{Boolean}*	暴力的描写（*Violence*カテゴリーに含まれる）を含むWebサイトへの接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockWeapons *{Boolean}*	武器に関するWebサイト（*Weapons*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockGambling *{Boolean}*	ギャンブルのWebサイト（*Gambling*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockDrugs *{Boolean}*	麻薬に関するWebサイト（*Drugs*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockObsceneLanguage *{Boolean}*	卑猥な表現（*ObsceneLanguage*カテゴリーに含まれる）を含むWebサイトへの接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。

パラメータ	説明
	`url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
`BlockChats` *{Boolean}*	チャットWebサイト（*Chats*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
`BlockTerrorism` *{Boolean}*	テロリズムに関するWebサイト（*Terrorism*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
`BlockFreeEmail` *{Boolean}*	無料メールサービスのWebサイト（*FreeEmail*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
`BlockSocialNetworks` *{Boolean}*	ソーシャルネットワーキングWebサイト（*SocialNetworks*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No

パラメータ	説明
BlockDueToCopyrightNotice *{Boolean}*	著作権者のリクエストに従って追加されたWebサイト（*DueToCopyrightNotice*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockOnlineGames *{Boolean}*	オンラインゲームWebサイト（*OnlineGames*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockAnonymizers *{Boolean}*	アノニマイザーWebサイト（*Anonymizers*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockCryptocurrencyMiningPools *{Boolean}*	仮想通貨マイニングのための一般的なサービスへのアクセスを提供するWebサイト（*CryptocurrencyMiningPool*カテゴリーに含まれる）への接続試行をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
BlockJobs *{Boolean}*	求人検索Webサイト（*Jobs*カテゴリーに含まれます）への接続試行をブロックします。

パラメータ	説明
	ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `url_category in "LinuxFirewall.BlockCategory" : Block as _match` デフォルト値：No
`Whitelist` *{domain list}*	ドメインのホワイトリスト（ブロックされたカテゴリーに含まれている場合でも、ユーザーの接続が許可されているドメイン。さらに、このリストに示されているドメインのすべてのサブドメインへユーザーがアクセスすることが許可されます）。 リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：ドメイン`example.com`と`example.net`のリストに追加します。 1. 設定ファイルに値を追加します。 • 1つの文字列に2つの値： `[LinuxFirewall]` `Whitelist = "example.com", "example.net"` • 2つの文字列（文字列ごとに1つの値）： `[LinuxFirewall]` `Whitelist = example.com` `Whitelist = example.net` 2. `drweb-ctl cfset`コマンドを使用して値を追加します。 `# drweb-ctl cfset LinuxFirewall.Whitelist -a example.com` `# drweb-ctl cfset LinuxFirewall.Whitelist -a example.net`

パラメータ	説明
	 このパラメータで示されるドメインリストの実際の使用方法は、Dr.Web Firewall for Linuxに定義されているスキャンルールでの使用*方法*によって決まります。 デフォルトルールのリスト（下記参照）は、ブロックされるWebソースカテゴリーのリストのドメインがこのリストに含まれている場合でも、このリストのドメイン（およびそのサブドメイン）へのアクセスが提供されることを保証します（ただしHTTPプロトコルを経由したサーバーへのリクエストの場合のみ）。さらに、このデフォルトのルールセットは、ホワイトリストドメインからダウンロードしたデータが*脅威に対してスキャンされる*ことを保証します（データはレスポンスで返され、変数`direction`には値`response`があるため）。 デフォルト値：*（未設定）*
`Blacklist` *{domain list}*	*ブラックリストとして使用できる*ドメインのリスト（つまり、これらのドメインがブロックされたカテゴリーに含まれていない場合でも、ユーザーの接続が禁止されているドメインのリスト。さらに、このリストに示されているドメインのすべてのサブドメインへユーザーがアクセスすることが禁止されます）。 リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：ドメイン`example.com`と`example.net`のリストに追加します。 1. 設定ファイルに値を追加します。 • 1つの文字列に2つの値： `[LinuxFirewall]` `Blacklist = "example.com", "example.net"` • 2つの文字列（文字列ごとに1つの値）： `[LinuxFirewall]` `Blacklist = example.com` `Blacklist = example.net`

パラメータ	説明
	2. `drweb-ctl cfset` コマンドを使用して値を追加します。 `# drweb-ctl cfset LinuxFirewall.Blacklist -a example.com` `# drweb-ctl cfset LinuxFirewall.Blacklist -a example.net` ⚠ このパラメータで示されるドメインリストの実際の使用方法は、Dr.Web Firewall for Linuxに定義されているスキャンルールでの使用*方法*によって決まります。 デフォルトルールのリスト（以下参照）は、このリストのドメイン（およびそのサブドメイン）へのアクセスがHTTPプロトコル上で常に禁止されることを保証します。このドメインがホワイトリストとブラックリストに同時に追加される場合、デフォルトのルールにより、そのドメインへのユーザーアクセスは確実にブロックされます。 デフォルト値：*（未設定）*
`ScanTimeout` *{time interval}*	SpIDer Gateによって開始された1つのファイルに対するスキャンのタイムアウト 指定可能な値：1秒（`1s`）から1時間（`1h`）まで。 デフォルト値：`30s`
`HeuristicAnalysis` *{On / Off}*	既知の脅威を検出するためのヒューリスティック解析を有効／無効にします。ヒューリスティック解析における検出の信頼性は高いですが、ウイルススキャンにかかる時間が長くなります。 ヒューリスティックアナライザによって検出された脅威に適用されるアクションは、`BlockSuspicious`パラメータ値として指定します。 使用可能な値： • `On` - スキャン時のヒューリスティック解析を有効にします。 • `Off` - ヒューリスティック解析を無効にします。 デフォルト値：`On`
`PackerMaxLevel` *{integer}*	圧縮されたオブジェクトの最大ネスティングレベル。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮され

パラメータ	説明
	た実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなども含まれている、他の圧縮されたオブジェクトが含まれる場合があります。このパラメータの値はネスティングの上限を指定します。この上限を超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 ネスティングレベルの制限はありません。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
ArchiveMaxLevel *{integer}*	他のアーカイブが含まれる可能性のあるアーカイブ（zip、rarなど）の最大ネスティングレベル（これらのアーカイブには他のアーカイブなどが含まれる場合もあります）。このパラメータの値はネスティングの上限を指定します。この上限を超えると、他のアーカイブに含まれるアーカイブはスキャンされません。 ネスティングレベルの制限はありません。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
MailMaxLevel *{integer}*	他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）の最大ネスティングレベル（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 ネスティングレベルの制限はありません。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
ContainerMaxLevel *{integer}*	他のオブジェクトが含まれる他のタイプのオブジェクト（HTMLページ、jarファイルなど）の最大ネスティング。このパラメータの値はネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 ネスティングレベルの制限はありません。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
MaxCompressionRatio *{integer}*	圧縮されたオブジェクトの最大圧縮率（非圧縮サイズと圧縮サイズの比率）。オブジェクトの比率が制限を超えると、SpIDer Gateによって開始されたファイルスキャン中にそのオブジェクトはスキップされます。 圧縮率には2よりも小さい値は指定できません。 デフォルト値：500

パラメータ	説明
BlockKnownVirus *{Boolean}*	既知の脅威を含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：Yes
BlockSuspicious *{Boolean}*	ヒューリスティックアナライザによって検出された未知の脅威を含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：Yes
BlockAdware *{Boolean}*	アドウェアを含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：Yes
BlockDialers *{Boolean}*	ダイアラープログラムを含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：Yes
BlockJokes *{Boolean}*	ジョークプログラムを含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。

パラメータ	説明
	`threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：No
`BlockRiskware` *{Boolean}*	リスクウェアを含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：No
`BlockHacktools` *{Boolean}*	ハッキングツールを含むデータの受信または送信をブロックします。 ブロックをアクティブにするには、次のルールを設定に追加する必要があります（以下の詳細を参照）。 `threat_category in "LinuxFirewall.BlockThreat" : Block as _match` デフォルト値：No
`BlockUnchecked` *{Boolean}*	スキャンできないトラフィックをブロックします。 このパラメータ値は、エラーのためにtrueまたはfalseに評価することが不可能なルールの処理に影響します。Noを指定した場合、ルールは実行されていないルールとしてスキップされます。Yesを指定した場合、Block as BlackListアクションが実行されます。 デフォルト値：No
`InterceptHook` *{path to file / Lua function}*	Luaで接続を処理するためのスクリプトまたはそのスクリプトを含むファイルへのパス（Luaでの接続処理セクションを参照）。 使用できないファイルを指定すると、コンポーネントを読み込む際にエラーが表示されます。 デフォルト値： `local dwl = require 'drweb.lookup'`

パラメータ	説明
	`function intercept_hook(ctx)` `  -- do not check if group == Root.TrustedGroup` `  if ctx.divert == "output" and ctx.group == "drweb"` `  then` `  return "pass"` `  end` `  -- do not check connections from privileged ports` `  -- except FTP active mode` `  if ctx.src.port >= 0 and ctx.src.port <= 1024` `  and ctx.src.port ~= 20` `  then` `  return "pass"` `  end` `  return "check"` `end`

接続スキャンの設定を変更しても、変更を加える前にアプリケーションによってすでに確立されている接続のスキャンには影響しません。それらをすでに実行中のアプリケーションに適用する必要がある場合は、アプリケーションを再起動するなどして、アプリケーションを強制的に切断してから再度接続する必要があります。

トラフィックモニタリングとアクセスブロックのルール

このセクションには、上記のパラメータの他、11個の*ルールセット*`RuleSet*`（`RuleSet0`、...、`RuleSet10`）も含まれます。これらは、トラフィックスキャン、Webリソースへのユーザーアクセスのブロック、インターネットからのコンテンツのダウンロードのブロックを直接管理します。条件の一部の値（IPアドレス範囲、Webサイトカテゴリーのリスト、Webソースのホワイト／ブラックリストなど）については、テキストファイルから読み込まれる値の置き換えがあり、LDAPを介して外部データソースから抽出されます（Dr.Web LookupDコンポーネントが使用されます）。接続を設定する際には、最終的な解決を含むルールが見つかるまで、すべてのルールが昇順で検査されます。ルールリストのギャップは無視されます。

ルールを表示して編集する

ルールリストを簡単に編集するために未設定のものが残されています。つまり、ルールが指定されていない`RuleSet` *<i>*があります（*<i>* - `RuleSet`ルールセット番号）。`RuleSet` *<i>*以外の項目を追加することは*できませんが*、`RuleSet` *<i>*の要素内のルールは追加および削除できます。ルールの表示と編集は、次のいずれかの方法で実行できます。

- （任意のテキストエディターで）設定ファイル設定ファイルを表示して編集する（このファイルにはデフォルトとは異なる値のパラメータのみが保存されます）。
- Web管理インターフェースを使用する（インストールされている場合）。

- コマンドラインベースのインターフェースDr.Web Ctl(`drweb-ctl cfshow`および`drweb-ctl cfset`コマンド)を使用する。

> ルールを編集して設定ファイルを変更した場合は、その変更を適用するためにDr.Web for UNIX Mail Serversを再起動します。それには`drweb-ctl reload`コマンドを使用します。

ルールを表示するには、`drweb-ctl cfshow`コマンドを使用します。

ルールセット`LinuxFirewall.RuleSet1`のコンテンツを表示するには、次のコマンドを使用します。

```
# drweb-ctl cfshow LinuxFirewall.RuleSet1
```

ルールを編集するには、`drweb-ctl cfset`コマンドを使用します(以降、*<rule>*はルールのテキストです)。

- `LinuxFirewall.RuleSet1`セットのすべてのルールを新しいルールで置き換えます。

```
# drweb-ctl cfset LinuxFirewall.RuleSet1 '<rule>'
```

- `LinuxFirewall.RuleSet1`ルールセットに新しいルールを次のように追加します。

```
# drweb-ctl cfset -a LinuxFirewall.RuleSet1 '<rule>'
```

- `LinuxFirewall.RuleSet1`セットから特定のルールを次のように削除します。

```
# drweb-ctl cfset -e LinuxFirewall.RuleSet1 '<rule>'
```

- `LinuxFirewall.RuleSet1`ルールセットを次のようにデフォルトの状態にリセットします。

```
# drweb-ctl cfset -r LinuxFirewall.RuleSet1
```

`drweb-ctl`ツールを使用してルールのリストを編集するときは、追加するルールのテキストを一重引用符または二重引用符で囲み、ルール自体のテキストに二重引用符が含まれている場合には、ルールのテキスト内にある二重引用符の前にバックスラッシュ(「\」)をエスケープ文字として使用します。

設定の`RuleSet`*<i>*変数には、ルールの格納について次のような特徴があります。

- 無条件ルールを追加するときは、条件部分とコロンを省略できます。ただし、そのようなルールは常にルールのリストに文字列「 : *<action>*」として格納されます。
- 複数のアクションを含むルール(「*<condition>* : *<action 1>*, *<action 2>*」など)を追加すると、そのようなルールは基本ルールのチェーン「*<condition>* : *<action 1>*」と「*<condition>* : *<action 2>*」に変更されます。
- ロギングまたはルールは、条件部分で条件の離接(論理和)を許可しないため、論理和を実装するには、各ルールに離接語条件がある条件で一連のルールを記録する必要があります。

接続をスキップするための無条件ルール(`Pass`アクション)を`LinuxFirewall.RuleSet1`セットに追加するには、次のコマンドのみを実行します。

```
# drweb-ctl cfset -a LinuxFirewall.RuleSet1 'Pass'
```

ただし、指定したルールセットからこのルールを削除するには、次のコマンドを実行する必要があります。

```
# drweb-ctl cfset -e LinuxFirewall.RuleSet1 ' : Pass'
```

LinuxFirewall.RuleSet1ルールを、接続用の標準テンプレートへのパスを未解決アドレスから変更してブロックを実行するルールセットに追加するには、次のコマンドを実行する必要があります。

```
# drweb-ctl cfset -a LinuxFirewall.RuleSet1 'src_ip not in
file("/etc/trusted_ip") : set http_template_dir = "mytemplates", Block'
```

ただし、このコマンドでは指定したセットに*2つのルール*が追加されるため、ルールのセットからこれらのルールを削除するには、次の2つのコマンドを実行する必要があります。

```
# drweb-ctl cfset -e LinuxFirewall.RuleSet1 'src_ip not in
file("/etc/trusted_ip") : set http_template_dir = "mytemplates"'
# drweb-ctl cfset -e LinuxFirewall.RuleSet1 'src_ip not in
file("/etc/trusted_ip") : Block'
```

「悪意のあるオブジェクト*KnownVirus*や*Terrorism*カテゴリーに該当するURLが検出された場合はブロックする」などのルールをLinuxFirewall.RuleSet1ルールセットに追加するには、このルールセットに次の2つのルールを追加する必要があります。

```
# drweb-ctl cfset -a LinuxFirewall.RuleSet1 'threat_category in (KnownVirus)
: Block as _match'
# drweb-ctl cfset -a LinuxFirewall.RuleSet1 'url_category in (Terrorism) :
Block as _match'
```

ルールのセットからこれらのルールを削除する場合にも、2つのコマンドを実行する必要があります（上記の例を参照）。

デフォルトのルールセット

デフォルトでは、次のブロックルールセットが指定されています。

```
RuleSet0 =
RuleSet1 = divert output : set HttpTemplatesDir = "output"
RuleSet1 = divert output : set MailTemplatesDir = "firewall"
RuleSet1 = divert input : set HttpTemplatesDir = "input"
RuleSet1 = divert input : set MailTemplatesDir = "server"
RuleSet1 = proc in "LinuxFirewall.ExcludedProc" : Pass
RuleSet1 =  : set Unwrap_SSL = false
RuleSet2 =
RuleSet3 =
RuleSet4 =
RuleSet5 = protocol in (Http), direction request, url_host in
"LinuxFirewall.Blacklist" : Block as BlackList
RuleSet5 = protocol in (Http), direction request, url_host in
"LinuxFirewall.Whitelist" : Pass
RuleSet6 =
RuleSet7 = protocol in (Http), direction request, url_category in
"LinuxFirewall.BlockCategory" : Block as _match
RuleSet8 =
```

```
RuleSet9 = protocol in (Http), divert input, direction request,
threat_category in "LinuxFirewall.BlockThreat" : Block as _match
RuleSet9 = protocol in (Http), direction response, threat_category in
"LinuxFirewall.BlockThreat" : Block as _match
RuleSet9 = protocol in (Smtp), threat_category in
"LinuxFirewall.BlockThreat" : REJECT
RuleSet9 = protocol in (Smtp), url_category in "LinuxFirewall.BlockCategory"
: REJECT
RuleSet9 = protocol in (Smtp), total_spam_score gt 0.80 : REJECT
RuleSet9 = protocol in (Pop3, Imap), threat_category in
"LinuxFirewall.BlockThreat" : REPACK as _match
RuleSet9 = protocol in (Pop3, Imap), url_category in
"LinuxFirewall.BlockCategory" : REPACK as _match
RuleSet9 = protocol in (Pop3, Imap), total_spam_score gt 0.80 : REPACK as
_match
RuleSet10 =
```

最初のルールは、ExcludedProcパラメータ（上記を参照）で指定されたプロセスによって接続が確立された場合、他の条件を検査せずに接続をスキップすることを示します。次のルール（条件なしで実行されます）は保護された接続のラップ解除をブロックします。このルールとその下位にあるすべてのルールは、接続が除外されたプロセスで設定されていない場合にのみ考慮されます。また、後続のすべてのルールはプロトコルに依存するため、保護された接続のラップ解除が無効になっている場合は、条件がtrueと評価されるかどうかを定義できないため、ルールは実行されません。

次のルールは、送信HTTP接続に対する処理を制御します。

1. 接続が確立されたホストがブラックリストに含まれている場合、そのホストがブラックリストに含まれているため、接続はブロックされます。他のスキャンは実行されません。
2. ホストがホワイトリストに含まれている場合、接続はスキップされ、他のスキャンは実行されません。
3. クライアントからリクエストされたURLが、望ましくないWebリソースのカテゴリーに含まれている場合は、脅威が検出されたために接続がブロックされます。他のスキャンは実行されません。
4. HTTP経由の脅威があるリモートホストから受信したレスポンスに、ブロックされたカテゴリーに属する脅威が含まれる場合、その脅威が検出されたために接続がブロックされます。他のスキャンは実行されません。
5. ローカルホストからリモートホストに移行したデータに、ブロックされたカテゴリーに属する脅威が含まれる場合、その脅威が検出されたために接続がブロックされます。他のスキャンは実行されません。

この5つのルールは、InspectHttpパラメータでOnが指定されている場合にのみ機能します。それ以外の場合、これらのルールはどれも機能しません。

RuleSet9で指定される次の6つのルールは、メールプロトコル（SMTP、POP3、IMAPプロトコル）を介して送受信されるデータのスキャンを制御します。これらのルールは、次の場合に有効になります。

- 送信されたメールメッセージに、添付ファイルが含まれている場合
- 送信されたメールメッセージに、望ましくないカテゴリーに属するURLが含まれている場合
- 送信されたメールメッセージがスパムとして分類された場合（スパムインデックスが0.8以上）

メールメッセージがSMTPプロトコルを介して送信される場合、メールの伝送（つまり送信または受信）はブロックされますが、IMAPとPOP3プロトコルの場合は、メールはそのコンテンツから悪意のあるコンテンツを削除するために処理（「再パッケージ」）されます。

メールメッセージのスパムの兆候をスキャンするためのコンポーネントであるDr.Web Anti-Spamが使用できない場合、スパムの兆候をスキャンするためのメールメッセージのスキャンは実行されません。この場合、スパムレベルのスキャン（`total_spam_score`値）を含むルールは使用できません。

メール処理ルールは、相応する`Inspect <EmailProtocol>`パラメータに`On`が指定されている場合にのみ実行されることに注意してください。それ以外の場合、これらのルールはどれも実行されません。さらに、送信されたメールメッセージのマルウェアの添付ファイルとスパムの兆候を調べるために、メールスキャン用のDr.Web MailDコンポーネントをインストールする必要があります。このコンポーネントがインストールされていない場合、*「検査できません」*というエラーのため、送信されたメールはブロックされます。検査できないメッセージの送信を許可するには、`BlockUnchecked = No`パラメータを設定します（上記を参照）。また、メールスキャンコンポーネントがインストールされていない場合は、`InspectSmtp`、`InspectPop3`、`InspectImap`パラメータに`No`を指定することをお勧めします。

トラフィックモニタリングとアクセスブロックのルールの例

1. IPアドレス範囲が*10.10.0.0～10.10.0.254*のユーザーに*Chats*以外のすべてのカテゴリーのWebサイトへのHTTPアクセスを許可する。

   ```
   protocol in (HTTP), src_ip in (10.10.0.0/24), url_category not in (Chats)
   : Pass
   ```

   ルール

   ```
   protocol in (HTTP), url_host in "LinuxFirewall.Blacklist" : Block as
   BlackList
   ```

   が、指定されたルールより上のルールリストに割り当てられている場合、IPアドレス範囲が*10.10.0.0～10.10.0.254*のユーザーは、ブラックリストのドメイン、つまりパラメータ`LinuxFirewall.Blacklist`にリストされているドメインへのアクセスもブロックされます。また、このルールが下に割り当てられている場合、IPアドレス範囲が*10.10.0.0～10.10.0.254*のユーザーは、ブラックリストのWebサイトにもアクセスできます。

   アクション`Pass`は最終的なものであるため、これ以上のルールはチェックされず、したがってダウンロードされたデータのウイルススキャンも実行されません。IPアドレス範囲が*10.10.0.0～10.10.0.254*のユーザーに、ブラックリストに含まれていない*Chats*以外のすべてのカテゴリーのWebサイトへのアクセスを許可し、同時に脅威のダウンロードをブロックするには、次のルールを使用します。

   ```
   protocol in (HTTP), url_category not in (Chats), url_host not in
   "LinuxFirewall.Blacklist", threat_category not in
   "LinuxFirewall.BlockCategory" : Pass
   ```

2. *インターネットからダウンロードした*ビデオファイルのコンテンツ（つまり、MIMEタイプが「`video/*`」のデータ（`*`はMIMEクラス`video`の任意のタイプ））のスキャンは実行しない。

   ```
   direction response, content_type in ("video/*") : Pass
   ```

   ローカルコンピューターから読み込んだファイル（MIMEタイプが「`video/*`」のファイルを含む）は、*レスポンス*ではなく*リクエスト*で送信されるために（つまり変数`direction`には値`request`があるため）スキャンされます。

Luaでの接続処理

このセクションの内容：

- 概要
- 接続処理のスクリプトにおける要件
- 例
- 使用中のテーブル
- 利用可能な補助モジュール

概要

Dr.Web Firewall for LinuxはLuaのプログラムインタプリタを介して対話をサポートします（バージョン5.3.4が使用され、Dr.Web for UNIX Mail Serversと一緒に提供されます）。Luaで書かれたスクリプトは、解析のためにSpIDer Gateに送信される前に、事前接続スキャンのためにコンポーネントによって使用されます。

このスクリプトへのパスがDr.Web Firewall for Linux設定（`InterceptHook`パラメータ）で指定されている場合、接続はLuaスクリプトで解析されます。それ以外の場合は、コンポーネント設定で指定されているデフォルト設定と処理ルール（`RuleSet*`パラメータ）を使用して接続処理が実行されます。

接続処理用のLuaスクリプトのその他の例については、次のリンクからご確認ください。
https://github.com/DoctorWebLtd/drweb-lua-examples/tree/master/firewall

接続処理のスクリプトにおける要件

このスクリプトには、接続スキャンモジュールのエントリポイントであるグローバル関数が含まれている必要があります（Dr.Web Firewall for Linuxは、新しく受信した接続を処理するためにこの関数を呼び出します）。この関数は、次の呼び出し規則を満たす必要があります。

- *関数名* - `intercept_hook`
- *引数* - Luaコンテキストテーブルのみ（処理された接続の関数からの情報へのアクセスを提供します。以下の表の説明を参照。）
- *戻り値* - 以下の表のいずれかの文字列値のみ

値	判定の説明
`pass`	SpIDer Gateによる接続のチェックを行わず、スキップします。
`check`	SpIDer Gateを使用して接続をチェックします。
`reject`	接続を破棄します（接続を開始したクライアントは、`RST`フラグ付きのTCPパッケージを受信します）。
`drop`	切断します（接続を開始したクライアントは確認応答を受信しません）。

例

1. 次のスクリプトは、Dr.Web Firewall for Linuxによって接続がチェックされないように、すべての接続に対して常にpass判定（スキップ）を返します。

```
-- Function of connection scanning written by the user
function intercept_hook(ctx)
  return "pass" -- do not scan the connection
end
```

2. 次のスクリプトを使用すると、以下の例外を除いて、Dr.Web Firewall for Linuxは確立されているすべての接続をチェックします。
   - drwebグループのユーザー権限で実行されているアプリケーションからの送信ローカル接続
   - （接続の所有者とその方向に関係なく）権限のあるポートから開始された接続
   - ローカルネットワークからのIPアドレスから始まる接続

```
function intercept_hook(ctx)
  -- Do not scan connections, initiated from the local
  -- host (divert == "output") by application under the name of
  -- "drweb" (group == "[drweb]")
  if ctx.divert == "output" and ctx.group == "drweb" then
    return "pass"
  end

  -- Do not scan connections, initiated from
  -- privileged ports (range is from 0 to 1024)
  if ctx.src.port >= 0 and ctx.src.port <= 1024 then
    return "pass"
  end

  -- Do not scan connections from local network IP addresses
  -- (IP address range 127.0.0.1/8)
  if ctx.src.ip.belongs("127.0.0.0/8") then
    return "pass"
  end

  -- Connection is scanned by default
  return "check"
end
```

スクリプトで使用されるテーブル

1. InterceptContextテーブル

このテーブルは、処理された接続に関するデータをintercept_hook関数に転送するために使用されます。このデータに基づいて、次のいずれかのアクションを接続に対して実行できます。

- チェックせずにスキップする。
- 接続を中断する。
- チェックのためにSpIDer Gateに接続を送信する。

Dr.Web Firewall for Linuxではテーブルにデータを入力します。テーブルのデータの中には、intercept_hook関数が実行されるまでに使用できるものもあります。他のデータ(いわゆる「遅延」データ)は、テーブルの該当するフィールドのリクエストに応じて直接計算されます。

フィールド	説明	データタイプ
src	接続を開始したクライアントのアドレスとポート 例: `if ctx.src.port >= 0 and ctx.src.port <= 1024 then return "pass" end`	TcpEndpointテーブル
dst	クライアントによって接続が開始されたサーバーのアドレスとポート 例: `if ctx.dst.ip.belongs("10.20.30.41/8") then return "reject" end`	TcpEndpointテーブル
divert	監視される接続のタイプ: • "output" - 送信接続 • "input" - 受信接続 • "forward" - トランジット接続 例: `if ctx.divert == "forward" then return "check" end`	文字列
iface_in	接続が開始されたインターフェースの名前 インターフェース名が特定されていない場合は、nil値になります。	文字列
iface_out	接続が開始された後にパケットが送信されたインターフェースの名前 インターフェース名が特定されていない場合は、nil値になります。	文字列
uid	発信接続を開始したユーザーのID 接続タイプ(divert)が"output"ではない場合、またはUIDを特定できない場合は、nil値になります。	番号
gid	その権限により発信接続が開始されたグループのID 接続タイプ(divert)が"output"ではない場合、またはGIDを特定できない場合は、nil値になります。	番号
user	発信接続を開始したユーザーの名前	文字列

フィールド	説明	データタイプ
	接続タイプ(`divert`)が"`output`"ではない場合、またはUIDを特定できない場合は、`nil`値になります。	
`group`	その権限により発信接続が開始されたグループの名前 接続タイプ(`divert`)が"`output`"ではない場合、またはUIDを特定できない場合は、`nil`値になります。	文字列
`pid`	発信接続を開始したプロセスのID 接続タイプ(`divert`)が"`output`"ではない場合、またはPIDを特定できない場合は、`nil`値になります。	番号
`exe_path`	送信接続を開始したアプリケーションファイルへの実行パス 接続タイプ(`divert`)が"`output`"ではない場合、または実行パスを特定できない場合は、`nil`値になります。	文字列
無効になったメタメソッド: *なし*		

2. TcpEndpointテーブル

このテーブルでは接続ポイント(クライアントまたはサーバー)のアドレスを指定します。

フィールド	説明	データタイプ
`ip`	IPアドレス	IpAddressテーブル
`port`	ポート番号	番号
無効になったメタメソッド: • `__tostring` - TcpEndpointを文字列に変換する関数。例: "`127.0.0.1:443`"(IPv4)または"`[::1]:80`"(IPv6) • `__concat` - TcpEndpointを文字列に連結する関数		

利用可能な補助モジュール

LuaのプログラムスペースでDr.Web for UNIX Mail Serversとやり取りするために、次の特定のモジュールをインポートできます。

モジュール名	機能
drweb	Luaプログラムを起動したDr.Web for UNIX Mail Serversコンポーネントのログに Luaプログラムからのメッセージを記録する機能と、Luaプロシージャの非同期実行の手段を提供するモジュール
drweb.lookup	Dr.Web LookupDモジュールを呼び出して外部ソースからデータを要求するためのツールを提供するモジュール

drwebモジュールの内容

1. 機能

このモジュールには、次のような機能があります。

- Luaプログラムからのメッセージを Dr.Web for UNIX Mail Servers コンポーネントログに保存する:
  - `log(<level>, <message>)` は *<message>* 文字列を Dr.Web for UNIX Mail Servers ログに *<level>* レベル（必要なレベルは、*「debug」*、*「info」*、*「notice」*、*「warning」*、*「error」* を使用して定義します）で書き込みます。
  - `debug(<message>)` は *<message>* 文字列を Dr.Web for UNIX Mail Servers ログに *DEBUG* レベルで書き込みます。
  - `info(<message>)` は *<message>* 文字列を Dr.Web for UNIX Mail Servers ログに *INFO* レベルで書き込みます。
  - `notice(<message>)` は *<message>* 文字列を Dr.Web for UNIX Mail Servers ログに *NOTICE* レベルで書き込みます。
  - `warning(<message>)` は *<message>* 文字列を Dr.Web for UNIX Mail Servers ログに *WARNING* レベルで書き込みます。
  - `error(<message>)` は *<message>* 文字列を Dr.Web for UNIX Mail Servers ログに *ERROR* レベルで書き込みます。
- Luaプロシージャの同期を管理する:
  - `sleep(<sec.>)` はこのLuaプロシージャインスタンスの実行を指定された秒数で一時停止します。
  - `async(<Lua function>[, <argument list>])` は、指定された関数を非同期的に起動し、指定された引数リストに渡します。`async` 関数呼び出しはすぐに完了し、戻り値（`Future` テーブル）を使用すると、*<Lua function>* の結果を取得できます。
- IpAddress テーブルにIPアドレスを追加する:
  - `ip(<address>)` は、IpAddress テーブルの形式で *<address>* 文字列として送信される、IPアドレスを指定します。IPv4またはIPv6アドレスのいずれかを使用できます。
- テキストファイルから外部データをアップロードする:
  - `load_set(<file path>)` は、指定されたテキストファイルのコンテンツから `true` 値を含むテーブルを生成します。ファイルから読み取られた文字列はキーとして使用されます。空の文字列と空白を含む文字列は無視されます。
  - `load_array(<file path>)` は、指定されたテキストファイルのコンテンツから文字列の配列を生成します。空の文字列と空白文字のみで構成される文字列は無視され、配列には含まれません。

2. テーブル

- `Future` テーブルは、`async` 関数を使用して関数を実行した後の保留中の結果を表します。

フィールド	説明	データタイプ
`wait`	`async` 関数を使用して開始した関数の結果を返す関数。関数がまだ実行を完了していない場合は、完了を待って結果を返します。`wait` が呼び出される前に関数が完了した場合、結果はすぐに返されます。開始された関数が失敗した場合、`wait` 呼び出しは同じエラーを生成します。	機能

フィールド	説明	データタイプ
無効になったメタメソッド: *なし*		

- IpAddressテーブルはIPアドレスを表します。

フィールド	説明	データタイプ
`belongs`	IpAddressテーブルに保存されているIPアドレスが、指定されたサブネット(IPアドレス範囲)に所属しているかどうかを確認する関数 "*<IP address>*"または"*<IP address>*/*<mask>*"のような文字列を唯一の引数として受け取ります。ここで、*<IP address>*はホストアドレスまたはネットワークアドレス(`"127.0.0.1"`など)、*<mask>*はサブネットワークマスク(`"255.0.0.0"`などのIPアドレスとして、または`"8"`などの数値形式で指定できます)です。 次のブール値を返します。 • `true`は、アドレスが指定されたアドレスの少なくとも1つと等しいか、指定されたサブネット(IPアドレスの範囲)の少なくとも1つに属していることを示します。 • `false` - それ以外の場合。	機能
無効になったメタメソッド: • `__tostring`は、文字列内のIpAddressを変更する関数(例:`"127.0.0.1"`(IPv4)または`"::1"`(IPv6))です。 • `__concat`は、IpAddressを文字列に結合する関数です。 • `__eq`は、2つのIpAddressが等しいことを確認する関数です。 • `__band`は、マスクを適用するための関数(例:`dw.ip('192.168.1.2') & dw.ip('255.255.254.0'))`です。		

3. 例

- 非同期的に開始される手順によって生成されるメッセージをログへ書き込む:

```
local dw = require "drweb"

-- This function waits two seconds and returns a string,
-- received as an argument
function out_msg(message)
 dw.sleep(2)
 return message
end

-- "Main" function
function intercept(ctx)
 -- Output of a string at the NOTICE level to the Dr.Web for UNIX Mail
Servers log
 dw.notice("Intercept function started.")

 -- An asynchronous start of two copies of the out_msg function
 local f1 = dw.async(out_msg, "Hello,")
 local f2 = dw.async(out_msg, " world!")

 -- Waiting for the completion of the copies of the function
 -- out_msg and output its results to log
 -- the Dr.Web for UNIX Mail Servers log at the DEBUG level
 dw.log("debug", f1.wait() .. f2.wait())
end
```

- スケジュールされた手順を作成する:

```
local dw = require "drweb"

-- Save the table Future in the future global variable in order
-- to preven the removal by the garbage collector
future = dw.async(function()
   while true do
     -- Everyday, the following message is displayed in the log
     dw.sleep(60 * 60 * 24)
     dw.notice("A brand new day began")
   end
end)
```

- 文字列で表現されたIPアドレスをIpAddressテーブルに変更する:

```
local dw = require "drweb"

local ipv4 = dw.ip("127.0.0.1")
local ipv6 = dw.ip("::1")
local mapped = dw.ip("::ffff:127.0.0.1")
```

drweb.lookupモジュールの内容

1. 機能

このモジュールは次の機能を提供します。

- `lookup(<request>, <parameters>)`は、Dr.Web LookupDモジュールを介して利用できる外部スト

レージからデータを要求します。*<request>*引数は、Dr.Web LookupD設定内のセクション（文字列 *<type>@<tag>*）に対応している必要があります。*<parameters>*引数は任意で、リクエストを生成するために使用される置換を表します。以下の自動的に許可されるマーカーを使用できます。

- `$u`、`$U`は、クライアントコンポーネントによって送信されたユーザー名（`user`）に自動的に置き換えられます。
- `$d`、`$D`は、クライアントコンポーネントによって送信されたドメイン（`domain`）に自動的に置き換えられます。

これらの引数はテーブルとして設定されます。このテーブルのキーと値は文字列でなければなりません。この関数は、リクエストの結果である文字列の配列を返します。

- `check`（*<checked string>*、*<request>*、*<parameters>*）は、Dr.Web LookupDモジュールを介して利用できる外部リポジトリで *<checked string>*が見つかった場合に`true`を返します。引数 *<request>*および *<parameters>*は`lookup`関数の引数と同じです（上記を参照）。*<checked string>*引数は、文字列または`__tostring`メタメソッドを持つテーブル（つまり、文字列にフォーマットできる）であると想定されます。

2. 例

- `LookupD.LDAP.users`データソースから取得したユーザーのログリストへ書き込む:

```
local dw = require "drweb"
local dwl = require "drweb.lookup"

-- "Main" function
function intercept(ctx)
 -- Writing the string at the NOTICE level to the Dr.Web for UNIX Mail
Servers log
 dw.notice("Intercept function started.")

 -- Writing the request results to the Dr.Web for UNIX Mail Servers log
 -- to the 'ldap@users' data source
 for _, s in ipairs(dwl.lookup("ldap@users", {user="username"})) do
   dw.notice("Result for request to 'ldap@users': " .. s)
 end

end
```

Dr.Web ClamD

Dr.Web ClamDコンポーネントは、Sourcefire, Inc.のアンチウイルス製品Clam AntiVirus（ClamAV ®）のコアコンポーネントであるclamdアンチウイルスデーモンのインターフェースを使用してエミュレーションを実行します。このインターフェースにより、ClamAV ®とやり取りできる外部アプリケーションは、アンチウイルススキャンにDr.Web for UNIX Mail Serversを使用できます。

動作原理

このコンポーネントは、ローカルファイルシステムのファイルの内容と、ソケットを介して外部アプリケーションによって送信されたデータのストリームの両方をチェックします。このようなチェックは、外部アプリケーションのリクエストに応じてコンポーネントによって実行されます。さらに、コンポーネントは外部アプリケーションがソケットを介してオープンファイル記述子（ディスクリプタ）を渡したファイルの内容をチェックできます。

渡されたファイル記述子に基づくファイルスキャンは、記述子がローカルのUNIXソケットを介して渡された場合にのみ実行できます。

外部アプリケーションからローカルファイルシステム内のファイルへのパスが提供された場合、コンポーネントはスキャンタスクをDr.Web File Checkerファイルチェッカーコンポーネントに送信します。それ以外の場合、コンポーネントは、ソケット経由で受信したデータをDr.Web Network Checkerに送信します。

デフォルトでは、コンポーネントはDr.Web for UNIX Mail Serversの起動時に自動的に起動されません。コンポーネントの起動を有効にするには、開始パラメータに`Yes`の値を設定し、クライアントアプリケーション用に少なくとも1つの接続ポイントを定義する必要があります。その後、コンポーネントは外部アプリケーションからのファイルまたはデータストリームのスキャンリクエストの待機を開始します。コンポーネントの設定では、外部アプリケーション用に複数の接続ポイントを設定し、必要に応じて各ポイントに異なるスキャン設定を調整できます。

外部アプリケーションは、`clamd`との統合モジュールを装備している場合には、メールサーバー（PostfixやEximなど）として表すことができます。詳細は、外部アプリケーションとの統合のセクションを参照してください。

検出された脅威をDr.Web for UNIX Mail Serversで*駆除することはできません*。外部アプリケーションにはスキャンの結果のみが通知されます。そのため、検出された脅威はすべて外部アプリケーションで駆除する必要があります。

コマンドライン引数

Dr.Web ClamDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-clamd [<parameters>]
```

Dr.Web ClamDは次のパラメータを処理できます。

パラメータ	説明

`--help`	機能：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 短縮形：`-h` 引数：なし
`--version`	機能：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 短縮形：`-v` 引数：なし

例：

```
$ /opt/drweb.com/bin/drweb-clamd --help
```

このコマンドは、Dr.Web ClamDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます（原則として、OSの起動時）。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-clamd`コマンドを使用します。

設定パラメータ

このセクションの内容

- コンポーネントパラメータ
- コンポーネント構成の特別な側面

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[ClamD]`セクションで指定されている設定パラメータを使用します。

コンポーネントパラメータ

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。

パラメータ	説明
	デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-clamd • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-clamd • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-clamd
Start *{Boolean}*	このコンポーネントはDr.Web ConfigD設定デーモンによって起動される必要があります。 このパラメータにYes値を指定すると、設定デーモンはただちにコンポーネントを開始するように指示されます。また、No値を指定すると、設定デーモンはただちにコンポーネントを終了するように指示されます。 デフォルト値：No
Endpoint.*<tag>*.ClamdSocket *{IP address | UNIX socket}*	*<tag>*という名前の新しい接続ポイントを作成し、ファイルの脅威をスキャンする必要があるクライアントにソケット（IPv4アドレスまたはUNIXソケットのアドレス）を割り当てます。 1つの*<tag>*ポイントに指定できるソケットは1つのみです。 デフォルト値：未設定
[Endpoint.*<tag>*.]DetectSuspicious *{Boolean}*	ヒューリスティックアナライザによって検出された疑わしいファイルについて通知します。 Endpoint.*<tag>*のプレフィックスが指定されている場合は、パラメータの値が*<tag>*接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：Yes
[Endpoint.*<tag>*.]DetectAdware *{Boolean}*	アドウェアを含むファイルについて通知します。 Endpoint.*<tag>*のプレフィックスが指定されている場合は、パラメータの値が*<tag>*接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：Yes

パラメータ	説明
[Endpoint.<*tag*>.]DetectDialers *{Boolean}*	ダイアラーを含むファイルについて通知します。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：Yes
[Endpoint.<*tag*>.]DetectJokes *{Boolean}*	ジョークプログラムを含むファイルについて通知します。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：No
[Endpoint.<*tag*>.]DetectRiskware *{Boolean}*	リスクウェアを含むファイルについて通知します。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：No
[Endpoint.<*tag*>.]DetectHacktools *{Boolean}*	ハッキングツールを含むファイルについて通知します。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：No
[Endpoint.<*tag*>.]ReadTimeout *{time interval}*	クライアントからのデータを待つ最大タイムアウト Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：5s
[Endpoint.<*tag*>.]StreamMaxLength *{size}*	クライアントから受信できるデータの最大サイズ（スキャンするデータをバイトのストリームとして送信するため） Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の

パラメータ	説明
	値が指定されていないすべてのポイントに設定されます。 デフォルト値：25mb
[Endpoint.<*tag*>.]ScanTimeout *{time interval}*	クライアントから受信した1つのファイル（またはデータの一部）をスキャンする最大時間 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 指定可能な値：1秒（1s）から1時間（1h）まで。 デフォルト値：3m
[Endpoint.<*tag*>.]HeuristicAnalysis *{On / Off}*	スキャンのヒューリスティック解析を有効にします。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 デフォルト値：On
[Endpoint.<*tag*>.]PackerMaxLevel *{integer}*	圧縮されたオブジェクトのネスティングレベルの上限。圧縮されたオブジェクトは、特別なソフトウェア（UPX、PELock、PECompact、Petite、ASPack、Morphineなど）で圧縮された実行コードです。そのようなオブジェクトには、圧縮されたオブジェクトなどが含まれる他の圧縮されたオブジェクトが含まれる場合があります。最大ネスティングレベルを超えると、他の圧縮されたオブジェクト内の圧縮されたオブジェクトはスキャンされません。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
[Endpoint.<*tag*>.]ArchiveMaxLevel *{integer}*	スキャンできるアーカイブ（zip、rarなど）の最大ネスティングレベル。アーカイブには、他のアーカイブなどを含むアーカイブが含まれる場合があります。最大ネスティングレベルを超えると、アーカイブ内のアーカイブはスキャンされません。

パラメータ	説明
	Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
[Endpoint.<*tag*>.]MailMaxLevel *{integer}*	他のファイルが含まれる可能性のあるメーラーのファイル（pst、tbbなど）の最大ネスティングレベル（これらのファイルには他のファイルなどが含まれる場合もあります）。このパラメータの値は、ネスティングの上限を指定します。この上限を超えると、他のオブジェクト内のオブジェクトはスキャンされません。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
[Endpoint.<*tag*>.]ContainerMaxLevel *{integer}*	他のオブジェクトを含む他のタイプのオブジェクト（HTMLページやjarファイルなど）の最大ネスティングレベル。最大ネスティングレベルを超えると、オブジェクト内のオブジェクトはスキャンされません。 Endpoint.<*tag*>のプレフィックスが指定されている場合は、パラメータの値が<*tag*>接続ポイントにのみ設定されます。それ以外の場合は、このパラメータに別の値が指定されていないすべてのポイントに設定されます。 このパラメータの値は、0より大きい任意の整数に設定できます。値を0に設定すると、ネストされたオブジェクトはスキャンされません。 デフォルト値：8
[Endpoint.<*tag*>.] MaxCompressionRatio *{integer}*	圧縮されたオブジェクトの最大許容圧縮率（非圧縮サイズと圧縮サイズの比率）。オブジェクトの比率が制限を超えると、そのオブジェクトはスキャン中にスキップされます。 圧縮率には2よりも小さい値は指定できません。 デフォルト値：500

コンポーネント構成の特別な側面

オプションのEndpoint.<*tag*>プレフィックスが付いているパラメータはグループ化できます。各グループは、クライアントがコンポーネントへの接続に使用する固有の接続*ポイント*(*エンドポイント*)を定義し、固有の<*tag*> IDが割り当てられています。同じグループに属するすべてのスキャンパラメータは、対応する接続ポイントに接続されているクライアントに対してデータがスキャンされる場合にのみ適用可能な設定を定義します。パラメータがEndpoint.<*tag*>のプレフィックスなしで指定されている場合は、すべての接続ポイントの値が設定されます。接続ポイントからパラメータを削除した場合は、このパラメータをプログラムのハードコードされたデフォルト値に戻す代わりに、同じ名前の対応する「親」パラメータの現在の値を使用します(Endpoint.<*tag*>プレフィックスなしで設定)。

ClamdSocketパラメータは、リスニングソケットとこのソケットが対応するグループ(接続ポイント)の両方を定義するため、必ずEndpoint.<*tag*>プレフィックスを付けて指定する必要があります。

例

外部アプリケーション(サーバー)の2つのグループに対して2つの接続ポイントを設定する必要があるとします。グループをそれぞれ*servers1*と*servers2*と呼びます。また、*servers1*グループのサーバーはUNIXソケットを介して接続できるのに対し、*servers2*グループのサーバーはネットワーク接続を介して接続できます。さらに、ヒューリスティック解析はデフォルトで無効にする必要がありますが、*servers2*グループのサーバーには使用する必要があるとします。これを設定する方法を次の例で説明します。

1) 設定ファイルを次のように編集します。

```
[ClamD]
HeuristicAnalysis = Off

[ClamD.Endpoint.servers1]
ClamdSocket = /tmp/srv1.socket

[ClamD.Endpoint.servers2]
ClamdSocket = 127.0.0.1:1234
HeuristicAnalysis = On
```

2) コマンドラインベース管理ツールDr.Web Ctlでは次のように設定します。

```
# drweb-ctl cfset ClamD.HeuristicAnalysis Off
# drweb-ctl cfset ClamD.Endpoint -a servers1
# drweb-ctl cfset ClamD.Endpoint -a servers2
# drweb-ctl cfset ClamD.Endpoint.servers1.ClamdSocket /tmp/srv1.socket
# drweb-ctl cfset ClamD.Endpoint.servers2.ClamdSocket 127.0.0.1:1234
# drweb-ctl cfset ClamD.Endpoint.servers2.HeuristicAnalysis On
```

どちらの方法でも効果は同じですが、設定ファイルを編集する場合は、drweb-configdコンポーネントにSIGHUP信号を送信して変更した設定を適用する必要があります(これを行うにはdrweb-ctl reloadコマンドを発行します)。

外部アプリケーションとの統合

clamdインターフェースのエミュレーションにより、clamdアンチウイルスデーモン（ClamAVに含まれます）に接続可能な外部アプリケーションとDr.Web ClamDを統合できます。

以下の表は、アンチウイルススキャンにclamdを使用できるアプリケーションの例を示しています。

製品	統合
メールメッセージサービス	
メールサーバー Postfix	**clamdの使い方** メールメッセージのウイルスや悪意のあるプログラムのスキャン。 **統合要件** 中間コンポーネント（clamsmtpd、clamav-milter、またはamavisd-new）を使用する。 **ドキュメントへのリンク** Postfixのドキュメント：http://www.postfix.org/documentation.html amavisd-newの説明とソースコードファイル：https://www.amavis.org/
メールサーバー Exim	**clamdの使い方** メールメッセージのウイルスや悪意のあるプログラムのスキャン。 **統合要件** Exim設定ファイルに次の設定を追加する。 `av_scanner = clamd:` *<path_to_clamd_UNIX_socket>* *<path_to_clamd_UNIX_socket>*は、Dr.Web ClamDで設定された接続ポイントのソケット（*エンドポイント*）に対応します。 **ドキュメントへのリンク** Eximのドキュメント：https://exim.org/docs.html
メールサーバー CommuniGate Pro	**clamdの使い方** メールメッセージのウイルスや悪意のあるプログラムのスキャン。 **統合要件** 中間コンポーネントとしてcgpavを使用する。 **ドキュメントへのリンク** CommuniGate Pro公式サイト：https://www.communigate.world/ cgpavの説明とソースコードファイル：http://program.farit.ru/index.html

clamdアンチウイルスデーモンと同様に、Dr.Web ClamDと直接通信する外部ソフトウェアコンポーネントの設定で、clamdに接続するためのアドレスをUNIXソケットへのパスとして、または設定された接続ポイント（*エンドポイント*）の1つでDr.Web ClamDにリッスンされるTCPソケットとして指定します。

CommuniGate ProのDr.Web ClamDへの接続例：

1. cgpav（バージョン1.5）のダウンロードと構築：

```
$ wget http://program.farit.ru/antivir/cgpav-1.5.tar.gz
$ tar -xzvf cgpav-1.5.tar.gz
$ cd cgpav-1.5/
$ ./configure
$ make && make install
```

設定段階で、`Choose Anti-Virus daemon`へのレスポンスを選択するときは`Clamav`を選択します。

2. Dr.Web ClamDの設定：

```
[ClamD]
Start = yes

[ClamD.Endpoint.mail]
ClamdSocket = /var/run/drweb.clamd
```

3. CommuniGate Proの設定：
   1) CommuniGate Pro設定ファイル（`/var/CommuniGate/Settings/cgpav.conf`）で、Dr.Web ClamDソケットへのパスを指定します。

```
clamd_socket = /var/run/drweb.clamd
```

   2) CommuniGate ProのWebインターフェースで以下を設定します。
      - **Setting** → **General** → **Helpers**に移動します。
        - **Content Filtering**で新しいフィルターを設定します。
          - 新しいフィルターを**Enabled**に切り替えます。
          - フィルター名（`drweb`など）を指定します。
          - **Program Path**パラメータで`cgpav`を指定します。
        - 変更内容を保存します。
      - **Setting** → **Mail** → **Rules**に移動します。
        - 新しいルール名（`drweb_scan`など）を指定し、**Add Rule**をクリックします。
        - **Highest**ルール設定を選択し、変更内容を保存します。
        - ルール名の右側にある**Edit**をクリックします。
          - **Data**ドロップダウンメニューで**Message Size**を選択します。
          - **Operation**フィールドで**greater than**を選択します。
          - **Parameter**フィールドで`1`を指定します。
          - **Action**フィールドで**ExternalFilter**を選択します。
          - **Parameter**で、以前に作成したフィルター（この場合は`drweb`）の名前を選択します。
        - 変更内容を保存します。

Dr.Web File Checker

ファイルスキャンコンポーネントDr.Web File Checkerはファイルシステムのファイルとディレクトリをスキャンするように設計されており、Dr.Web for UNIX Mail Serversの他のコンポーネントによってファイルシステムオブジェクトをスキャンするために使用されます。さらにこのコンポーネントは、隔離されたファイルが保存されているディレクトリの内容を管理するため、隔離マネージャーとしても機能します。

動作原理

このコンポーネントは、すべてのファイルシステムオブジェクト（ファイル、ディレクトリ、ブートレコード）へのアクセスに使用されます。スーパーユーザー（*root*）権限で起動します。

スキャン済みのすべてのファイルとディレクトリにインデックスを付け、特別なキャッシュに確認されたオブジェクトに関するすべてのデータを保存します。スキャン済みのオブジェクトやスキャン後変更されていないオブジェクトを繰り返しスキャンしないようにします（このようなオブジェクトに対するスキャンリクエストを受信した際は、キャッシュから取得した以前のスキャン結果が返されます）。

ファイルシステムオブジェクトを確認するリクエストがDr.Web for UNIX Mail Serversのコンポーネントから受信されると、このオブジェクトにスキャンが必要かどうかを確認します。必要な場合は、Dr.Web Scanning Engineに対してスキャンタスクが生成されます。スキャンされたオブジェクトに脅威が含まれている場合、Dr.Web File Checkerはその脅威を検出された脅威レジストリに入れ、脅威への対応としてスキャンを開始したクライアントコンポーネントによって指定されている場合は駆除（修復、削除、または隔離）します。スキャンは、製品のさまざまなコンポーネントによって開始できます。

スキャン中、ファイルチェックコンポーネントでは、スキャン結果と適用されたアクションがあればそれを詳述するレポートを生成してクライアントコンポーネントに送信します。

標準のスキャン方法とは別に、内部使用には以下の特別な方法があります。

- *「flow」スキャン方法*。このスキャン方法を使用するクライアントコンポーネントは、検出と駆除のパラメータを一度だけ初期化します。これらのパラメータは、このクライアントコンポーネントからのファイルをスキャンするための今後のすべてのリクエストに適用されます。
- *「プロキシ」スキャン方法*。この方法を使用する場合、ファイルチェックコンポーネントは、検出された脅威にアクションを適用せずに、また今後のアクションを許可するために検出された脅威に関するレコードを保存せずに、ファイルをスキャンします。スキャン処理を開始したコンポーネントによって必要なアクションが適用される必要があります。この方法はDr.Web ClamDコンポーネントによって使用されます。

Dr.Web Ctlユーティリティの`flowscan`コマンド（`drweb-ctl`コマンドで起動）を使用し、*「フロー」*スキャン方法でファイルをスキャンできます。ただし、通常のオンデマンドスキャンでは`scan`コマンドを使用することを推奨します。

作業中、ファイルスキャンコンポーネントは脅威のレジストリを保持して隔離を管理するだけでなく、ファイルスキャン全体の統計情報も収集し、最後の1分間、5分間、15分間の、1秒間に確認された平均ファイル数を計算します。

コマンドライン引数

Dr.Web File Checkerを起動するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-filecheck [<parameters>]
```

Dr.Web File Checkerは次のパラメータを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-filecheck --help
```

このコマンドは、Dr.Web File Checkerに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。Dr.Web for UNIX Mail Servers の他のコンポーネントからファイルシステムスキャンのリクエストを受信するときに、Dr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理し、必要に応じてファイルをスキャンするには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（`drweb-ctl`コマンドを使用して起動されます）。

Dr.Web File Checkerを使用して任意のファイルまたはディレクトリをスキャンするには、Dr.Web Ctlの`scan`コマンドを使用します。

```
$ drweb-ctl scan <path to file or directory>
```

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-filecheck`コマンドを使用します。

設定パラメータ

コンポーネントは、Dr.Web for UNIX Mail Servers の統合された設定ファイルの`[FileCheck]`セクションで指定されている設定パラメータを使用します。

このセクションは以下のパラメータを保存します。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-filecheck • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-filecheck • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-filecheck
DebugClientIpc *{Boolean}*	詳細なIPCメッセージをデバッグレベルでログファイルに含めるかどうかを示します（LogLevel = DEBUGの場合など）。 デフォルト値：No
DebugScan *{Boolean}*	ファイルスキャン中に受信した詳細メッセージをデバッグレベルでログファイルに書き込みます（LogLevel = DEBUGの場合など）。 デフォルト値：No
DebugFlowScan *{Boolean}*	*「フロー」*メソッドによるファイルスキャンに関する詳細メッセージをデバッグレベルでログファイルに書き込みます（LogLevel = DEBUGの場合など）。 デフォルト値：No
DebugProxyScan *{Boolean}*	*「プロキシ」*メソッドによるファイルスキャンに関する詳細メッセージをデバッグレベルでログファイルに書き込みます（LogLevel = DEBUGの場合など）。通常、このスキャン方法はDr.Web ClamDコンポーネントによって使用されます。 デフォルト値：No
DebugCache *{Boolean}*	スキャンのキャッシュされた結果に関する詳細メッセージをデバッグレベルでログファイルに書き込みます（LogLevel = DEBUGの場合など）。 デフォルト値：No
MaxCacheSize *{size}*	スキャンしたファイルに関するデータを保存するためのキャッシュの最大許容サイズ。 0を指定した場合、キャッシュは無効になります。 デフォルト値：50mb
RescanInterval *{time interval}*	前回のスキャンの結果がキャッシュ内にある場合にファイルが再スキャンされない期間（保存された情報が最新のものと見なされる期間）。

パラメータ	説明
	指定可能な値：0秒（0s）から1分（1m）まで。 設定された間隔が1s未満の場合は遅延がないため、ファイルはリクエストに応じてスキャンされます。 デフォルト値：1s
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s

Dr.Web Network Checker

ネットワークチェッカーエージェントDr.Web Network Checkerでは、ネットワーク経由で受信したデータをスキャンエンジンでスキャンできる他、分散ファイルでの脅威のスキャンを実行できます。このコンポーネントを使用すると、ネットワークホストを介してデータ(ファイルの内容など)を送受信できるようDr.Web for UNIX Mail Serversがインストールされているネットワークホスト間の接続を調整し、スキャンを実行できます。このコンポーネントは、設定されているすべての利用可能なネットワークホストへのスキャンタスクの自動分配を(ネットワークを介して送受信することにより)調整します。このコンポーネントは、スキャンタスクによって生じる負荷をホスト間に分散させます。リモートホストとの接続が設定されていない場合、コンポーネントはすべてのデータをローカルのDr.Web Scanning Engineにのみ送信します。

このコンポーネントは、ネットワーク接続を介して受信したデータのスキャンに常に使用されます。したがって、コンポーネントが見つからないか使用できない場合、スキャン用にネットワーク接続を介してデータを送信するコンポーネント(Dr.Web ClamD、SpIDer Gate、Dr.Web MailD)のパフォーマンスは不正確になります。

このコンポーネントは、Dr.Web File Checkerコンポーネントを置き換えることができないため、ローカルファイルシステムにあるファイルの分散スキャンを整理するようには作られていません。ローカルファイルの分散スキャンを整理するには、Dr.Web MeshDコンポーネントを使用します。

ネットワークを介して転送されるデータのスキャンの負荷が高まると、利用できるファイル記述子の数が減少するため、スキャンに問題が生じる場合があります。この場合、Dr.Web for UNIX Mail Serversに利用できるファイル記述子の制限数を増やす必要があります。

スキャン時には、SSL/TLSを適用することにより、オープンチャネル経由または保護されたチャネル経由のいずれかでデータを共有できます。安全なHTTPS接続を使用するには、ファイルを共有するホストに適切なSSLサーバー証明書とプライベートキーを提供する必要があります。SSLキーと証明書を生成するには、`openssl`ユーティリティを使用できます。`openssl`ユーティリティを使用して証明書とプライベートキーを生成する方法の例については、付録E. SSL証明書を生成するのセクションを参照してください。

動作原理

このコンポーネントにより、スキャンでローカルファイルシステムのファイルとして表されていないデータをDr.Web Scanning Engineエンジン(ローカルまたはリモートホストにある)に送信できます。このデータは、(Dr.Web MailD、Dr.Web ClamD)接続を介してスキャンのデータを送信するコンポーネントによって処理されます。これらのコンポーネントは、ローカルホストにある場合でも、Dr.Web Scanning Engineエンジンへのファイル転送に常にDr.Web Network Checkerを使用することに注意してください。そのため、Dr.Web Network Checkerが利用できない場合、*これらのコンポーネントは正しく機能しません*。

また、Dr.Web Network Checkerにより、Dr.Web for UNIX Mail Serversがインストールされているネットワーク(またはその他のDr.Web for UNIXソリューション10.1以降)の特定のノードセットとDr.Web for UNIX Mail Serversを接続することで、ローカルファイルシステムでファイルとして表されない分散されたチェックデータの有無をまとめて確認できます。これによりこのコンポーネントは、検証でデータを交換する一連のネットワークノードである*スキャンクラスタ*を構築および設定できます(各インスタンスにDr.Web Network Checker分散検証エージェントの独自のインスタンスが必要です)。スキャンクラスタに含まれるネットワークの各ノードで、Dr.Web Network Checkerはスキャンデータのタスクの自動分散を実行し、接続が設定されているすべての使用可能なノードにネットワークを介して転送します。同時に、リモートノードで使用可能なリソースの量に応じて、データ検証によって発生するノードの負荷分散が実行されます(読み込みで使用可能なリソース量の指標として、このノードのス

キャンコアDr.Web Scanning Engineによって生成される子のスキャンプロセス数が使用されます）。使用されている各ノードでのチェックを待機しているファイルキューの長さも推定されます。

この場合、スキャンクラスタに含まれる任意のネットワークノードは、リモートスキャンにデータを送信するスキャンクライアントの他、検証のために指定されたネットワークノードからデータを受信するスキャンサーバーとして機能します。必要に応じて、ノードがスキャンサーバーまたはスキャンクライアントとしてのみ機能するように分散スキャンエージェントを設定できます。

スキャン用にネットワーク経由で受信したデータは、一時ファイルとしてローカルファイルシステムに保存され、Dr.Web Scanning Engineエンジンに送信されるか、利用できない場合はスキャンクラスタの他のノードに送信されます。

設定にある`InternalOnly`パラメータを使用すると、Dr.Web Network Checkerの動作モードを管理できます。これは、Dr.Web Network CheckerがDr.Web for UNIX Mail Serversをスキャンクラスタに含めるために使用されるのか、Dr.Web for UNIX Mail Serversローカルコンポーネントのみの内部目的のために使用されるのかを示します。

> ファイルのチェック（スキャンクラスタのノードに対するスキャンジョブの分配を含む）にDr.Web Network Checkerを使用する独自のコンポーネント（外部アプリケーション）を作成できます。そのためにDr.Web Network Checkerコンポーネントには、Google Protobufテクノロジーに基づくカスタムAPIが用意されています。Dr.Web Network Checker APIの他、Dr.Web Network Checkerを使用するクライアントアプリケーションのサンプルコードが`drweb-netcheck`パッケージの一部として提供されています。

スキャンクラスタの作成例は、スキャンクラスタの作成セクションにあります。

コマンドライン引数

Dr.Web Network Checkerを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-netcheck [<parameters>]
```

Dr.Web Network Checkerは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-netcheck --help
```

このコマンドは、Dr.Web Network Checkerに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接実行することはできません。必要に応じて、Dr.Web ConfigD構成デーモンによって自動的に実行されます（通常はOSの起動時）。コンポーネント設定で、`FixedSocket`パラメータの値が指定されており、`InternalOnly`パラメータが`No`に設定されている場合、エージェントは常に実行されており、指定されたUNIXソケットを介してクライアントに使用可能です。ネットワーク経由でスキャンを開始するには、Dr.Web for UNIX Mail Servers管理用のDr.Web Ctlコマンドラインツールを使用できます（`drweb-ctl`コマンドで始まるもの）。リモートホストへの接続が設定されていない場合は、ローカルスキャンが開始されます。

Dr.Web Network Checkerを使用して任意のファイルまたはディレクトリをスキャンするには、Dr.Web Ctlツールの`netscan`コマンドを使用します。

```
$ drweb-ctl netscan <path to file or directory>
```

コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-netcheck`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの[NetCheck]セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-netcheck • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-netcheck • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-netcheck
FixedSocket *{path to file / address}*	固定されたDr.Web Network Checkerエージェントインスタンスのソケット。 このパラメータが指定されている場合、Dr.Web ConfigD設定デーモンは、このソケットを介してクライアントが使用可能な分散スキャンエージェントの実行中のコンポーネントのコピーが、常に存在することを確認します。 使用可能な値： • *<path to file>*はローカルのUNIXソケットへのパスです。 • *<address>*は、ペア*<IP address>*:*<port>*で示されるネットワークソケットです。 デフォルト値：*（未設定）*
InternalOnly *{Boolean}*	コンポーネントの動作モードの管理。 値がYesに設定されている場合、LoadBalance*設定やFixedSocketパラメータの値に関係なく、コンポーネントはDr.Web for UNIX Mail Serversコンポーネントの内部目的にのみ使用され、スキャンクラスタへのDr.Web for UNIX Mail Serversの組み込みや、外部（Dr.Web for UNIX Mail Serversに対して）クライアントアプリケーションの処理には使用されません。 デフォルト値：No
RunAsUser *{UID / user name}*	その権限によりコンポーネントを実行するユーザーの名前。このユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。

パラメータ	説明
	ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「name:」というプレフィックスを付けて指定します。たとえば、RunAsUser = name:123456です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：drweb
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された値を超えると、コンポーネントはシャットダウンします。 LoadBalanceAllowFromパラメータまたはFixedSocketパラメータが設定されている場合、この設定は無視されます（指定した時間を経過しても、コンポーネントは動作を終了しません）。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s
LoadBalanceUseSsl *{Boolean}*	他のホストへの接続にSSL/TLSを使用します。 使用可能な値： • Yes - SSL/TLSを使用します。 • No - SSL/TLSを使用しません。 このパラメータがYes値に設定されている場合、証明書と対応するプライベートキーを、このホストとこのホストがやり取りするホストに対して指定する必要があります（パラメータLoadBalanceSslCertificateおよびLoadBalanceSslKey）。 デフォルト値：No
LoadBalanceSslCertificate *{path to file}*	安全なSSL/TLS接続を介して他のホストと通信するためにDr.Web Network Checkerによって使用されるSSL証明書へのパス。 証明書ファイルとプライベートキーファイル（後述のパラメータで指定されます）は、一致するペアを形成する必要があります。 デフォルト値：*（未設定）*
LoadBalanceSslKey *{path to file}*	安全なSSL/TLS接続を介して他のホストと通信するためにDr.Web Network Checkerによって使用されるプライベートキーへのパス。 証明書ファイルとプライベートキーファイル（上記のパラメータで指定されます）は、一致するペアを形成する必要があります。 デフォルト値：*（未設定）*
LoadBalanceSslCa *{path}*	信頼できるルート証明書のリストがあるディレクトリまたはファイルへのパス。これらの証明書には、SSL/TLSプロトコルを介してデータを交換するときにスキャンクラスタ内のエージェントが使用する、証明書の信頼性を証明する証明書が必要です。

パラメータ	説明
	パラメータ値が空の場合、このホストで動作しているDr.Web Network Checkerは、やり取りをするエージェントの証明書を認証しません。ただし設定によっては、ホスト上で動作しているエージェントが使用する証明書を、これらのエージェントが認証できます。 デフォルト値：*（未設定）*
`LoadBalanceSslCrl` *{path}*	失効した証明書のシステムリストを含むディレクトリまたはファイルへのパス。 パラメータ値が指定されていない場合、このホストで実行されているDr.Web Network Checkerは、やり取りするエージェントの証明書の有効性をチェックしません。ただし設定によっては、このホストで実行されているエージェントが使用する証明書の有効性をチェックする場合があります。 デフォルト値：*（未設定）*
`LoadBalanceServerSocket` *{address}*	リモートホストからスキャン用に送信されたファイルを受信するために、Dr.Web Network Checkerがこのホスト上でリッスンしているネットワークソケット（IPアドレスとポート）（スキャンサーバーとして動作できる場合）。 デフォルト値：*（未設定）*
`LoadBalanceAllowFrom` *{IP address}*	Dr.Web Network Checkerが（スキャンサーバーとして）スキャン用のファイルを受信するリモートネットワークホストのIPアドレス。 リストをパラメータ値として指定できます。リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：ホストアドレス192.168.0.1と10.20.30.45のリストに追加します。 1. 設定ファイルに値を追加します。 • 1行に2つの値： `[NetCheck]` ` LoadBalanceAllowFrom = "192.168.0.1", "10.20.30.45"` • 2行（1行に1つの値）： `[NetCheck]` ` LoadBalanceAllowFrom = 192.168.0.1` ` LoadBalanceAllowFrom = 10.20.30.45`

パラメータ	説明
	2. drweb-ctl cfset コマンドを使用して値を追加します。 `# drweb-ctl cfset NetCheck.LoadBalanceAllowFrom -a 192.168.0.1` `# drweb-ctl cfset NetCheck.LoadBalanceAllowFrom -a 10.20.30.45` このパラメータが空の場合、削除されたファイルをスキャンのために受信することはできません(ホストはスキャンサーバーとして動作しません)。 デフォルト値: *(未設定)*
LoadBalanceSourceAddress *{IP address}*	リモートスキャン用にファイルを転送するために、ホスト上のDr.Web Network Checkerによって使用されるネットワークインターフェースのIPアドレス(ホストがスキャンサーバーとして動作し、複数のネットワークインターフェースを持つ場合)。 空の値を指定すると、OSカーネルによって自動的に選択されたネットワークインターフェースが使用されます。 デフォルト値: *(未設定)*
LoadBalanceTo *{address}*	ホスト上のDr.Web Network Checkerが(ネットワークスキャンクライアントとして)リモートスキャン用のファイルを送信できるリモートホストのソケット(IPアドレスまたはポート)。 リストをパラメータ値として指定できます。リストの値は、コンマ(引用符内の各値)で区切る必要があります。パラメータはセクションで複数回指定できます(この場合、そのすべての値が1つのリストにまとめられます)。 例: ソケット192.168.0.1:1234および10.20.30.45:5678をリストに追加します。 1. 設定ファイルに値を追加します。 • 1つの文字列に2つの値: `[NetCheck]` `LoadBalanceTo = "192.168.0.1:1234", "10.20.30.45:5678"` • 2つの文字列(文字列ごとに1つの値): `[NetCheck]` `LoadBalanceTo = 192.168.0.1:1234` `LoadBalanceTo = 10.20.30.45:5678`

パラメータ	説明
	2. `drweb-ctl cfset` コマンドを使用して値を追加します。 `# drweb-ctl cfset NetCheck.LoadBalanceTo -a 192.168.0.1:1234` `# drweb-ctl cfset NetCheck.LoadBalanceTo -a 10.20.30.45:5678` パラメータ値が空の場合、ローカルファイルをリモートスキャン用に転送できません（ホストはネットワークスキャンクライアントとして動作しません）。 デフォルト値：*（未設定）*
`LoadBalanceStatusInterval` *{time interval}*	ワークロードに関する情報を含む次のメッセージをすべてのスキャンクライアントに送信する際にホストによって考慮されるタイムインターバル（`LoadBalanceAllowFrom` パラメータで指定）。 デフォルト値：`1s`
`SpoolDir` *{path to directory}*	スキャン用にネットワーク経由で送信され、Dr.Web Network Checker によって受信されたファイルの保存に使用されるローカルファイルシステムのディレクトリ。 デフォルト値：`/tmp/netcheck`
`LocalScanPreference` *{fractional number}*	ファイル（ローカルファイルまたはネットワーク経由で受信したファイル）をスキャンするスキャンサーバーが選択されたときに考慮される、ホストの相対的な重み（プライオリティー）。ローカル端末の相対的な重みが、スキャンサーバーとして利用可能なすべてのホストの重みよりも大きい場合、ファイルはローカルでスキャンされます。 最小値：`1` デフォルト値：`1`

スキャンクラスタの作成

このセクションの内容：

- 注意事項
- スキャンクラスタの作成例
- クラスタノードの設定
- クラスタの動作確認

注意事項

ファイルや他のオブジェクトのスキャン中に分散チェックを実行できるスキャンクラスタを作成するには、各ノードに Dr.Web Network Checker コンポーネントがインストールされた一連のネットワークノードが必要です。クラスタノードにスキャン対象のデータの送受信以外の機能を持たせるには、ノードにスキャンエンジン Dr.Web Scanning Engine もインストールする必要があります。そのため、スキャンクラスタのノードを作成するには、次のコンポーネントの最小セット（最低限のもの）をサーバーにインストールする必要があります（ここにリストされているコンポーネン

トの機能を保証するために自動的にインストールされるDr.Web for UNIX Mail Serversの他のコンポーネントは、スキップされます）。

1. Dr.Web Network Checker（`drweb-netcheck`パッケージ）は、ノード間のネットワークを提供するコンポーネントです。
2. Dr.Web Scanning Engine（`drweb-se`パッケージ）は、ネットワーク経由で受信したデータをスキャンするために必要なスキャンエンジンです。場合によっては、このコンポーネントが存在しません。その場合、ノードは他のスキャンクラスタノードに対してチェック対象のデータの送信のみを行います。

*ピアツーピア*ネットワークのスキャンクラスタを構成するノード、つまり各ノードは、このノードのDr.Web Network Checkerコンポーネントで定義されている設定に応じて、*スキャンクライアント*（他のノードにスキャン用にデータを送信）または*スキャンサーバー*（他のノードからスキャン用にデータを受信）として機能できます。適切に設定すれば、クラスタノードは同時にスキャンクライアントとスキャンサーバーの両方として機能します。

スキャンクラスタ設定に関連するDr.Web Network Checkerのパラメータは、`LoadBalance`で始まる名前を持ちます。

スキャンクラスタの作成例

下の図に表示されているスキャンクラスタの構成例を確認してください。

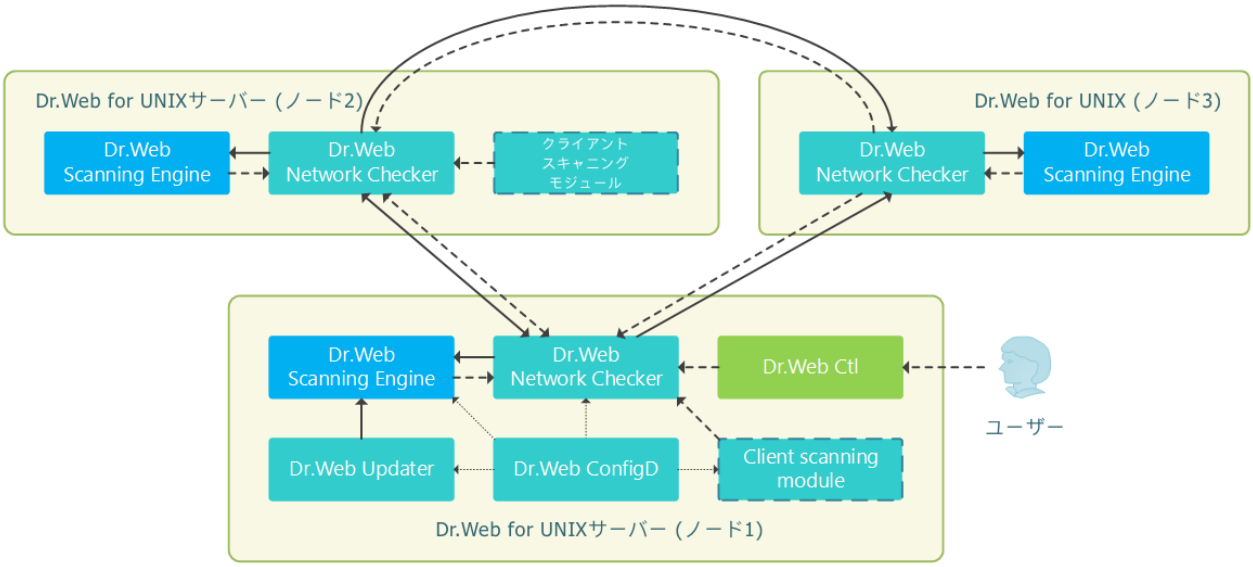

図14. スキャニングクラスタの構造

ここでは、クラスタは3つのノード（図では*ノード1*、*ノード2*、*ノード3*として表示）で構成されると想定します。この場合、ノード1とノード2は、UNIXサーバー用のDr.Webの上位製品がインストールされているサーバーです（Dr.Web for UNIX file serversまたはDr.Web for UNIXインターネットゲートウェイなど。製品タイプは不問）。ノード3は、ノード1とノード2から転送されたファイルのスキャンをサポートするためにのみ使用されます。したがって、必要最小限のコンポーネントセット（Dr.Web Network CheckerおよびDr.Web Scanning Engine）のみがインストールされます。ノードの操作性を確保するために自動的にインストールされる他のコンポーネント（Dr.Web ConfigDなど）は図に記載されていません。ノード1と2は、相互間でサーバーおよびスキャンクライアントの両方として機能し（スキャンに関連する負荷の相互分散を実行）、ノード3はサーバーとしてのみ機能し、ノード1と2からタスクを受信します。

これらのコンポーネントは、ローカルにインストールされたスキャンエンジンDr.Web Scanning Engineとクラスタパートナーノードの間で分散され、負荷分散に応じてスキャンサーバーとして機能します。

ローカルファイルシステムでファイルとして表されていないデータをスキャンするコンポーネントのみが、検証のクライアントモジュールとして機能することに注意してください。これは、SpIDer Guard ファイルシステムモニターおよびDr.Web File Checkerコンポーネントによるファイルの分散スキャンに使用できるスキャンクラスタを指します。

クラスタノードの設定

指定したクラスタ構成をカスタマイズするには、すべてのクラスタノードでDr.Web Network Checkerの設定を変更する必要があります。以下の設定はすべて.iniファイルで指定されます（設定ファイルのフォーマットの説明を参照）。

ノード1

```
[NetCheck]
InternalOnly=No
LoadBalanceUseSsl = No
LoadBalanceServerSocket = <Node 1 IP address>:<Node 1 port>
LoadBalanceAllowFrom = <Node 2 IP address>
LoadBalanceSourceAddress = <Node 1 IP address>
LoadBalanceTo = <Node 2 IP address>:<Node 2 port>
LoadBalanceTo = <Node 3 IP address>:<Node 3 port>
```

ノード2

```
[NetCheck]
InternalOnly=No
LoadBalanceUseSsl = No
LoadBalanceServerSocket = <Node 2 IP address>:<Node 2 port>
LoadBalanceAllowFrom = <Node 1 IP address>
LoadBalanceSourceAddress = <Node 2 IP address>
LoadBalanceTo = <Node 1 IP address>:<Node 1 port>
LoadBalanceTo = <Node 3 IP address>:<Node 3 port>
```

ノード3

```
[NetCheck]
InternalOnly=No
LoadBalanceUseSsl = No
LoadBalanceServerSocket = <Node 3 IP address>:<Node 3 port>
LoadBalanceAllowFrom = <Node 1 IP address>
LoadBalanceAllowFrom = <Node 2 IP address>
```

注意：

- 他の（ここに記載されていない）Dr.Web Network Checkerパラメータは変更されません。
- IPアドレスとポート番号は実際のものに変更する必要があります。
- この例では、ノード間のデータ交換におけるSSLの使用は無効になっています。SSLを使用する必要がある場合は、`LoadBalanceUseSsl`パラメータに値`Yes`を設定するとともに、パラメータ

LoadBalanceSslCertificate、LoadBalanceSslKey、LoadBalanceSslCaに必要な値を設定する必要があります。

クラスタの動作確認

データディストリビューションモードでのクラスタの動作を確認するには、ノード1とノード2で次のコマンドを使用します。

```
$ drweb-ctl netscan <path to file or directory>
```

指定されたコマンドを実行するときは、指定されたディレクトリのファイルをDr.Web Network Checkerでチェックする必要があります。これにより、カスタマイズされたクラスタノードにチェックが分散されます。スキャン前に各ノードのネットワークチェックの統計を表示するには、次のコマンドを使用してDr.Web Network Checkerの統計の表示を実行します（統計の表示を中断するにはCTRL+Cを押します）。

```
$ drweb-ctl stat -n
```

Dr.Web Scanning Engine

Dr.Web Scanning Engineスキャンエンジンでは、ディスクデバイスのファイルやブートレコード(*MBR - マスターブートレコード*、*VBR - ボリュームブートレコード*)に含まれている、ウイルスなどの悪意のあるオブジェクトを検索できます。このコンポーネントはスキャンエンジンDr.Web Virus-Finding Engineをメモリに読み込んで起動する他、このエンジンが脅威の検出のために使用するDr.Webウイルスデータベースの読み込みも行います。

このスキャンエンジンは、Dr.Web for UNIX Mail Serversの他のコンポーネント(Dr.Web File CheckerとDr.Web Network Checker、部分的にはDr.Web MeshD)からスキャンリクエストを受信するサービスとして、デーモンモードで動作します。Dr.Web Scanning EngineとDr.Web Virus-Finding Engineが存在しないか使用できない場合、このノードではアンチウイルススキャンは実行されません(ただし、Dr.Web MeshDコンポーネントを含むDr.Web for UNIX Mail Serversを除きます。このコンポーネントの設定には、スキャンエンジンサービスを提供するローカルクラウドノードへの接続が含まれています)。

動作原理

このコンポーネントは、埋め込まれた脅威についてファイルシステムオブジェクト(ファイルおよびブートディスクレコード)をスキャンするリクエストを、Dr.Web for UNIX Mail Serversのコンポーネントから受信するサービスとして動作します。また、スキャンタスクをキューに入れ、リクエストされたオブジェクトをDr.Web Virus-Finding Engineスキャンエンジンを使用してスキャンします。脅威が検出され、スキャンタスクによって脅威を修復するように指示があった場合、スキャンされたオブジェクトにこのアクションを適用できるのであれば、スキャンエンジンは修復を試みます。

スキャンエンジン、Dr.Web Virus-Finding Engineスキャンエンジン、およびウイルスデータベースは1つの単位を構成しており、分離することはできません。スキャンエンジンはウイルスデータベースをダウンロードし、クロスプラットフォームのスキャンエンジンDr.Web Virus-Finding Engineの動作環境を提供します。ウイルスデータベースとスキャンエンジンは、Dr.Web for UNIX Mail Serversに含まれているDr.Web Updater更新コンポーネントによって更新されますが、このコンポーネントはスキャンエンジンの一部ではありません。対応するコマンドがユーザーから送信された場合、更新コンポーネントはDr.Web ConfigD設定デーモンによって定期的または強制的に実行されます。さらに、Dr.Web for UNIX Mail Serversが集中管理モードで動作していている場合、ウイルスデータベースとスキャンエンジンの更新はDr.Web ES Agentによって実行されます。後者のコンポーネントは集中保護サーバーとやり取りし、更新を受け取ります。

Dr.Web Scanning Engineは設定デーモンDr.Web ConfigDの管理下でも、自律モードでも動作できます。前者の場合、デーモンはエンジンを実行し、アンチウイルスデータベースが最新であることを確認します。後者の場合、エンジンの起動とウイルスデータベースの更新は、このエンジンを使用する外部アプリケーションによって実行されます。ファイルのスキャンを要求するスキャンエンジンにリクエストを発行するDr.Web for UNIX Mail Serversのコンポーネントは、他の外部アプリケーションと同じインターフェースを使用します。

ユーザーは、ファイルのチェックのためにDr.Web Scanning Engineを使用する独自のコンポーネント(外部アプリケーション)を作成できます。このため、Dr.Web Scanning EngineにはGoogle Protobufに基づく特別なAPIが含まれています。Dr.Web Scanning Engine APIガイドと、Dr.Web Scanning Engineを使用したクライアントアプリケーションの例を入手するには、Doctor Webパートナーケア部門(https://partners.drweb.com/)にお問い合わせください。

受信したタスクは、優先度(高い、通常、低い)ごとのキューに自動的に分配されます。キューの選択は、タスクを作成したコンポーネントによって異なります。たとえば、ファイルシステムモニターによって作成されたタスクは、応答時間がモニターにとって重要であるため、優先順位が高くなります。スキャンエンジンは、スキャン用に受信したすべてのタスクの数やキューの長さなど、操作に関わる統計を計算します。平均負荷率として、スキャンエンジン

は1秒あたりのキューの平均長を使用します。このレートは、直近1分間、直近5分間、直近15分間の平均です。

Dr.Web Virus-Finding Engineスキャンエンジンは、機械語の命令やその他の実行可能コードの属性に基づいて潜在的に危険なオブジェクトを検出するために設計された、シグネチャ解析（シグネチャベースの脅威の検出）やその他のヒューリスティック解析および動作解析の方法をサポートしています。

ヒューリスティック解析は信頼性の高い結果を保証できず、次のようなエラーが発生する可能性があります。

- *最初のタイプのエラー*。これらのエラーは、安全なオブジェクトが悪意のあるものとして検出された場合に発生します（誤検知）。
- *2番目のタイプのエラー*。これらのエラーは、悪意のあるオブジェクトが安全であると検出されたときに発生します。

したがって、ヒューリスティックアナライザによって検出されたオブジェクトは*疑わしい*ものとして扱われます。

疑わしいオブジェクトは隔離に移動することをお勧めします。ウイルスデータベースを更新した後、そのようなファイルはシグネチャ解析を使用してスキャンできます。2番目のタイプのエラーを避けるには、ウイルスデータベースを最新の状態に保ってください。

Dr.Web Virus-Finding Engineスキャンエンジンを使用すると、ファイルと圧縮されたオブジェクトの両方、または異なるコンテナ内のオブジェクト（アーカイブやメールメッセージなど）をスキャンして修復できます。

コマンドライン引数

スキャンエンジンDr.Web Scanning Engineをコマンドラインから実行するには、次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-se <socket> [<parameters>]
```

必須の*<socket>*引数は、クライアントコンポーネントの要求を処理するためにDr.Web Scanning Engineによって使用されるソケットのアドレスを示します。ファイルパス（UNIXソケット）としてのみ設定できます。

Dr.Web Scanning Engineは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし。
追加の起動パラメータ（設定ファイルのパラメータと同じものであり、必要に応じて置き換えます）。	

--CoreEnginePath	**機能**：Dr.Web Virus-Finding Engine スキャンエンジンのライブラリへのパスを指定します。 **短縮形**：なし。 **引数**：*<path to the file>* - 使用するライブラリへのフルパス。
--VirusBaseDir	**機能**：ウイルスデータベースファイルがあるディレクトリへのパスを指定します。 **短縮形**：なし。 **引数**：*<path to the directory>* - ウイルスデータベースディレクトリへのパス。
--TempDir	**機能**：一時ファイルがあるディレクトリへのパスを指定します。 **短縮形**：なし。 **引数**：*<path to the directory>* - 一時ファイルを含むディレクトリへのフルパス。
--Key	**機能**：キーファイルへのパスを指定します。 **短縮形**：なし。 **引数**：*<path to the file>* - 使用するキーファイルへのフルパス。
--MaxForks	**機能**：スキャン中に Dr.Web Scanning Engine によって起動できる子プロセスの最大許容数を指定します。 **短縮形**：なし。 **引数**：*<number>* - 子プロセスの最大許容数。
--WatchdogInterval	**説明**：Dr.Web Scanning Engine が、子プロセスが動作可能かどうかをチェックし、応答を停止したプロセスを停止する頻度。 **短縮形**：なし。 **引数**：*<time interval>* - 子プロセスをチェックする頻度。
--Shelltrace	**機能**：シェルトレースをオンにします（Dr.Web Virus-Finding Engine によって実行されたファイルスキャンの詳細情報をログに記録します）。 **短縮形**：なし。 **引数**：なし。
--LogLevel	**説明**：動作中に Dr.Web Scanning Engine によって実行されるロギングのレベルを設定します。 **短縮形**：なし。 **引数**：*<logging level>*。使用可能な値： • DEBUG - 最も詳細なロギングレベル。すべてのメッセージとデバッグ情報が登録されます。 • INFO - すべてのメッセージが登録されます。 • NOTICE - すべてのエラーメッセージ、警告、通知が登録されます。 • WARNING - すべてのエラーメッセージと警告が登録されます。 • ERROR - エラーメッセージのみが登録されます。
--Log	**説明**：コンポーネントメッセージのロギングの方法を指定します。 **短縮形**：なし。 **引数**：*<log type>*。使用可能な値： • Stderr[:ShowTimestamp] - メッセージは標準エラーストリームの *stderr* に出力されます。 追加オプション ShowTimestamp は、すべてのメッセージにタイムスタンプを追加するために使用します。 • Syslog[:*<facility>*] - メッセージはシステムロギングサービス syslog に送信されます。

	追加オプション *<facility>*は、`syslog`のメッセージ登録レベルを指定するために使用します。次の値を使用できます。 ○ `DAEMON` - デーモンのメッセージ。 ○ `USER` - ユーザープロセスのメッセージ。 ○ `MAIL` - メールプログラムのメッセージ。 ○ `LOCAL0` - ローカルプロセス0のメッセージ。 ... ○ `LOCAL7` - ローカルプロセス7のメッセージ。 • *<path>* - すべてのメッセージが登録されているファイルへのパス。 **例**： `--Log /var/opt/drweb.com/log/se.log` `--Log Stderr:ShowTimestamp` `--Log Syslog:DAEMON`

例：

```
$ /opt/drweb.com/bin/drweb-se /tmp/drweb.ipc/.se --MaxForks=5
```

このコマンドはDr.Web Scanning Engineスキャンエンジンのインスタンスを起動し、クライアントコンポーネントとのインタラクション用の`/tmp/drweb.ipc/.se` UNIXソケットを作成し、子スキャンプロセスの数を5つに制限します。

スタートアップノート

必要に応じて、Dr.Web Scanning Engineスキャンエンジンのインスタンスはいくつでも起動できます。インスタンスは、（Dr.Web for UNIX Mail Serversのコンポーネントだけでなく）クライアントアプリケーションにもスキャンサービスを提供します。その場合、コンポーネントの設定で`FixedSocketPath`パラメータの値が指定されていると、スキャンエンジンの1つのインスタンスがDr.Web ConfigD設定デーモンによって常に実行され、このUNIXソケットを介してクライアントから常に利用可能になります。コマンドラインから直接起動したスキャンエンジンのインスタンスは、設定デーモンが実行中であっても、設定デーモンへの接続を確立せずに自律モードで動作します。コンポーネントの動作を管理し、必要に応じてファイルをスキャンするには、Dr.Web for UNIX Mail ServersのコマンドラインベースツールDr.Web Ctlを使用できます（`drweb-ctl`コマンドを使用して起動します）。

Dr.Web Scanning Engineを使用して任意のファイルまたはディレクトリをスキャンするには、Dr.Web Ctlツールの`rawscan`コマンドを使用します。

```
$ drweb-ctl rawscan <path to file or directory>
```

コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-se`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[ScanEngine]`セクションで指定されている設定パラメータを使用します。

このセクションは以下のパラメータを保存します。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-se • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-se • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-se
FixedSocketPath *{path to file}*	Dr.Web Scanning EngineスキャンエンジンのUNIXソケットへのパス。 このパラメータが指定されている場合、Dr.Web ConfigD設定デーモンは、このソケットを介してクライアントが使用可能なスキャンエンジンのコピーが常に実行されていることを確認します。 デフォルト値：*（未設定）*
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 FixedSocketPathパラメータが設定されている場合、この設定は無視されます（指定した時間を経過しても、コンポーネントは動作を終了しません）。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：1h
MaxForks *{integer}*	同時に実行できる、Dr.Web Scanning Engineによって実行される子プロセスの最大許容数。 デフォルト値：使用可能なCPUコアの2倍の数が*自動的に使用されます*。算出された数が4未満の場合は4になります。
BufferedIo *{On / Off}*	ファイルをスキャンするときは、バッファ付き入出力（I/O）を使用します。 FreeBSDおよびGNU/Linux OSでバッファ付きI/Oを使用すると、低速ディスクでのファイルスキャン速度を上げることができます。 デフォルト値：Off
WatchdogInterval *{time interval}*	応答を停止したプロセス（「watchdog」）を検出するために、子プロセスが動作可能かどうかをDr.Web Scanning Engineがチェックする頻度。 デフォルト値：1.5s

Dr.Web Updater

更新コンポーネントのDr.Web Updaterは、Doctor Web更新サーバーからウイルスデータベースおよびDr.Web Virus-Finding Engineスキャンエンジンの利用可能な更新をすべて受信し、更新をDr.Web for UNIX製品コンポーネントのローカルクラウドと（Dr.Web MeshDが製品に含まれている場合はそれを経由して）同期させることを目的に設計されています。

Dr.Web for UNIX Mail Serversが集中管理モードで動作している場合、更新は集中管理サーバー（Dr.Web Enterprise Serverなど）から受信されます。その場合、すべての更新はDr.Web ES Agent経由でサーバーから受信され、Dr.Web Updaterは更新のダウンロードには使用されません（Dr.Web for UNIX製品のローカルクラウドと更新が同期されることもありません）。

動作原理

このコンポーネントは、Doctor Web更新サーバーへの接続を確立して、ウイルスデータベースとDr.Web Virus-Finding Engineスキャンエンジン、Webリソースカテゴリーのデータベース、アンチスパムコンポーネントの更新がないか確認することを目的に設計されています。利用可能な更新ゾーンを構成するサーバーのリストは、特別なファイルに保存されます（ファイルは改変を防ぐために署名されています）。Doctor Web更新サーバーへの接続では、基本認証とダイジェスト認証のみがサポートされています。

Dr.Web for UNIX Mail Serversが集中管理サーバーに接続されていない場合、またはモバイルモードでサーバーに接続されている場合、Dr.Web UpdaterはDr.Web ConfigD設定デーモンによって自動的に起動されます。起動は、設定で指定された周期で実行されます。適切なコマンドをユーザーから受け取った場合、このコンポーネントは設定デーモンによって起動することもできます（スケジュールされていない更新）。

サーバー上で利用可能になった更新は、*<var_dir>*/cacheディレクトリ（GNU/Linuxの場合は var/opt/drweb.com/cache）にダウンロードされ、その後Dr.Web for UNIX Mail Serversの作業ディレクトリに移されます。

デフォルトでは、更新はすべてDr.Webの全製品に共通の更新ゾーンから実行されます。更新ゾーンに含まれる、デフォルトで使用されるサーバーのリストは、*DrlDirパラメータで定義されたディレクトリにあるファイルで指定され、更新タイプ別にグループ化されています（ウイルスデータベースおよびスキャンエンジンと、Webリソースカテゴリーのデータベース）。これらのファイルは、更新されたコンポーネント（ウイルスデータベース、スキャンエンジン、アンチスパムコンポーネント）によってグループ化されています。ユーザーのリクエストに応じて（更新タイプごとに）特別な更新ゾーンを作成できます。これが、*CustomDrlDirパラメータで指定されたディレクトリにある、別のファイル（デフォルト名はcustom.drl）で指定されているサーバーリストです。この場合、更新コンポーネントは、デフォルトゾーンのサーバーを使用せずに、これらのサーバーからのみ更新を受信します。

特別な更新ゾーンを使用しない場合は、コンポーネント設定で該当するパラメータの*CustomDrlDir値を削除します。

> ⚠ サーバーリストを含むファイルの中身は署名されているため、ファイルを変更することはできません。更新サーバーの特別なリストを作成する必要がある場合は、テクニカルサポートにお問い合わせください。

このコンポーネントは、ユーザーが次に更新のロールバックをリクエストする場合に備えて、更新したファイルをバックアップできます。バックアップするファイルの場所と詳細レベルは設定で指定できます。更新をロールバックするに

は、Dr.Web Ctl Dr.Web Ctlコマンドラインからソリューションを管理するDr.Web for UNIX Mail Servers用のコマンドラインツールを使用します（drweb-ctlコマンドで実行されます）。

Dr.Web for UNIX Mail ServersがDr.Web for UNIX製品のローカルクラウドに接続されていて、集中管理サーバーに接続されていない場合、Dr.Web Updaterコンポーネントは、クラウドホストによって受信された更新の同期にも使用されます。つまり、更新サーバーが受信した更新をクラウドに送信し、クラウドから更新を受信することになるため、Dr.Web更新サーバーの総負荷を軽減できます。このオプションはコンポーネントの設定で有効または無効にできます。

コマンドライン引数

Dr.Web Updaterを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-update [<parameters>]
```

Dr.Web Updaterは次のオプションを処理できます。

パラメータ	説明
--help	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：-h **引数**：なし
--version	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：-v **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-update --help
```

このコマンドは、Dr.Web Updaterに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理し、ウイルスデータベースとスキャンエンジンを更新するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これはdrweb-ctlコマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、man 1 drweb-updateコマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの[Update]セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-update • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-update • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-update
RunAsUser *{UID / user name}*	このパラメータは、コンポーネントを実行するユーザー名を決定します。ユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「name:」というプレフィックスを付けて指定します。たとえば、RunAsUser = name:123456です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：drweb
UpdateInterval *{time interval}*	Dr.Web更新サーバーで更新を確認する頻度。これは、（自動または手動で開始した）更新サーバーへの接続が成功してから次に更新の実行を試みるまでの時間間隔です。 デフォルト値：30m
RetryInterval *{time interval}*	前回の試行が失敗した場合に、更新サーバーを使用して更新の実行を再試行する頻度。 指定可能な値：1分（1m）から30分（30m）まで。 デフォルト値：3m
MaxRetries *{integer}*	前回の試行が失敗した場合に、更新サーバーを使用して（RetryIntervalで指定された頻度で）更新の実行を繰り返し試みる回数。 値が0に設定されている場合、試行は繰り返されません（次の更新は、UpdateIntervalで指定された時間を経過した後に実行されます）。

パラメータ	説明
	デフォルト値：3
Proxy *{connection string}*	Dr.Web更新サーバーへの接続時にアップデーターコンポーネント（Dr.Web Updater）が使用するプロキシサーバーに接続するためのパラメータを保存します（外部サーバーへの直接接続がネットワークのセキュリティポリシーによって禁止されている場合など）。 パラメータ値が指定されていない場合、プロキシサーバーは使用されません。 使用可能な値： *<connection string>*は、プロキシサーバーの接続文字列です。文字列のフォーマット（URL）は以下のとおりです。 [*<protocol>*: //] [*<user>*: *<password>*@] *<host>*: *<port>* 各パラメータは次のとおりです。 • *<protocol>*は、使用されるプロトコルタイプです（現在のバージョンでは、httpのみが使用可能です）。 • *<user>*は、プロキシサーバーに接続するためのユーザー名です。 • *<password>*は、プロキシサーバーに接続するためのパスワードです。 • *<host>*は、プロキシのホストアドレスです（IPアドレスまたはドメイン名、つまりFQDN）。 • *<port>*は使用するポートです。 URLの *<protocol>*および *<user>:<password>*の部分がない場合があります。プロキシサーバーのアドレス *<host>:<port>*は必須です。 ユーザー名またはパスワードに「@」、「%」、「:」の文字が含まれている場合、これらの文字はそれぞれ「%40」、「%25」、「%3A」の16進コードに変更する必要があります。 例： 1. 設定ファイルでの設定。 • ポート*123*を使用した*proxyhost.company.org*でホストされているプロキシサーバーへの接続： `Proxy = proxyhost.company.org:123` • ポート*3336*を使用し、パスワードが「passw」のユーザー「legaluser」としてHTTPプロトコルを経由した*10.26.127.0*でホストされているプロキシサーバーへの接続： `Proxy = http:// legaluser:passw@10.26.127.0:3336` • ポート*3336*、ユーザー名「user@company.com」、パスワード「passw%123:」を使用した*10.26.127.0*でホストされているプロキシサーバーへの接続： `Proxy = user%40company.com:passw%25123%3A@10.26.127.0:3336`

パラメータ	説明
	2. drweb-ctl cfset コマンドを使用して、同じ値を設定する。 `# drweb-ctl cfset Update.Proxy proxyhost.company.org:123` `# drweb-ctl cfset Update.Proxy http://legaluser:passw@10.26.127.0:3336` `# drweb-ctl cfset Update.Proxy user%40company.com:passw%25123%3A@10.26.127.0:3336` デフォルト値：*（未設定）*
ExcludedFiles *{file name}*	Dr.Web Updaterコンポーネントによって更新されないファイルの名前を定義します。 リストをパラメータ値として指定できます。リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：以下のファイルをリストに追加します。123.vdbおよび456.dws。 1. 設定ファイルに値を追加します。 • 1つの文字列に2つの値： `[Update]` `ExcludedFiles = "123.vdb", "456.dws"` • 2つの文字列（文字列ごとに1つの値）： `[Update]` `ExcludedFiles = 123.vdb` `ExcludedFiles = 456.dws` 2. drweb-ctl cfset コマンドを使用して値を追加します。 `# drweb-ctl cfset Update.ExcludedFiles -a 123.vdb` `# drweb-ctl cfset Update.ExcludedFiles -a 456.dws` デフォルト値：drweb32.lst
NetworkTimeout *{time interval}*	更新プロセス中にUpdaterコンポーネントのネットワーク関連の動作に課されるタイムアウト時間。 このパラメータは、接続が一時的に切断されたときに使用されます。タイムアウトが切れる前に接続が再度確立された場合、中断された更新プロセスが続行されます。 75sを超えるタイムアウト値を指定した場合は、TCPタイムアウトによって接続が閉じられるため効力を持ちません。 最小値：5s デフォルト値：60s

パラメータ	説明
BaseDrlDir *{path to directory}*	標準的な更新ゾーンの更新サーバーへの接続に使用されるファイルを含むディレクトリへのパスを定義します。更新コンポーネントがウイルスデータベースおよびスキャンエンジンを更新するために使用されます。 デフォルト値：*<var_dir>*/drl/bases • GNU/Linuxの場合：/var/opt/drweb.com/drl/bases • FreeBSDの場合：/var/drweb.com/drl/bases
BaseCustomDrlDir *{path to directory}*	特別な「カスタマイズされた」更新ゾーンへの接続に使用されるファイルを含むディレクトリへのパスを定義します。ウイルスデータベースとスキャンエンジンを更新するために使用されます。 パラメータで定義されたディレクトリ内に、空でない署名付きサーバーリストファイル（.drlファイル）がある場合、更新はこれらのサーバーからのみ実行され、主要なゾーンサーバー（上記を参照）はウイルスデータベースおよびスキャンエンジンの更新には使用されません。 デフォルト値：*<var_dir>*/custom-drl/bases • GNU/Linuxの場合：/var/opt/drweb.com/custom-drl/bases • FreeBSDの場合：/var/drweb.com/custom-drl/bases
BaseUpdateEnabled *{Boolean}*	ウイルスデータベースとスキャンエンジンの更新が許可されているかどうかを示すインジケーター。 使用可能な値： • Yes - 更新は許可されており、実行されます。 • No - 更新は許可されていないため実行されません。 デフォルト値：Yes
VersionDrlDir *{path to directory}*	サーバーへの接続に使用されるファイルを含むディレクトリへのパスを定義します。Dr.Web for UNIX Mail Serversのバージョンの更新に使用されます。 デフォルト値：*<var_dir>*/drl/version • GNU/Linuxの場合：/var/opt/drweb.com/drl/version • FreeBSDの場合： /var/drweb.com/drl/version
VersionUpdateEnabled *{Boolean}*	Dr.Web for UNIX Mail Serversコンポーネントのバージョンの更新が許可されているかどうかを示すインジケーター。 使用可能な値： • Yes - 更新は許可されており、実行されます。 • No - 更新は許可されていないため実行されません。 デフォルト値：Yes
DwsCustomDrlPath *{path to file}*	特別な更新ゾーンのサーバーのリストを含む署名付きファイルへのパス。Webリソースカテゴリーのデータベースを更新するために使用されます。 パラメータが空ではなく、指定されたファイルが存在する場合、更新にはサーバーのみが使用されます。リストのメインファイル（上記を参照）は無視されます。パラメータによって認識されたファイルが空の場合、更新の試行は失敗します。

パラメータ	説明
	デフォルト値：*<var_dir>*/drl/dws/custom.drl • GNU/Linuxの場合：/var/opt/drweb.com/drl/dws/custom.drl • FreeBSDの場合：/var/drweb.com/drl/dws/custom.drl
DwsDrlDir *{path to directory}*	標準的な更新ゾーンのサーバーに接続するためのファイルを含むディレクトリへのパスを定義します。Webリソースカテゴリーのデータベースを更新するために使用されます。 デフォルト値：*<var_dir>*/drl/dws • GNU/Linuxの場合：/var/opt/drweb.com/drl/dws • FreeBSDの場合：/var/drweb.com/drl/dws
DwsCustomDrlDir *{path to directory}*	特別な「カスタマイズされた」更新ゾーンのサーバーに接続するためのファイルを含むディレクトリへのパスを定義します。Webリソースカテゴリーのデータベースを更新するために使用されます。 パラメータで定義されたディレクトリ内に、空でない署名付きサーバーリストファイル（.drlファイル）がある場合、更新はこれらのサーバーからのみ実行され、主要なゾーンサーバー（上記を参照）はWebリソースカテゴリーのデータベースを更新するためには使用されません。 デフォルト値：*<var_dir>*/custom-drl/dws • GNU/Linuxの場合：/var/opt/drweb.com/custom-drl/dws • FreeBSDの場合：/var/drweb.com/custom-drl/dws
DwsUpdateEnabled *{Boolean}*	Webリソースカテゴリーのデータベースの更新が許可されているかどうかを示すインジケーター。 使用可能な値： • Yes - 更新は許可されており、実行されます。 • No - 更新は許可されていないため実行されません。 デフォルト値：Yes
AntispamDrlDir *{path to directory}*	標準的な更新ゾーンのサーバーに接続するためのファイルを含むディレクトリへのパスを定義します。アンチスパムライブラリを更新するために使用されます。 デフォルト値：*<var_dir>*/drl/antispam • GNU/Linuxの場合：/var/opt/drweb.com/drl/antispam • FreeBSDの場合：/var/drweb.com/drl/antispam
AntispamCustomDrlDir *{path to directory}*	特別な「カスタマイズされた」更新ゾーンのサーバーに接続するためのファイルを含むディレクトリへのパスを定義します。アンチスパムライブラリを更新するために使用されます。 パラメータで定義されたディレクトリ内に、空でない署名付きサーバーリストファイル（.drlファイル）がある場合、更新はこれらのサーバーからのみ実行され、主要なゾーンサーバー（上記を参照）はアンチスパムライブラリを更新するためには使用されません。 デフォルト値：*<var_dir>*/custom-drl/antispam • GNU/Linuxの場合：/var/opt/drweb.com/custom-drl/antispam

パラメータ	説明
	• FreeBSDの場合：/var/drweb.com/custom-drl/antispam
AntispamUpdateEnabled *{Boolean}*	アンチスパムライブラリの更新が許可されているかどうかを示すインジケーター。 使用可能な値： • Yes - 更新は許可されており、実行されます。 • No - 更新は許可されていないため実行されません。 デフォルト値：No
BackupDir *{path to directory}*	ロールバックに備えて、更新済みファイルの旧バージョンが保存されているディレクトリへのパス。更新するたびに、更新したファイルのみがバックアップされます。 デフォルト値：/tmp/update-backup
MaxBackups *{integer}*	更新済みファイルの以前のバージョンの最大保存数。この数を超えると、最も古いコピーが次回の更新時に削除されます。 パラメータ値がゼロの場合、以前のバージョンのファイルは保存されません。 デフォルト値：0
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 このコンポーネントは、スケジュールによる次の更新時または明示的なコマンドdrweb-ctl update [--local-cloud]で起動されます。更新が完了すると、指定された時間、待機します。新しいリクエストがない場合は（UseLocalCloud = Yesの場合のクラウドとのインタラクションを含む）、次の更新を試行するまでシャットダウンします。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s
UseLocalCloud *{Boolean}*	Dr.Web更新サーバーに加えて、Dr.Web MeshDコンポーネント経由でDr.Web for UNIX製品のローカルクラウドと連携して、更新を同期します（クラウドに更新を送信し、クラウドから更新を取得します）。 使用可能な値： • No - 更新にはDr.Web更新サーバーのみ使用します。クラウドとの更新の同期は無効ですが、drweb-ctl update --local-cloudコマンドを使用して明示的に実行できます。 • Yes - ホスト上の更新をローカルクラウドと同期します（利用可能な更新がある場合はクラウドから更新を取得し、ホスト上の更新の方が新しい場合はクラウドに更新を送信します）。 デフォルト値：Yes

Dr.Web ES Agent

アンチウイルス集中管理エージェントDr.Web ES Agentは、Dr.Web for UNIX Mail Serversを集中管理サーバー（Dr.Web Enterprise Serverなど）に接続するように設計されています。

Dr.Web for UNIX Mail Serversが集中管理サーバーDr.Web ES Agentに接続されている場合、ローカル設定とライセンスキーファイルは、集中管理サーバーに保存されているキーファイルに応じて同期されます。さらにDr.Web ES Agentは、ウイルスイベントに関する統計情報や実行中のコンポーネントのリストとそのステータスを集中管理サーバーに送信します。

またDr.Web ES Agentは、更新コンポーネントDr.Web Updaterを経由せずに、接続されている集中管理サーバーから直接Dr.Web for UNIX Mail Serversのウイルスデータベースを更新します。

動作原理

Dr.Web ES Agentは集中管理サーバー（Dr.Web Enterprise Serverなど）への接続を確立します。これにより、ネットワーク管理者はネットワーク内で共通のセキュリティポリシーを実装し、特にすべてのネットワーク端末とサーバーに対して同じスキャン設定と脅威検出への対応を設定できます。さらに、集中管理サーバーは最新のウイルスデータベースを保存するので、ネットワークの内部更新サーバーの役割も果たします（この場合、更新はDr.Web ES Agentで実行され、Dr.Web Updaterは使用されません）。

Dr.Web ES Agentを集中管理サーバーに接続する際、エージェントはプログラムコンポーネントとライセンスキーファイルの最新の設定を受信したことを確認します。その後、それらの設定はDr.Web ConfigD設定デーモンに送信されて、管理対象のコンポーネントに適用されます。さらに、コンポーネントは端末のファイルシステムオブジェクトをスキャンするタスク（スケジュールされたタスクを含む）も受信します。

Dr.Web ES Agentは、検出された脅威と適用されたアクションに関するサーバーの統計情報を収集して送信します。

Dr.Web ES Agentを集中管理サーバーに接続するには、ホスト（集中管理サーバーでは「端末」）のパスワードとIDに加えて、認証のためにサーバーによって使用されるパブリック暗号化キーファイルが必要です。端末IDの代わりに、端末が含まれるプライマリグループのIDを指定できます。必要なIDとパブリックキーファイルについては、アンチウイルスネットワークの管理者に問い合わせてください。

さらに、このオプションが集中管理サーバーで許可されている場合は、保護されたサーバー（「ワークステーション」）にホストを「新規端末」として接続できます。この場合、管理者が接続リクエストを確認した後に、集中管理サーバーでは自動的にIDとパスワードを生成し、今後の接続のためにそれをエージェントに送信します。

コマンドライン引数

Dr.Web ES Agentを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-esagent [<parameters>]
```

Dr.Web ES Agentは次のオプションを処理できます。

パラメータ	説明

`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-esagent --help
```

このコマンドは、Dr.Web ES Agentに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。OSの起動時にDr.Web ConfigD設定デーモンによって自動的に起動します。コンポーネントの動作を管理し、Dr.Web for UNIX Mail Serversを集中管理サーバーに接続するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-esagent`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[ESAgent]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。 デフォルト値：`Notice`
`Log` *{log type}*	コンポーネントのロギング方式。 デフォルト値：`Auto`
`ExePath` *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*`/bin/drweb-esagent`

パラメータ	説明
	• GNU/Linuxの場合：`/opt/drweb.com/bin/drweb-esagent` • FreeBSDの場合：`/usr/local/libexec/drweb.com/bin/drweb-esagent`
`DebugIpc` *{Boolean}*	IPCメッセージをデバッグログ`LogLevel = DEBUG`に書き込みます（Dr.Web ES AgentとDr.Web ConfigD設定デーモンとの間のやり取り）。 デフォルト値：`No`
`MobileMode` *{On / Off / Auto}*	集中管理サーバーに接続したときのモバイルモードを有効／無効にします。 使用可能な値： • `On` - 集中管理サーバーで許可される場合は、モバイルモードを使用します（つまり、Dr.Web Updaterを介してDoctor Webの更新サーバーから更新を実行します）。 • `Off` - モバイルモードを使用せずに、集中管理モードで動作を継続します（更新は常に集中管理サーバーから受信します）。 • `Auto` - 集中管理サーバーで許可される場合は、モバイルモードを使用し、使用できる接続や接続品質の高さに応じて、Dr.Web Updaterを介したDoctor Webの更新サーバーと集中管理サーバーの両方から更新を実行します。 このパラメータの動作はサーバーの権限に依存することに注意してください。モバイルモードが使用するサーバーで許可されていない場合、このパラメータは無効です。 デフォルト値：`Auto`
`Discovery` *{On / Off}*	エージェントが、集中管理サーバーに組み込まれているネットワークインスペクターからの*discovery*リクエストの受信を有効にするかどうかを指定します（*discovery*リクエストは、アンチウイルスネットワークの構造と状態を確認するためにインスペクターによって使用されます）。 使用可能な値： • `On` - *discovery*リクエストの受信と処理を有効にします。 • `Off` - *discovery*リクエストの受信と処理を無効にします。 このパラメータは集中管理サーバーの設定よりも優先順位が高いことに注意してください。パラメータ値が`Off`に設定されている場合、このオプションがサーバーで有効になっていても、エージェントはdiscoveryリクエストを受信しません。 デフォルト値：`On`
`UpdatePlatform` *{platform name}*	エージェントが、集中管理サーバーからのスキャンエンジンの更新の受信を有効にするかどうかを指定します。スキャンエンジンは指定のプラットフォーム向けに開発されています。*プラットフォーム名*は、プラットフォーム名を含む文字列です。 使用可能な値： • GNU/Linuxの場合：`unix-linux-32`、`unix-linux-64`、`unix-linux-mips` • FreeBSDの場合：`unix-freebsd-32`、`unix-freebsd-64` • Darwinの場合：`unix-darwin-32`、`unix-darwin-64`

パラメータ	説明
	 パラメータ値は確実に変更が必要な場合にのみ変更することを強くお勧めします。 デフォルト値：*現在使用されているプラットフォームによる*
`SrvMsgAutoremove` *{integer}*	集中管理サーバーからのメッセージが自動的に削除されるまでの保存期間 使用可能な値：1週間（`1w`）から365日（`365d`）まで 保存期間は整数で、サフィックス（`s`、`m`、`h`、`d`、`w`）を付けて指定します。 デフォルト値：`1w`

Dr.Web HTTPD

Dr.Web HTTPDは、HTTP経由で（たとえばWebブラウザ経由で）Dr.Web for UNIX Mail Serversとローカルおよびリモートで対話するためのインフラストラクチャを提供します。このコンポーネントは、Dr.Web for UNIX Mail Serversを管理するためのインターフェースを提供します。

Dr.WebのWebインターフェースを介してDr.Web for UNIX Mail Serversを管理するだけでなく、Dr.Web HTTPDのコマンドインターフェース（HTTP API）を直接使用し、HTTPSを介してDr.Web for UNIX Mail Serversのコンポーネントと対話することもできます。この機能により、Dr.Web for UNIX Mail Serversを管理するための独自のインターフェースを作成できます。

Dr.Web HTTPDが提供するHTTP APIの詳細については、該当するセクションを参照してください。

安全なHTTPS接続を使用するには、適切なSSLサーバー証明書とプライベートキーをDr.Web HTTPDに提供する必要があります。デフォルトでは、インストール中にDr.Web HTTPD用のSSLサーバー証明書とSSLプライベートキーが自動的に生成されますが、必要に応じて独自の証明書とキーを生成することもできます。また、Dr.Web HTTPDによって信頼されている認証局証明書で署名されたユーザーの個人用認証証明書を、Dr.Web HTTPDに接続するときの自動クライアント認証に使用することもできます。

SSLキーと証明書を生成するには、`openssl`ユーティリティを使用できます。`openssl`ユーティリティを使用して証明書とプライベートキーを生成する方法の例については、付録E. SSL証明書を生成するのセクションを参照してください。

動作原理

Dr.Web HTTPDはDr.Web for UNIX Mail Serversの動作を管理するためのWebサーバーです。Dr.Web HTTPDがあれば、外部Webサーバー（Apache HTTP ServerやNginxなど）やWebminなどの管理サービスを使用せずに済みます。さらにこのコンポーネントは、同じホスト上にあるそのようなサーバーやサービスと同時に機能することができ、それらの動作を妨げることはありません。

Dr.Web HTTPDサーバーは、HTTPおよびHTTPSプロトコルを介し、設定で指定されたソケットで受信したリクエストを処理します。このため、このサーバーは、Webサーバーと同じホスト上で動作しているときに、それらのサーバーと競合することはありません。Dr.Web for UNIX Mail Serversの管理には、安全なHTTPSプロトコルが使用されます。

Dr.Web管理Webインターフェースをインストールすることは、Dr.Web for UNIX Mail Serversを正しく機能させるための必須事項ではありません。インストールしなくても問題ありません。対応するブロックが破線で囲まれているのはこのためです。

Dr.Web HTTPDコンポーネントは、Dr.Web for UNIX Mail Servers Dr.Web ConfigD設定デーモンの他、ファイルスキャン用のDr.Web File Checkerコンポーネントやその他のコンポーネントにコマンドを送信します。これらのコマンドは、提供されているHTTP APIを介して受信したコマンドに基づいています。

Dr.Web HTTPDを使用するDr.Web for UNIX Mail Serversの管理WebインターフェースがDr.Web for UNIX Mail Serversに含まれている場合は、該当するセクションにその説明が記載されています。

Dr.Webの管理WebインターフェースがDr.Web for UNIX Mail Serversに含まれていない場合は、Dr.Web HTTPDによるHTTP APIを対話に使用する、任意の外部管理インターフェースを接続できます（HTTP APIの説明のセクションで説明しています）。

コマンドライン引数

Dr.Web HTTPDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-httpd [<options>]
```

Dr.Web HTTPDは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-httpd --help
```

このコマンドは、Dr.Web HTTPDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます（通常はOSの起動時）。コンポーネントが実行されていて、Webインターフェースがインストールされている場合は、標準のWebブラウザを使用して、Webインターフェースが提供されているアドレスにHTTPS経由でアクセスするだけで、Dr.Web for UNIX Mail Serversのコンポーネントを管理できます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを取得するには、`man 1 drweb-httpd`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[HTTPD]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
LogLevel *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-httpd • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-httpd • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-httpd
Start *{Boolean}*	Dr.Web ConfigD設定デーモンによってコンポーネントを起動するかどうかを指定します。 このパラメータにYes値を指定すると、設定デーモンはただちにコンポーネントを開始します。また、No値を指定すると、設定デーモンはただちにコンポーネントを終了します。 デフォルト値：管理インターフェースがインストールされているかどうかによって*異なります*。
AdminListen *{address, ...}*	Dr.Web HTTPDが、管理者権限を持つクライアントからの（HTTPS経由での）接続をリッスン（待ち受け）しているネットワークソケット（すべてのネットワークソケットは*<IP address>:<port>*で構成されます）のリスト。これらのソケットは、管理Webインターフェース（Webインターフェースがインストールされている場合）への接続とHTTP APIへのアクセスの両方に使用されます。 リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が1つのリストにまとめられます）。 例：ソケット192.168.0.1:1234および10.20.30.45:5678をリストに追加します。 1. 設定ファイルに値を追加します。 • 1行に2つの値： `[HTTPD]` `AdminListen = "192.168.0.1:1234", "10.20.30.45:5678"` • 2行（1行に1つの値）： `[HTTPD]` `AdminListen = 192.168.0.1:1234` `AdminListen = 10.20.30.45:5678`

パラメータ	説明
	2. drweb-ctl cfset コマンドを使用して値を追加します。 `# drweb-ctl cfset HTTPD.AdminListen -a 192.168.0.1:1234` `# drweb-ctl cfset HTTPD.AdminListen -a 10.20.30.45:5678` 値が指定されていない場合、HTTP APIとWebインターフェース(インストールされている場合)を使用することはできません。 デフォルト値:127.0.0.1:4443
PublicListen *{address, ...}*	Dr.Web HTTPDが、制限された権限を持つクライアントからの(HTTP経由での)接続をリッスン(待ち受け)しているネットワークソケット(すべてのネットワークソケットは *<IP address>:<port>* で構成されます)のリスト リストの値は、コンマ(引用符内の各値)で区切る必要があります。パラメータはセクションで複数回指定できます(この場合、そのすべての値が1つのリストにまとめられます)。 例:ソケット192.168.0.1:1234および10.20.30.45:5678をリストに追加します。 1. 設定ファイルに値を追加します。 • 1行に2つの値: `[HTTPD]` `PublicListen = "192.168.0.1:1234", "10.20.30.45:5678"` • 2行(1行に1つの値): `[HTTPD]` `PublicListen = 192.168.0.1:1234` `PublicListen = 10.20.30.45:5678` 2. drweb-ctl cfset コマンドを使用して値を追加します。 `# drweb-ctl cfset HTTPD.PublicListen -a 192.168.0.1:1234` `# drweb-ctl cfset HTTPD.PublicListen -a 10.20.30.45:5678` これらのアドレス(ソケット)では、HTTP APIのすべてのコマンドにアクセスしたり、管理Webインターフェースにアクセスしたりすることはできません。 デフォルト値:*(未設定)*
AdminSslCertificate *{path to file}*	Webインターフェースサーバーが管理ソケットへの接続を確立するクライアントとHTTPS経由で通信するために使用するサーバー証明書ファイルへのパス。 このファイルは、コンポーネントのインストール中に自動的に生成されます。 証明書ファイルとプライベートキーファイル(後述のパラメータで指定されます)は、一致するペアを形成する必要があります。

パラメータ	説明
	デフォルト値：*<etc_dir>*/certs/serv.crt • GNU/Linuxの場合：/etc/opt/drweb.com/certs/serv.crt • FreeBSDの場合：/usr/local/etc/drweb.com/certs/serv.crt
AdminSslKey *{path to file}*	Webインターフェースサーバーが管理ソケットへの接続を確立するクライアントとHTTPS経由で通信するために使用するプライベートキーファイルへのパス。 このファイルは、コンポーネントのインストール中に自動的に生成されます。 証明書ファイル（前述のパラメータで指定されます）とプライベートキーファイルは、一致するペアを形成する必要があります。 デフォルト値：*<etc_dir>*/certs/serv.key • GNU/Linuxの場合：/etc/opt/drweb.com/certs/serv.key • FreeBSDの場合：/usr/local/etc/drweb.com/certs/serv.key
AdminSslCA *{path to file}*	HTTPS経由で管理ソケットに接続しているクライアントから提供された証明書をチェックするための、信頼できる認証局（CA）の証明書として機能する証明書ファイルへのパス クライアントの証明書が、このパラメータで指定された証明書で署名されている場合、このクライアントは認証のためにログイン情報／パスワードのペアを入力する必要はありません。また、このパラメータで設定された証明書で署名されているクライアント証明書を使用するクライアントでは、ログイン情報／パスワードによる認証は禁止されます。 この証明書ベースの認証に成功したクライアントは、常にスーパーユーザー（*root*）として扱われます。 デフォルト値：*（未設定）*
WebconsoleRoot *{path to directory}*	管理Webインターフェースがインストールされている場合に、その管理Webインターフェースによって使用されるファイルがあるディレクトリ（Apache HTTP Serverのhtdocsディレクトリ相当）へのパス デフォルト値：*<opt_dir>*/share/drweb-httpd/webconsole • GNU/Linuxの場合：/opt/drweb.com/share/drweb-httpd/webconsole • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-httpd/webconsole
AccessLogPath *{path to file}*	クライアントからWebインターフェースサーバーへのすべてのHTTP/HTTPSリクエストが登録されるファイルへのパス。 指定しない場合、HTTP/HTTPSリクエストはファイルに記録されません。 デフォルト値：*（未設定）*

HTTP APIの説明

このセクションの内容

• 概要

- ユーザー認証と承認
- Dr.Web for UNIX Mail Serversの管理
- 脅威のリストの管理
- 隔離の管理
- HTTP APIの使用例

1. 概要

HTTP APIは、HTTPプロトコルを介してDr.Web for UNIX Mail Serversを制御および管理する手段として提供されます（セキュリティを確保するために、HTTPSプロトコルが使用されます）。

HTTPプロトコルのバージョン1.0が使用されます。APIは、HTTPプロトコルの標準メソッドである`GET`と`POST`を使用します。特に指定がない限り、すべてのデータはJSONオブジェクトの形式で送信されます。HTTP POSTリクエストの本文でJSONオブジェクトを送信する場合は、`Content-Type:`ヘッダーを`application/json`の値で使用します。

HTTPリクエストに対するHTTPレスポンスのフォーマット

- 特に指定のない限り、すべてのリクエストへのレスポンスとしてJSONオブジェクトが返されます。リクエストの処理中にエラーが発生した場合、Error JSONが返されます。
- レスポンスとして送信されたJSONオブジェクトにArrayタイプのフィールドがあり、この配列に要素が1つも含まれていない場合、このフィールドはサーバーからのレスポンスから省略されます。
- 特に指定のない限り、すべてのレスポンスの`Content-Type`:ヘッダーフィールドには`application/json`値があります。
- 存在しないエンドポイントをクライアントが要求した場合、*コード*フィールドに`EC_UNEXPECTED_MESSAGE`が含まれるError JSONオブジェクトが返されます。
- SCS（*Secure Cookie Sessions for HTTP*）が使用されている場合（以下を参照）、レスポンスには*SCS cookie*が含まれます。

JSONオブジェクト内の文字列のエンコード

- 文字列はUTF-8エンコーディング（BOMなし）で送信されます。ASCII表の一部ではない記号は、送信JSON文字列内で`\uXXXX`のようなシーケンスでエスケープされませんが、UTF-8エンコードで送信されます。
- 受信JSONオブジェクトの文字列には、UTF-8でエンコードされた記号と`\uXXXX`のようなエスケープシーケンスの両方を含めることができます。

データ転送に関する一般的な制限

- 本文にJSONオブジェクトが含まれるPOSTリクエストでは、RFC 7159に準拠するすべての記号が許可されます。
- GETリクエストでは、RFC 1945に準拠するすべての記号がURIで許可されます。
- RFC 1945に準拠する記号は、リクエストの他のどの部分（ヘッダーまたは本文）でも使用できます。

2. ユーザー認証と承認

APIの使用を開始するには、サーバーによる認証が必要です。承認の手段は2つ用意されています。

1. RFC 6896に準拠したSCSを使用する。
2. Dr.Web HTTPDが信頼できるCAの証明書と見なす特別な証明書で署名された、クライアントのSSL証明書を使用する。この場合、クライアントは、認証を受けるためにルートの認証情報を正しく入力したかのように扱われます（X.509クライアント証明書が使用されます）。

SCSを使用する場合、認証を確認するcookieは、リクエストでは`Cookie:`、レスポンスでは`Set-Cookie:`がヘッダーで送信されます。
SSL証明書による承認の場合、cookieは使用されません。

SCSで承認する場合、`login`コマンドを送信することでAPIの使用が始まります。このコマンドが正常に実行されると、レスポンスとしてSCS cookieがクライアントに送信されます。
クライアント証明書で承認する場合、`login`コマンドを実行する必要はありません。実行しようとすると、レスポンスにError JSONオブジェクトが返されます。

2.1. ログインとパスワードを指定する（SCS）

ユーザー認証および承認コマンド:

APIコマンド	説明
`login`	アクション: 指定されたユーザー名とパスワードに基づいてクライアントを認証し、HTTP APIのコマンドを使用することをクライアントに許可します。認証が成功すると、SCS cookieが返されます。 URI: `/api/10.2/login` **HTTP**メソッド: POST 入力パラメータ: AuthOptionsオブジェクト 正常に実行された結果: 空のオブジェクト、SCS cookie
`logout`	アクション: 提供されたSCS cookieを取り消します。その後、取り消されたSCS cookieを含むHTTP API呼び出しへのレスポンスとして、`EC_NOT_AUTHORIZED`エラーコードを含むErrorオブジェクトが返されます。 URI: `/api/10.2/logout` **HTTP**メソッド: GET 入力パラメータ: SCS cookie 正常に実行された結果: 空のオブジェクト
`whoami`	アクション: 認証されたユーザーの名前を表示します。 URI: `/api/10.2/whoami` **HTTP**メソッド: GET

APIコマンド	説明
	入力パラメータ：（SCS cookie）* 正常に実行された結果：whoamiオブジェクト、（SCS cookie）

*）SCS cookieは、SCSによる認証が使用される場合にのみ送受信する必要があるため、これ以降は括弧に入れて表記します。

SCSで認証する場合にのみ、loginとlogoutコマンドが使用されます。

使用されるオブジェクトの説明

1）AuthOptions - 完全なHTTP APIを使用するために認証および承認される必要があるユーザーのログインデータを含むオブジェクト：

```
{
 "user": string, //User name
 "password": string //User's password
}
```

管理者グループ（DebianとUbuntuではsudoers、CentOSとFedoraではwheel、Astra Linuxではastra-adminなど）のメンバーであるユーザーを指定できます。ユーザーが管理者グループのメンバーでない場合、レスポンスにEC_NOT_AUTHORIZEDエラーが返されます。

2）whoami - HTTP APIを使用することを許可されたユーザーの名前を含むオブジェクト：

```
{
 "whoami" :
 {
 "user": string //User name
 }
}
```

3）Error - 発生したエラーに関する情報を含むオブジェクト：

```
{
 "error" :
 {
 "code" : string, //A string specifying an error code that looks like
EC_XXX
 *"what": string //Description of the error
 }
}
```

アスタリスク（*）のパラメータは任意です。

リクエストの処理中にエラーが発生した場合にHTTP APIコマンドへのレスポンスとして返されるError JSONオブジェクトには、数値のエラーコードではなく、Dr.Web for UNIX Mail Serversのコンポーネントによって使用される内部文字列型コードを含むcodeフィールドがあります。このコードは、EC_XXXのような文字列です。対応する数値コードやエラーの詳細情報を確認するには、『管理者マニュアル』の付録Fにある「既知のエラー」セクションを参照してください。

2.2. 個人証明書を使用する認証

SSL証明書による認証は、Dr.Web HTTPDの設定で信頼済みとして指定された認証局証明書によって個人証明書が署名されていることを前提としています。証明書で認証された場合、すべてのリクエストは*root*ユーザーの権限で行われたものと見なされます。

個人ユーザー証明書で承認するには

1. 認証局証明書で署名された個人証明書を作成します。
2. Dr.Web HTTPDの設定（パラメータAdminSslCA）で、個人証明書に署名する認証局証明書へのパスを指定します。
3. Dr.Web HTTPDに接続するたびに、署名付き証明書を使用します。

必要に応じて、付録E. SSL証明書を生成するのセクションを参照してください。

3. Dr.Web for UNIX Mail Serversを管理する

設定パラメータの現在の値を表示および変更するためのAPIコマンド:

APIコマンド	説明
設定を管理するコマンド	
get_lexmap	アクション：現在の設定（ここではパラメータの「語彙マップ」と呼ばれます）のパラメータ値を取得します。 URI: /api/10.2/get_lexmap HTTPメソッド：GET 入力パラメータ：（SCS cookie） 正常に実行された結果：LexMapsオブジェクト、（SCS cookie）
set_lexmap	アクション：現在の設定の指定されたパラメータを設定または（デフォルトに）リセットします（パラメータの「語彙マップ」として送信されます）。 URI: /api/10.2/set_lexmap HTTPメソッド：POST 入力パラメータ：（SCS cookie）、LexMapオブジェクト 正常に実行された結果：SetOptionResultオブジェクト、（SCS cookie）

APIコマンド	説明
コマンドの更新	
`start_update`	アクション：更新を起動します。 URI: `/api/10.2/start_update` HTTPメソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：StartUpdateオブジェクト、（SCS cookie）
`stop_update`	アクション：アクティブな更新プロセスを停止します。 URI: `/api/10.2/stop_update` HTTPメソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：空のオブジェクト、（SCS cookie）
`baseinfo`	アクション：ダウンロードしたウイルスベースに関する情報を表示します。 URI: `/api/10.2/baseinfo` HTTPメソッド：GET 入力パラメータ：（SCS cookie） 正常に実行された結果：VirusBaseInfoオブジェクトを含むBaseInfoResultオブジェクト（SCS cookie）
ライセンス管理コマンド	
`install_license`	アクション：指定されたキーファイルをインストールします。 URI: `/api/10.2/install_license` HTTPメソッド：POST 入力パラメータ：（SCS cookie）、キーファイル本体（またはキーファイルを含むアーカイブ） 正常に実行された結果：空のオブジェクト、（SCS cookie）
集中管理サーバーに接続するためのコマンド	
`esconnect`	アクション：集中管理モードを有効にします。 URI: `/api/10.2/esconnect` HTTPメソッド：POST

APIコマンド	説明
	入力パラメータ：（SCS cookie）、ESConnectionオブジェクト 正常に実行された結果：空のオブジェクト、（SCS cookie）
esdisconnect	アクション：集中管理モードを無効にします。 URI: /api/10.2/esdisconnect HTTPメソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：空のオブジェクト、（SCS cookie）
その他	
idpass	アクション：脅威Dr.Web MailDを含む、再圧縮されたアーカイブのパスワードを取得します。 URI: /api/10.2/idpass HTTPメソッド：POST 入力パラメータ：（SCS cookie）、IdpassRequestオブジェクト 正常な結果：パスワードを含む文字列、（SCS cookie）

製品のコンポーネントの設定が返され、いわゆる*語彙マップ*、つまり一連のパラメータと値のペアとして設定されます。LexMapsオブジェクトには常に3つのLexMapsオブジェクトが含まれます。

- *active* - パラメータの現在の値
- *hardcoded* - 値がない、または無効な値のパラメータに自動的に割り当てられるデフォルト値
- *master* - クライアントによって設定された設定パラメータの値

get_lexmapコマンドは、実際にインストールされて実行されているコンポーネントだけでなく、Dr.Web for UNIX Mail Serversに含めることができるすべてのコンポーネントについて、常に3つすべての設定パラメータ値を返します。

JSONオブジェクトの説明

1）LexMaps - パラメータ値のアクティブ、デフォルト、およびユーザー設定の語彙マップを含むオブジェクト。

```
{
 "active": LexMap, //Active (current) values of configuration parameters
 "hardcoded": LexMap, //Default values of configuration parameters
 "master": LexMap //Configuration parameter values set
 //by the user
}
```

これらの各フィールドは、次にLexOptionオブジェクトの配列が格納されるLexMapオブジェクトです。

2) LexMap - パラメータの語彙マップを含むオブジェクト。

```
{
 "option": LexOption[] //Array of configuration options
}
```

3) LexOption - 単一のパラメータまたは設定のセクション(パラメータのグループ)を含むオブジェクト。

```
{
 "key": string, //Name of the option (configuration parameter/section)
 *"value": LexValue, //If this option is a single parameter
 *"map": LexMap //If this option is a section
}
```

アスタリスク(*)のパラメータは任意です。

LexOptionオブジェクトは、Dr.Web for UNIX Mail Serversの設定のセクションまたは単一のパラメータを表します。このオブジェクトには、常にセクションの名前または単一のパラメータの名前に対応するkeyフィールドがあります。これに加えて、このオブジェクトが表すもの(単一のパラメータまたはセクション)に応じて、valueフィールド(単一のパラメータを表す場合)またはmapフィールド(セクションを表す場合)もあります。セクションもまた、LexMapタイプのオブジェクトです。一方、単一のパラメータの値は、パラメータの値を文字列形式で指定するitemフィールドを含むLexValueタイプのオブジェクトです。

4) LexValue - パラメータに割り当てられた値の配列を含むオブジェクト。

```
{
 "item": string[] //Array of parameter values
}
```

set_lexmapコマンドは、その入力としてLexMapオブジェクトを受け取ります。これには、値を新しい値に変更するか、デフォルトにリセットするすべてのパラメータを含める必要があります。デフォルト値にリセットするパラメータには、valueフィールドを含めないでください。ユーザーがset_lexmapコマンドで指定した語彙マップに記載されていないパラメータは変更されません。set_lexmapコマンドは、その実行の結果として、コマンドで指定されたすべてのパラメータの変更結果を含むSetOptionResultオブジェクトを返します。

5) SetOptionResult - itemフィールドにパラメータの変更結果の配列を含むオブジェクト。

```
{
 "item": SetOptionResultItem[] //Array of results
}
```

このオブジェクトには、コマンドで指定されたすべてのパラメータの変更結果を表すSetOptionResultItemオブジェクトの配列が含まれています。

6) SetOptionResultItem - あるパラメータの値を変更することに関する情報を含むオブジェクト。

```
{
 "option": string, //Name of the parameter
 "result": string, //Result of changing the value (error code)
 *"lower_limit": string, //The lowest permitted value
 *"upper_limit": string //The highest permitted value
}
```

アスタリスク(*)のパラメータは任意です。

optionフィールドには、アクションが適用されたパラメータの名前が含まれており、resultフィールドには、このパラメータの値を変更しようとした結果が含まれています。新しい値がパラメータに正常に割り当てられた

場合、このフィールドにはEC_OKが含まれます。エラーの場合（このフィールドがEC_OKに等しくない場合）、このオブジェクトには、このパラメータの最大許容値と最小許容値を保持するlower_limitフィールドとupper_limitフィールドをオプションで含めることができます。

7）StartUpdateオブジェクトには、開始された更新プロセスに関するデータが含まれます。

```
{
 "start_update":
  {
    "attempt_id" : number //Identifier of a launched updating process
  }
}
```

8）ESConnectionオブジェクトには、開始された更新プロセスに関するデータが含まれます。

```
{
 *"server": string,    //<Host address>:<port> (without the http/https
prefix)
 "certificate": string, //Base64 server key
 *"newbie": boolean,   //False by default
 *"login": string,     //User name
 *"password": string   //Password
}
```

アスタリスク（*）のパラメータは任意です。

パラメータloginとpasswordは、newbie = trueの場合にのみ指定されます。
接続前に、集中管理サーバーから証明書ファイルをダウンロードし、次のコマンドを実行します。

```
$ cat certificate.pem |base64
```

このコマンドを実行して得られた文字列がcertificateのパラメータ値として使用されます。

9）BaseInfoResultオブジェクトには、ダウンロードしたウイルスベースのデータが含まれます。

```
{
 "vdb_base_stamp" : number //Timestamp of the base
 "vdb_bases" : VirusBaseInfo[] //Detailed information upon the base
}
```

10）VirusBaseInfoオブジェクトには、各ウイルスベースに関する情報が含まれます。

```
{
  "path" : string //Path to the base file
  "virus_records" : number //The number of records in the base
  "version" : number //Base version
  "timestamp" : number //Base timestamp
  "md5" : string //base MD5-hash
  "load_result" : string //The result of downloading the base (EC_OK if
the base has been downloaded successfully)
  *"sha1" : string - base SHA1-hash
}
```

アスタリスク（*）のパラメータは任意です。

11）IdpassRequestオブジェクトには、パスワードで保護されたアーカイブに関するデータが含まれます。

```
{
 "id": string,      //Identifier from the letter
 *"secret": string //Secret word (optional field)
}
```

アスタリスク(*)のパラメータは任意です。

secretフィールドが指定されておらず、パスワードタイプがPlainでない場合は、設定パラメータMailD.RepackPasswordの値が使用されます。パスワードタイプがPlainの場合は、エラー(Errorオブジェクト)が返されます(EC_INVALID_ARGUMENT)。

4. オブジェクトのスキャン

オブジェクトをスキャンするためのAPIコマンド:

APIコマンド	説明
データスキャン(Dr.Web Network Checkerコンポーネント呼び出しを使用)	
scan_request	アクション: 必要なパラメータを使用してデータをスキャンする接続(*endpoint*)の順序。 URI: /api/10.2/scan_request **HTTP**メソッド: POST 入力パラメータ: (SCS cookie)、ScanOptionsオブジェクト 正常に実行された結果: ScanEndpointオブジェクト、(SCS cookie)
scan_endpoint	アクション: 作成された*endpoint*接続でのデータスキャン(ファイル本体など)の起動。 URI: /api/10.2/scan_endpoint/*<endpoint>* **HTTP**メソッド: POST 入力パラメータ: (SCS cookie)、検証可能なデータ 正常に実行された結果: ScanResultオブジェクト、(SCS cookie)
scan_path	アクション: 指定されたパスにあるファイルまたはディレクトリをスキャンします。 URI: /api/10.2/scan_path **HTTP**メソッド: POST 入力パラメータ: (SCS cookie)、ScanPathOptionsオブジェクト 正常に実行された結果: ScanPathResultオブジェクト、(SCS cookie)
scan_stat	アクション: スキャン統計の表示。 URI: /api/10.2/scan_stat

APIコマンド	説明
	HTTPメソッド：GET 入力パラメータ：（SCS cookie）、統計のフォーマット（JSONまたはCSV） 正常に実行された結果：ScanStatオブジェクト（JSON形式が選択されている場合）、（SCS cookie） CSV形式が選択されている場合、ScanStatのフィールドに対応するテーブルが返されます。

JSONオブジェクトの説明

1）`ScanOptions`は、ファイルスキャン用の*エンドポイント*を作成するために使用されるパラメータを含むオブジェクトです。

```
{
 "scan_timeout_ms": number, //A time-out to scan one file, in ms
 "cure": boolean, //Apply cure to infected file
 "heuristic_analysis": boolean, //Use heuristic analysis
 "packer_max_level": number, //Maximum nesting level for packed objects
 "archive_max_level": number, //Maximum nesting level for archives
 "mail_max_level": number, //Maximum nesting level for email messaged
 "container_max_level": number, //Maximum nesting level for other
compound objects (containers)
 "max_compression_ratio": number, //Maximum a compression value
 "min_size_to_scan" : number, //Minimal size of an object to be scanned
 "max_size_to_scan" : number,  //Maximum size of an object to be scanned
 "threat_hash" : boolean   //Return SHA1 and SHA256 of all threats
}
```

2）`ScanPathOptions`は、指定されたパスにあるファイルまたはディレクトリをスキャンするために使用されるパラメータを含むオブジェクトです。

```
{
 "path" : string //Absolute path to the file or the directory to be
scanned
    *"exclude_path" : string[] //List of the paths excluded from scanning
(it is allowed to use masks)
    *"scan_timeout_ms" : number //Scan timeout for an object
    *"archive_max_level" : number//Maximum nesting level for archived
objects
    *"packer_max_level" : number //Maximum nesting level for packed
objects
    *"mail_max_level" : number //Maximum nesting level for email messages
    *"container_max_level" : number//Maximum nesting level for other
compound objects (containters)
    *"max_compression_ratio" : number//Maximum compression value
    *"heuristic_analysis" : bool  //Use heuristic analysis (true by
default)
    *"follow_symlinks" : bool  //Follow symbolic links
    *"min_size_to_scan" : number //Minimal size of an object to be
scanned
    *"max_size_to_scan" : number //Maximal size of an object to be
```

```
scanned
    *"timeout_ms" : number - //Scan timeout for all objects
    *"threat_hash" : bool - //Return SHA1 and SHA256 of all threats
}
```

アスタリスク(*)のパラメータは任意です。

3) `ScanPathResult`は、指定されたパスにあるオブジェクトのスキャン結果を含むオブジェクトです。

```
{
 ScanPathResult:
    "results": ScanResult[] //Scan results
    *"error": string  //Error if the scanning process terminated (the
scanning timeout is expirer, for instance)
}
```

アスタリスク(*)のパラメータは任意です。

スキャンが成功した場合、レスポンスには`error`文字列は含まれません。

4) `ScanResult`は、スキャンの結果を含むオブジェクトです。

```
{
 ScanResult:
    "scan_report" : ScanReport //The information upon the threat found
    *"sha1" : string  //The SHA1 hash of the threat
    *"sha256" : string //The SHA256 hash of the threat
}
```

アスタリスク(*)のパラメータは任意です。

5) `ScanReport`は、脅威が検出されたファイルに関する情報を含むオブジェクトです。

```
{
 ScanReport:
"object" : string  //Name of the object scanned
    For a file  //The absolute path, for a nested object - the name of
the file
    Always points to temporary file when calling scan_endpoint
*"size" : number //Object size
*"compressed_size" : number //Object size before extraction
*"core_fingerprint" : string //Scan engine fingerprint
*"packer" : string[]  //The list of packers used to pack the object
*"compression_ratio" : number //Archive compression ratio
*"archive" : Archive //Information on the archive or container type, if
the object scanned was identified as an archive or a container
*"virus" : Virus[] //Viruses detected in the objects (if found)
*"item" : ScanReport[] //Reports on scanning of the nested objects (if
there were some)
*"error" : string  //Scanning error (if occured)
*"heuristic_analysis" : bool //Indicates if heuristic analysis was used
*"cured" : bool //The object was cured
*"cured_by_deletion" : bool //The object was deleted.
*"new_path" : string //The new path to the object renamed when being
cured
*"user_time" : number //Type spent for syscalls when scanning
```

```
*"system_time" : number //Time spent in the userspace
}
```

アスタリスク(*)のパラメータは任意です。

virusフィールドとerrorフィールドは、スキャン中に脅威が検出されず、エラーが発生しなかった場合には、存在しない可能性があります。scan_endpointを呼び出すためには、Dr.Web Network Checkerコンポーネントによってローカルサーバーファイルシステムに作成され、スキャンに関するデータを含み、scan_endpointリクエストの本文で送信される一時ファイルをscan_endpointフィールドで必ず指定します。

6) ScanEndpointは、ファイルスキャン用に作成された*エンドポイント*に関するデータを含むオブジェクトです。

```
{
 "endpoint": string //Unique identifier of the created endpoint
}
```

オブジェクト本体で返されるendpoint文字列は、scan_endpointコマンド(URIの一部)でファイルスキャンを開始するために使用されます。

7) VirusInfoは、検出された脅威に関する情報を含むオブジェクトです。

```
{
 "type": string, //Type of the detected threat
 "name": string  //Name of the threat
}
```

typeフィールド(脅威タイプ)は文字列SE_XXXです。

- SE_KNOWN_VIRUSは既知のウイルスです。
- SE_VIRUS_MODIFICATIONは既知のマルウェアの亜種です。
- SE_UNKNOWN_VIRUSは未知のウイルス(疑わしいオブジェクト)です。
- SE_ADWAREはアドウェアです。
- SE_DIALERはダイアラープログラムです。
- SE_JOKEはジョークプログラムです。
- SE_RISKWAREは潜在的に危険なプログラムです。
- SE_HACKTOOLはハッキングツールです。

8) Archiveは、アーカイブ、圧縮されたオブジェクト、メールメッセージ、およびその他のコンテナに関する情報を含むオブジェクトです。

```
{
 "type" : string - the type of the archive:
        "SE_ARCHIVE" - archive
        "SE_MAIL" - e-mail message
        "SE_CONTAINER" - other container
 "name" : string - archive format
}
```

9) ScanStatは、スキャン統計を含むオブジェクトです。

```
{
 "origin": string //The application by the request of which the scanning
```

```
was initialized
    #Counters for infected objects:
    "known_virus": number //Number of objects infected by known viruses
    "virus_modification": number //Number of objects infected by
modifications of known viruses
    "unknown_virus": number //Number of objects infected by unknown
viruses
    "adware": number //Number of objects with SE_ADWARE
    "dialer": number //Number of objects with SE_DIALER
    "joke": number //Number of objects with SE_JOKE
    "riskware": number //Number of objects with SE_RISKWARE
    "hacktool" : number //Number of objects with SE_HACKTOOL
    "cured": number //Number of cured threats
    "quarantined": number //Number of quarantined threats
    "deleted": number  //Number of deleted threats
}
```

5. 脅威のリストの管理

スキャン中またはファイルシステムモニター（SpIDer Guard）によって検出された脅威のリストを管理できるように、HTTP APIには次のコマンドが用意されています。

APIコマンド	説明
`threats`	アクション：検出されたすべての脅威のIDを一覧表示します。 URI: `/api/10.2/threats/` HTTPメソッド：GET 入力パラメータ：（SCS cookie） 正常に実行された結果：脅威IDの配列
`threat_info`	アクション：脅威に関する情報を脅威のIDである *<threat ID>* で取得します。 URI: `/api/10.2/threat_info/` *<threat ID>* HTTPメソッド：GET 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、FileThreatオブジェクト
`cure_threat`	アクション：脅威のIDである *<threat ID>* で指定された脅威の修復を試みます。 URI: `/api/10.2/cure_threat/` *<threat ID>* HTTPメソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト

APIコマンド	説明
delete_threat	アクション：脅威のIDである*<threat ID>*で指定された脅威を含むファイルを削除します。 URI: /api/10.2/delete_threat/*<threat ID>* **HTTP**メソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト
ignore_threat	アクション：脅威のIDである*<threat ID>*で指定された脅威を無視します。 URI: /api/10.2/ignore_threat/*<threat ID>* **HTTP**メソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト
quarantine_threat	アクション：脅威のIDである*<threat ID>*で指定された脅威を隔離します。 URI: /api/10.2/quarantine_threat/*<threat ID>* **HTTP**メソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト

指定されたアプリケーションで見つかったそれぞれの脅威には、一意の整数で表されたID*<threat ID>*があります。すべてのIDのリストはthreatsコマンドによって返されます。threat_info、cure_threat、delete_threat、ignore_threat、およびquarantine_threatコマンドでは、threatsコマンドが返すIDのみが許容されます。

アクション履歴を含む、指定された脅威に関するすべての情報は、threat_infoリクエストを使用して取得できます。情報はFileThreatオブジェクトとして返されます。

JSONオブジェクトの説明

1）FileThreatは、次のデータを含むオブジェクトです。

```
{
 "threat_id": number, //Threat identifier
 "detection_time": UNIXTime, //Time when the threat was detected
 "report": ScanReport, //Report about scanning the file
 "stat": FileStat, //Information about the file
 "origin": string, //Name of the component that detected the threat
 "origin_pid": number, //PID of the component that detected the threat
 "task_id": number, //Identifier of the scanning task
```

```
 //in the scan engine
 "history": ActionResult[] //History of actions applied to the threat (an
array)
}
```

reportフィールドにはScanReportオブジェクトが含まれます。statフィールドにはFileStatオブジェクトが含まれ、historyフィールドにはActionResultオブジェクト（ファイルに適用されたアクションの履歴）の配列が含まれます。

2）ScanReport - 脅威が検出されたファイルに関する情報を含むオブジェクト。

```
{
 "object": string, //File system object that contains the threat
 "size": number, //Size (in bytes) of the file that contains the threat
 "virus": VirusInfo[], //List of details about the found
 //threats
 *"error": string, //An error message
 "heuristic_analysis": bool //Flag that shows whether heuristic
 //analysis was used
}
```

アスタリスク（*）のパラメータは任意です。

virusフィールドは、検出されたすべての脅威に関する情報を含むVirusInfoオブジェクトの配列です。errorフィールドは、エラーが発生した場合にのみ表示されます。

3）FileStatは、ファイル統計を含むオブジェクトです。

```
{
 "dev": number, //Device containing the file
 "ino": number, //The file inode number
 *"size": number, //Size of the file
 *"uid": User, //User ID of the file's owner
 *"gid": Group, //Group ID of the owning group
 *"mode": number, //The mode of access to the file
 *"mtime": UNIXTime, //Date/time when the file was last modified
 *"ctime": UNIXTime //Date/time when the file was created
 *"rsrc_size": number, //
 *"finder_info": string, //
 *"ext_finder_info": string, //
 *"uchg": string, //
 *"volume_name": string, //Volume name
 *"volume_root": string, //Root (mount point) of the volume
 *"xattr": XAttr[] //Extended information about the file
}
```

アスタリスク（*）のパラメータは任意です。

xattrフィールドには、XAttrオブジェクトの配列が含まれています。このオブジェクトには、nameおよびvalueの2つの*文字列*タイプのフィールドがあります。uidフィールドとgidフィールドにはそれぞれユーザーオブジェクトとグループオブジェクトが含まれており、これらのオブジェクトにはそれぞれファイルの所有者とファイルを所有しているグループに関する情報が含まれています。これらのオブジェクトにはそれぞれ次の2つのフィールドがあります。

- uid（gid） - ユーザー（グループ）のID（数値）
- username（groupname） - ユーザー（グループ）の名前（文字列）

4) ActionResultは、ファイルに適用されたアクションとその結果に関する情報を含むオブジェクトです。

```
{
 "action": string, //The action applied
 "action_time": UNIXTime, //Date/time when the action was applied
 "result": string, //Result of applying the action
 "cure_report": ScanReport //Report about applying the action
}
```

cure_threat、delete_threat、ignore_threat、およびquarantine_threatコマンドは、正常に実行されると空のオブジェクトを返します。リクエストされたアクションが失敗した場合は、Errorオブジェクトが返されます。

6. 隔離の管理

隔離オブジェクトを管理するため、HTTP APIには次のコマンドが用意されています。

APIコマンド	説明
quarantine	アクション：隔離されたオブジェクトのIDの一覧を表示します。 URI: /api/10.2/quarantine/ HTTPメソッド：GET 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、QuarantineIdオブジェクト（隔離内のオブジェクト）の配列
qentry_info	アクション：隔離オブジェクトのIDである *<entry ID>* で指定された隔離オブジェクトに関する情報を取得します。 URI: /api/10.2/qentry_info/ *<entry ID>* HTTPメソッド：GET 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、QEntryオブジェクト
cure_qentry	アクション：隔離オブジェクトのIDである *<entry ID>* で指定された隔離オブジェクトの修復を試みます。 URI: /api/10.2/cure_qentry/ *<entry ID>* HTTPメソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト

APIコマンド	説明
delete_qentry	アクション：隔離オブジェクトのIDである *<entry ID>* で指定された隔離オブジェクトを削除します。 URI: /api/10.2/delete_qentry/ *<entry ID>* **HTTP**メソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト
restore_qentry	アクション：隔離オブジェクトのIDである *<entry ID>* で指定された隔離オブジェクトを元の場所に復元します。 URI: /api/10.2/restore_qentry/ *<entry ID>* **HTTP**メソッド：POST 入力パラメータ：（SCS cookie） 正常に実行された結果：（SCS cookie）、空のオブジェクト

それぞれの隔離オブジェクトには一意のIDがあります。QuarantineIdとして表されるIDのリストは、quarantineコマンドによって返されます。IDはchunk_idとentry_idの2つの部分で構成されます。

JSONオブジェクトの説明

1）QuarantineIdは、隔離オブジェクトの2つの部分からなるIDの両方の部分を含むオブジェクトです。

```
{
 "chunk_id": string,
 "entry_id": string
}
```

これら2つのフィールドが一体となって隔離オブジェクトのIDを構成します。qentry_info、cure_qentry、delete_qentry、またはrestore_qentryコマンドを使用して隔離オブジェクトにアクションを適用するには、隔離オブジェクトの一般的なIDである *<entry ID>* を *<entry_id>@<chunk_id>* の形式で指定する必要があります。qentry_infoコマンドを使用すると、指定されたIDとともに隔離オブジェクトに関する詳細情報を取得できます。このコマンドはQEntryタイプのオブジェクトを返します。

2）QEntry - 隔離オブジェクトに関する情報を含むオブジェクト。

```
{
 "entry_id": string, //Parts of the identifier of
 *"chunk_id": string, //this quarantined object
 *"quarantine_dir": string, //Quarantine directory
 "restore_path": string, //path where the quarantined
 //object will be restored
 "creation_time": number, //Date/time of moving to quarantine
 //(in UNIX time)
```

```
 "report": ScanReport, //Report about scanning the object
 //(see ScanReport described above)
 "stat": FileStat, //Statistical information about the file
 //(see FileStat described above)
 *"history": QEntryOperation[], //History of operations performed on the
object
 *"who": RemoteUser, //The remote owner of the file (if
 //the file was quarantined from a file server
 //storage)
 *"detection_time": number, //Date/time of detecting the threat
 *"origin": string, //Component that detected the threat
}
```

アスタリスク(*)のパラメータは任意です。

reportフィールドにはScanReportオブジェクトが含まれます。statフィールドにはFileStatオブジェクトが含まれ、historyフィールドには、隔離オブジェクトに適用されたアクションの履歴が含まれます。各アクションエントリはQEntryOperationオブジェクトによって記述されます。オプションのwhoフィールドには、削除されたユーザーに関する情報がRemoteUserオブジェクトの形式で含まれます。

3) QEntryOperationは、隔離オブジェクトに適用された操作に関するデータを含むオブジェクトです。

```
{
 "action": string, //Operation performed on the object
 //(see the possible values below)
 "action_time": number, //Date/time when the operation was performed
(UNIX Time)
 "result": string, //Error when trying to perform the operation (a code
 //EC_XXX)
 *"restore_path": string, //path for restoring the quarantined object
 //(if action = "QENTRY_ACTION_RESTORE")
 *"rescan_report": ScanReport //Report about rescanning (if
 //action = "QENTRY_ACTION_RESCAN")
}
```

アスタリスク(*)のパラメータは任意です。

actionフィールドには、以下の値を指定できます。

- QENTRY_ACTION_DELETEは、隔離オブジェクトの削除を試みます。
- QENTRY_ACTION_RESTOREは、隔離オブジェクトの復元を試みます。
- QENTRY_ACTION_RESCANは、隔離オブジェクトの再スキャンを試みます。
- QENTRY_ACTION_CUREは、隔離オブジェクトの修復を試みます。

4) RemoteUserは、ファイルを所有するリモートユーザーに関する情報を含むオブジェクトです(ファイルがファイルサーバーストレージから隔離に再配置された場合)。

```
{
 *"ip": string, //IP-address of the user
 *"user": string, //User name
 *"domain": string //Domain of the user
}
```

アスタリスク(*)のパラメータは任意です。

cure_qentry、delete_qentry、restore_qentryコマンドの実行が成功すると、空のオブジェクトが返されます。隔離オブジェクトに対して要求された操作がエラーで終了した場合（たとえば、ファイルを復元できなかった場合）、空のオブジェクトの代わりにErrorオブジェクトが返されます。

7. HTTP APIの使用例

HTTP APIの動作をテストするには、curlユーティリティを使用します。API呼び出しの一般的なフォーマットは以下のとおりです。

```
$ curl https://<HTTPD.AdminListen>/<HTTP API URI> -k -X <HTTP method name>
 [-H 'Content-Type: application/json' --data-binary '@<file of the JSON object>']
 [-c <cookie file> [-b <cookie file>]] [> <file of the result>]
```

- -kオプションでは、curlがSSL証明書を確認しないように指定します。
- -Xオプションでは、使用するHTTPメソッド（GETまたはPOST）を指定します。
- -Hオプションは、Content-Type: application/jsonヘッダーの追加に使用します。
- --data-binary（または-d）オプションは、テキストファイルに保存されたJSONオブジェクトをリクエストに追加するために使用します。
- SCSを使用して承認を得る場合、送受信したSCS cookieを含むファイルをそれぞれ-bと-cのパラメータで指定する必要があります。

  curlオプションの詳細な説明については、manページを参照してください（curl --helpまたはman curlコマンドを実行してください）。

1. ユーザー名とパスワード（SCS用）を指定して、クライアントを認証および承認する。

   JSON形式のAuthOptionsオブジェクトがあらかじめuser.jsonというファイルに書き込まれている必要があります。例：

   {"user":"*<ユーザー名>*","password":"*<パスフレーズ>*"}。

   リクエスト：

   ```
   $ curl https://127.0.0.1:4443/api/10.2/login -k -X POST -H 'Content-Type:
   application/json' --data-binary '@user.json' -c cookie.file
   ```

   レスポンス：

   ```
   HTTP/1.0 200 OK
   Content-Type: application/json
   Content-Length: 2
   Set-Cookie:
   DWToken=6QXy4wn_JGov9A1GohWP_kvMK3dN6ccKegjNgKcmHpb_AqSrHg9cNX_yFJhxPDgr|
   MTQ2Mjg3Mzg4NQ==|cWd4Ow==|GywBUVOhU4w2LF_BKT5frg==|
   kR_rip5nrpxWjJ2dfZ7Xfmvi3rE=; Secure; HttpOnly; Max-Age: 900; Path=/
   Pragma: no-cache

   {}
   ```

   Set-Cookieヘッダーフィールドには、それ以降のHTTP APIへのすべてのリクエストで使用する必要があるSCS cookieが含まれています。認証と承認が成功した場合、レスポンスの本文には空のオブジェクトが含まれています。ユーザーが承認されなかった場合は、次のようなErrorオブジェクトが返されます。

```
HTTP/1.0 403 Forbidden
Content-Type: application/json
Content-Length: 35
Pragma: no-cache

{"error":{"code":"EC_AUTH_FAILED"}}
```

2. *ID が 1* である脅威に関する情報を取得する。

   リクエスト:

```
$ curl https://127.0.0.1:4443/api/10.2/threat_info/1 -k -X GET -c
cookie.file -b cookie.file
```

   レスポンス:

```
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 574
Set-Cookie: DWToken=<...>;
 Secure; HttpOnly; Max-Age: 900; Path=/
Pragma: no-cache

{"threat_id":1,"detection_time":1462881660,
"report":{"object":"/sites/site1/eicar.com.txt","size":68,"packer":[],
"virus":[{"type":"SE_KNOWN_VIRUS","name":"EICAR Test File (NOT a
Virus!)"}],
"heuristic_analysis":true,"core_fingerprint":"0D2DD5A869DAB7AE354153A4D5F
70F45",
"item":[],"log":[],"user_time":0,"system_time":0},"stat":
{"dev":2049,"ino":898,
"size":68,"uid":{"uid":1000,"username":"user"},"gid":
{"gid":1000,"groupname":"user"},
"mode":33204,"mtime":1441028214,"ctime":1460738554,"xattr":[]},
"origin":"APP_COMMAND_LINE_TOOL","origin_pid":2726,"task_id":1,"history":
[]}
```

3. *ID が 1* である脅威を隔離へ移動する。

   リクエスト:

```
$ curl -v -c cookie.jar -b cookie.jar -k -X POST -H 'Content-
Type:application/json'
https://127.0.0.1:4443/api/10.2/quarantine_threat/1
```

   レスポンス:

```
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 2
Set-Cookie: DWToken=<...>; Secure; HttpOnly; Max-Age: 900; Path=/
Pragma: no-cache

{}
```

4. 指定された **ID** を持つ隔離オブジェクトに関する情報を表示する。

リクエスト:

```
$ curl -v -k -X GET -c cookie.jar -b cookie.jar
https://127.0.0.1:4443/api/10.2/qentry_info/3070d3ce-7b6e-4143-9d9f-
89ba3473a781@801:2108d
```

レスポンス:

```
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 781
Set-Cookie: DWToken=<...>; Secure; HttpOnly; Max-Age: 900; Path=/
Pragma: no-cache

{"entry_id":"3070d3ce-7b6e-4143-9d9f-
89ba3473a781","chunk_id":"3830313A3231303864",
"quarantine_dir":"2F686F6D652F757365722F2E636F6D2E64727765622E71756172616
E74696E65",
"restore_path":"2E2E2F7473742F65696361722E636F6D2E747874","creation_time"
:1462888884,
"report":{"object":"/home/user/tst/eicar.com.txt","size":68,"packer":[],
"virus":[{"type":"SE_KNOWN_VIRUS","name":"EICAR Test File (NOT a
Virus!)"}],
"heuristic_analysis":true,"core_fingerprint":"467CD4C6D423C55448B71CD5B81
52776",
"item":[],"log":[],"user_time":0,"system_time":0},"stat":
{"dev":2049,"ino":898,
"size":68,"uid":{"uid":1000,"username":"user"},"gid":
{"gid":1000,"groupname":"user"},
"mode":33204,"mtime":1441028214,"ctime":1462888421,"xattr":[]},"history":
[],
"detection_time":1462888667,"origin":"APP_COMMAND_LINE_TOOL"}
```

5. **設定を変更する**: Dr.Web CloudDを無効にする。

   JSON形式のLexMapオブジェクトがあらかじめ`lexmap_ls_off.json`というファイルに書き込まれている必要があります。

   ```
   {"option":[{"key":"Root","map":{"option":
   [{"key":"UseCloud","value":{"item":["no"]}}]}}]}
   ```

   リクエスト:

   ```
   $ curl -v -k -c cookie.jar -b cookie.jar -X POST -H 'Content-Type:
   application/json' --data-binary '@lexmap_ls_off.json'
   https://127.0.0.1:4443/api/10.2/set_lexmap
   ```

   レスポンス:

   ```
   HTTP/1.0 200 OK
   Content-Type: application/json
   Content-Length: 58
   Set-Cookie: DWToken=<...>; Secure; HttpOnly; Max-Age: 900; Path=/
   Pragma: no-cache

   {"item":[{"option":"Root.UseCloud","result":"EC_OK"}]}
   ```

6. **設定を変更する**: Dr.Web CloudDを有効にする。

JSON形式のLexMapオブジェクトが`lexmap_ls_on.json`という名前のファイルに保存されている必要があります。

```
{"option":[{"key":"Root","map":{"option":
[{"key":"UseCloud","value":{"item":["yes"]}}]}}]}
```

リクエスト:

```
$ curl -v -k -c cookie.jar -b cookie.jar -X POST -H 'Content-Type:
application/json' --data-binary '@lexmap_ls_on.json'
https://127.0.0.1:4443/api/10.2/set_lexmap
```

レスポンス:

```
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 58
Set-Cookie: DWToken=<...>; Secure; HttpOnly; Max-Age: 900; Path=/
Pragma: no-cache

{"item":[{"option":"Root.UseCloud","result":"EC_OK"}]}
```

Dr.Web SNMPD

Dr.Web SNMPDは、SNMPプロトコルを実行するモニタリングシステムにDr.Web for UNIX Mail Serversを統合するよう設計されたSNMPエージェントです。この統合により、Dr.Web for UNIX Mail Serversのコンポーネントのステータスを追跡したり、脅威の検出と駆除に関する統計を収集したりできます。このエージェントのサポートにより、モニタリングシステムまたはSNMPマネージャーに以下の情報が提供されます。

- 任意のDr.Web for UNIX Mail Serversコンポーネントのステータス。
- 検出されたさまざまなタイプの脅威の数（Dr.Web分類に対応）。

さらに、エージェントは、脅威を検出したときと検出された脅威の駆除が失敗したときにSNMPトラップ通知を送信します。エージェントはSNMPプロトコルバージョン2cおよび3をサポートします。

エージェントが送信できる情報の説明は、Doctor Webによって作成されたMIB（*管理情報ベース*）の特別なセクションに格納されています。UNIX系オペレーティングシステム用にDr.Webが定義したMIBセクションには、以下の情報が指定されています。

1. 脅威の検出と駆除、Dr.Web for UNIX Mail Serversコンポーネントに関連するエラーに関するSNMPトラップ通知のフォーマット。
2. Dr.Web for UNIX Mail Servers操作の統計。
3. Dr.Web for UNIX Mail Serversコンポーネントのステータス。

SNMPプロトコルを介して取得できる情報の詳細については、対応するセクションを参照してください。

動作原理

このセクションの内容

- 概要
- システムSNMPエージェントとの統合

概要

デフォルトでは、このコンポーネントはDr.Web for UNIX Mail Serversの起動時に自動的に実行されます。実行されると、このコンポーネントはMIB Dr.Webで記述されている構造に従ってデータを構造化し、外部SNMPマネージャーからのデータ受信要求を待機します。このコンポーネントはDr.Web ConfigD設定デーモンから、Dr.Web for UNIX Mail Serversのコンポーネントのステータスに関する情報と、検出された脅威に関する通知を直接受信します。

脅威は、Dr.Web for UNIX Mail Serversコンポーネントによるスキャン中にスキャンエンジンによって検出されます。何らかの脅威が検出されると、（この脅威の種類に）該当するカウントが1つ増え、通知を受信できるすべてのSNMPマネージャーは、検出された脅威について通知するSNMPトラップを受け取ります。

カウンターの収集値（たとえば、検出された脅威のカウンター）は、Dr.Web SNMPDの起動の合間は保存されません。したがって、Dr.Web SNMPDが何らかの理由（Dr.Web for UNIX Mail Serversの通常の再起動を含む）で再起動されると、収集されたカウンターの値は0にリセットされます。

システムSNMPエージェントとの統合

メインシステムのSNMPエージェントsnmpd（net-snmp）がすでにサーバー上で動作している場合にDr.Web SNMPエージェントを正しく動作させるには、Dr.Web MIBブランチを介したsnmpdからDr.Web SNMPへのSNMPリクエスト送信を設定します。そのためには、次の行を追加してsnmpd設定ファイル（GNU/Linuxの場合は、通常`/etc/snmp/snmpd.conf`）を編集します。

```
proxy -v <version> -c <community> <address>:<port> <MIB branch>
```

各パラメータは次のとおりです。

- *<version>* - 使用中のSNMPバージョン（`2c`、`3`）
- *<community>* - Dr.Web SNMPDによって使用される「コミュニティストリング」
- *<address>:<port>* - Dr.Web SNMPDによってリッスンされるネットワークソケット
- *<MIB branch>* - Dr.Webが使用する変数とSNMPトラップの説明を含むMIBブランチのOID（OIDは`.1.3.6.1.4.1.29690`）

Dr.Web SNMPエージェントをデフォルト設定で使用している場合、追加する行は以下のようになります。

```
proxy -v 2c -c public localhost:50000 .1.3.6.1.4.1.29690
```

この場合、ポート161はシステムの標準snmpdによって使用されるため、ListenAddressパラメータでDr.Web SNMPD用に別のポートを指定する必要があります（この例では、`50000`）。

コマンドライン引数

OSのコマンドラインからDr.Web SNMPDを起動するには、次のコマンドを使用します。

```
$ <opt_dir>/bin/drweb-snmpd [<parameters>]
```

Dr.Web SNMPDは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-snmpd --help
```

このコマンドは、Dr.Web SNMPDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます（原則として、OSの起動時）。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-snmpd`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Servers の統合された設定ファイルの`[SNMPD]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel`	コンポーネントのログレベル。

パラメータ	説明
{logging level}	パラメータの値が指定されていない場合は、[Root] セクションの DefaultLogLevel パラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行ファイルへのパス。 デフォルト値：*<opt_dir>*/bin/drweb-snmpd • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-snmpd。 • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-snmpd。
Start *{Boolean}*	Dr.Web ConfigD 設定デーモンによってコンポーネントを起動するかどうかを指定します。 このパラメータに Yes 値を指定すると、設定デーモンはただちにコンポーネントを開始します。また、No 値を指定すると、設定デーモンはただちにコンポーネントを終了します。 デフォルト値：No
RunAsUser *{UID / user name}*	コンポーネントの実行に必要な権限を有するユーザー。このユーザー名は、ユーザーの UID またはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまり、数字の UID に似ている場合）は、「name:」というプレフィックスを付けて指定します。たとえば、RunAsUser = name:123456 です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：drweb
ListenAddress *{address}*	クライアント接続（SNMP マネージャー）を待機する Dr.Web SNMPD がリッスンするアドレス（IP アドレスとポート）。 snmpd とのインタラクションには、標準ポート（161）とは異なるポートの指定と、snmpd にプロキシ用の設定を行う必要があります。 デフォルト値：127.0.0.1:161
SnmpVersion *{V2c / V3}*	SNMP プロトコルの現在のバージョン（*SNMPv2c* または *SNMPv3*）。 デフォルト値：V2c
V3EngineId *{string}*	*SNMPv3 のエンジン ID* の識別子（文字列）（RFC 3411 に準拠）。 デフォルト値：800073FA044452574542

パラメータ	説明
`TrapReceiver` *{address list}*	Dr.Web for UNIX Mail Servers のコンポーネントが脅威を検出した後に、Dr.Web SNMPD によって *SNMP トラップ* 通知が送信されるアドレスのリスト（IP アドレスとポート）。 リストをパラメータ値として指定できます。リストの値は、コンマ（引用符内の各値）で区切る必要があります。パラメータはセクションで複数回指定できます（この場合、そのすべての値が 1 つのリストにまとめられます）。 例：ソケット 192.168.0.1:1234 および 10.20.30.45:5678 をリストに追加します。 1. 設定ファイルに値を追加します。 • 1 つの文字列に 2 つの値： セクション [SNMPD] `TrapReceiver = "192.168.0.1:1234", "10.20.30.45:5678"` • 2 つの文字列（文字列ごとに 1 つの値）： `[SNMPD]` `TrapReceiver = 192.168.0.1:1234` `TrapReceiver = 10.20.30.45:5678` 2. `drweb-ctl cfset` コマンドを使用して値を追加します。 `# drweb-ctl cfset SNMPD.TrapReceiver -a 192.168.0.1:1234` `# drweb-ctl cfset SNMPD.TrapReceiver -a 10.20.30.45:5678` デフォルト値：*（未設定）*
`V2cCommunity` *{string}*	Dr.Web の MIB 変数 が読み取りアクセスされたときに、SNMP マネージャー（*SNMPv2c* プロトコル）を認証するための文字列「SNMP read community」。 `SnmpVersion = V2c` の場合、このパラメータが使用されます。 デフォルト値：`public`
`V3UserName` *{string}*	Dr.Web の MIB 変数 が読み取りアクセスされたときに、SNMP マネージャー（*SNMPv3* プロトコル）を認証するためのユーザー名。 `SnmpVersion = V3` の場合、このパラメータが使用されます。 デフォルト値：`noAuthUser`
`V3Auth` *{SHA(<pwd>) | MD5(<pwd>) | None}*	Dr.Web の MIB 変数 が読み取りアクセスされたときに、SNMP マネージャー（*SNMPv3* プロトコル）を認証する方法。

パラメータ	説明
	使用可能な値： • `SHA(<PWD>)` - パスワードのSHAハッシュが使用されます（*<PWD>*文字列）。 • `MD5(<PWD>)` - パスワードのMD5ハッシュが使用されます（*<PWD>*文字列）。 • `None` - 認証は無効になります。 ここで、*<PWD>*はプレーンテキストのパスワードです。 コマンドラインからパラメータ値を指定する場合、シェルによってはスラッシュ記号 \ を使用した角括弧のエスケープを必要とする場合があります。 例： 1. 設定ファイル内のパラメータ値： `V3Auth = MD5(123456)` 2. `drweb-ctl cfset`コマンドを使用して、コマンドラインから同じパラメータ値を指定する場合： `drweb-ctl cfset SNMPD.V3Auth MD5\(123456\)` `SnmpVersion = V3`の場合、このパラメータが使用されます。 デフォルト値：`None`
`V3Privacy` *{DES(<secret>) | AES128(<secret>) | None}*	SNMPメッセージの暗号化方式（*SNMPv3*プロトコル）。 使用可能な値： • `DES(<secret>)` - DES暗号化アルゴリズム。 • `AES128(<secret>)` - AES128暗号化アルゴリズム。 • `None` - SNMPメッセージは暗号化されません。 ここで、*<secret>*は、マネージャーとエージェントが共有するシークレットキーです（*プレーンテキスト*）。 コマンドラインからパラメータ値を指定する場合、シェルによってはスラッシュ記号 \ を使用した角括弧のエスケープを必要とする場合があります。 例： 1. 設定ファイル内のパラメータ値： `V3Privacy = AES128(supersecret)` 2. `drweb-ctl cfset`コマンドを使用して、コマンドラインから同じパラメータ値を指定する場合： `drweb-ctl cfset SNMPD.V3Privacy AES128\(supersecret\)` `SnmpVersion = V3`の場合、このパラメータが使用されます。

パラメータ	説明
	デフォルト値：None

SNMPモニタリングシステムとの統合

Dr.Web SNMPエージェントは、SNMPプロトコル、バージョン*2c*または*3*を使用するモニタリングシステムのデータプロバイダーとして機能します。制御に使用できるデータのリストとデータ構造は、Dr.Web for UNIX Mail Serversに付属のDr.Web MIB記述ファイルDRWEB-SNMPD-MIB.txtに記述されています。このファイルは、ディレクトリ*<opt_dir>*/share/drweb-snmpd/mibsにあります。

簡単に設定できるように、このコンポーネントには一般的なモニタリングシステム用の設定テンプレートが付属しています。

- Munin
- Nagios
- Zabbix

モニタリングシステム用のカスタマイズテンプレートは、*<opt_dir>*/share/drweb-snmpd/connectorsディレクトリにあります。

Muninモニタリングシステムとの統合

Muninモニタリングシステムには、モニタリング対象ホストにローカルに存在するクライアントmunin-nodeから統計を収集する中央サーバー（マスター）muninが含まれています。サーバーのリクエストに応じて、各モニタリングクライアントは、サーバーに転送されたデータを提供する*プラグイン*（*プラグイン*）を起動することによって、モニタリング対象ホストの動作に関するデータを収集します。

Dr.Web SNMPDとMuninモニタリングシステム間の接続を可能にするために、すぐに利用できるmunin-node用プラグインDr.Webが用意されています。このプラグインは*<opt_dir>*/share/drweb-snmpd/connectors/munin/pluginsディレクトリにあります。このプラグインは、次の2つのグラフの作成に必要なデータを収集します。

- 検出された脅威の数。
- ファイルのスキャン統計情報
- メールメッセージのスキャン統計（Dr.Web MailDコンポーネントでのみメール統計を取得できます。Dr.Web MailDはには含まれていません）。

これらのプラグインは、SNMPプロトコルバージョン1、2c、3をサポートしています。これらのテンプレートプラグインに基づいて他のプラグインを作成すれば、Dr.Web SNMPD経由でDr.Web for UNIX Mail Serversのコンポーネントのステータスをポーリングできます。

<opt_dir>/share/drweb-snmpd/connectors/muninディレクトリには、以下のファイルがあります。

ファイル	説明
plugins/snmp__drweb_malware	ホスト上のDr.Web for UNIX Mail Serversによって検出された脅威の数を収集するために、SNMP経由で

ファイル	説明
	Dr.Web SNMPDをポーリングするためのmunin-nodeプラグイン。
plugins/snmp__drweb_filecheck	ホスト上のDr.Web for UNIX Mail Serversによってスキャンされたファイルの統計を収集するために、SNMP経由でDr.Web SNMPDをポーリングするためのmunin-nodeプラグイン。
plugins/snmp__drweb_maild_multi	ホスト上のDr.Web for UNIX Mail Serversによってスキャンされたメールメッセージの統計を収集するために、SNMP経由でDr.Web SNMPDをポーリングするために使用されるmunin-nodeプラグイン。 このプラグインは、Muninバージョン1.4以上で利用可能な機能である*multigraph*を使用します。
plugin-conf.d/drweb.cfg	Dr.Webプラグインのmunin-node設定の例（環境変数用）。

Muninにホストを接続する

この説明では、Muninモニタリングシステムがモニタリングサーバーにすでにデプロイされており、モニタリング対象ホストにはDr.Web SNMPDがインストール済みでmunin-nodeとともに機能している（コンポーネントはsnmpdとともにプロキシモードで機能する）と想定しています。

1. モニタリング対象ホストの設定
   - snmp__drweb_*ファイルを、プラグインライブラリmunin-nodeがあるディレクトリにコピーします（ディレクトリはOSによって異なります）。たとえば、Debian/Ubuntuオペレーティングシステムであれば、パスは/usr/share/munin/pluginsとなります。
   - 提供されているDr.Webプラグインを接続して、munin-nodeを設定します。これを行うには、munin-nodeと一緒に配布されるmunin-node-configureユーティリティを使用します。

     次は、コマンドの例です。

     ```
     $ munin-node-configure --shell --snmp localhost
     ```

     端末画面に、プラグインに必要なシンボリックリンクを作成するためのコマンドのリストを表示します。コピーして、コマンドラインで実行します。指定されたコマンドでは、次のことを前提としています。

     1) munin-nodeは、Dr.Web SNMPDがインストールされているのと同じホストにインストールされている。インストール先が異なる場合は、localhost値ではなく、モニタリング対象ホストの適切なFQDNまたはIPアドレスを指定してください。
     2) Dr.Web SNMPDはSNMPバージョン2cを使用する。他のバージョンの場合は、munin-node-configureコマンドで適切なSNMPバージョンを指定してください。このコマンドには、プラグインを柔軟に設定するための引数がいくつかあります。たとえば、SNMPプロトコルのバージョン、モニタリング対象ホストでSNMPエージェントがリッスンしているポート、*コミュニティストリング*の実際の値などを指定できます。必要に応じて、munin-node-configureコマンドのマニュアルを参照してください。
   - 必要であれば、munin-node用にインストールしたDr.Webプラグインを実行する環境のパラメータ値を定義（または再定義）します。環境パラメータとして、値*コミュニティストリング*が使用されます。これは、SNMP

エージェントなどが使用するポートです。これらのパラメータは、ファイル/etc/munin/plugin-conf.d/drwebに定義する必要があります（必要に応じて作成します）。このファイルの例として、提供されているファイルdrweb.cfgを使用します。

- munin-node設定ファイル（munin-node.conf）で、モニタリング対象のパラメータの値を受け取るためにmuninサーバー（マスター）をmunin-nodeに接続できるホストのすべてのIPアドレスを含めるための正規表現を指定します。たとえば、次のようになります。

```
allow ^10\.20\.30\.40$
```

この場合、IPアドレス10.20.30.40のみがホストパラメータを受信できます。

- たとえば、次のコマンドを使用してmunin-nodeを再起動します。

```
# service munin-node restart
```

2. Muninサーバー（マスター）の設定

モニタリング対象ホストのアドレスと識別子をMunin設定ファイルmunin.confに追加します。このファイルは、デフォルトでは/etcディレクトリにあります（Debian/Ubuntuオペレーティングシステムでは/etc/munin/munin.confになります）。

```
[<ID>; <hostname>.<domain>]
address <host IP address>
use_node_name yes
```

ここで、*<ID>*は表示されるホストの識別子、*<hostname>*はホストの名前、*<domain>*はドメインの名前、*<host IP address>*はホストのIPアドレスです。

Muninモニタリングシステムの設定に関する公式マニュアルについては、http://guide.munin-monitoring.org/en/latest/を参照してください。

Zabbixモニタリングシステムとの統合

Dr.Web SNMPDとZabbixモニタリングシステム間の接続を確立するために必要なファイルテンプレートは、*<opt_dir>*/share/drweb-snmpd/connectors/zabbixディレクトリにあります。

ファイル	説明
zbx_drweb.xml	Dr.Web for UNIX Mail Serversがインストールされているモニタリング対象ホストを説明するためのテンプレート
snmptt.drweb.zabbix.conf	SNMPトラップハンドラーである*snmptt*ユーティリティの設定

モニタリング対象ホストの機能を説明するためのテンプレート：

- カウンター（*Zabbix*の用語では「アイテム」）の説明。デフォルトでは、テンプレートはSNMP v2で使用されるように設定されています。
- 既存のグラフのセット：スキャンされたファイルの数と検出された脅威のタイプ別の分布。

Zabbixにホストを接続する

この説明では、Zabbixモニタリングシステムがモニタリングサーバーにすでにデプロイされており、モニタリング対象ホストにはDr.Web SNMPDがインストール済みで機能している（コンポーネントはsnmpdとともにプロキシモードで機能する）と想定しています。さらに、モニタリング対象ホストから*SNMPトラップ*通知（保護対象サーバーでDr.Web for UNIX Mail Serversによって検出された脅威に関する通知を含む）を受信する場合は、モニタリングサーバーに`net-snmp`パッケージをインストールします（標準ツールの`snmptt`および`snmptrapd`が使用されます）。

1. Zabbix Webインターフェースの**設定**→**テンプレート**タブで、*<opt_dir>*`/share/drweb-snmpd/connectors/zabbix/zbx_drweb.xml`ファイルからモニタリング対象ホストのテンプレートをインポートします。
2. モニタリング対象ホストを適切なリストに追加します（**ホスト** → **ホストの作成**）。ホストの適切なパラメータとSNMPインターフェースの設定を指定します（ホストの`drweb-snmpd`とsnmpdの設定と一致する必要があります）。
   - ホストタブ：

     ホスト名：*drweb-host*

     表示名：*DRWEB_HOST*

     グループ：*Linux servers*を選択します

     SNMPインターフェース：**追加**をクリックして、Dr.Web SNMPDによって使用されるIPアドレスとポートを指定します（Dr.Web SNMPDはローカルホストで動作すると見なされるため、デフォルトではアドレス*127.0.0.1*とポート*161*が指定されています）。
   - テンプレートタブ：

     追加を押し、*DRWEB*を確認し、**選択**を押します。
   - マクロタブ：

     マクロ：*{$SNMP_COMMUNITY}*

     値：SNMP V2cに「read community」を指定します（デフォルトでは、*public*）。

     保存をクリックします。

     注意：*{$SNMP_COMMUNITY}*マクロは、ホストテンプレートで直接指定できます。

デフォルトでは、インポートされた*DRWEB*テンプレートはSNMP v2用に設定されています。他のバージョンのSNMPを使用する場合は、該当するページでテンプレートに必要な編集を行います。

3. テンプレートがモニタリング対象ホストにバインドされた後、SNMP設定が正しく指定されていれば、Zabbixモニタリングシステムはテンプレートのカウンター（*アイテム*）のデータ収集を開始します。収集されたデータは、**監視データ**→**最新データ**と**監視データ**→**グラフ**に表示されます。
4. Dr.Web SNMPDから*SNMPトラップ*通知を収集するために、特別な*アイテム**drweb-traps*が使用されます。受信した*SNMPトラップ*通知のログは、**監視データ** → **最新データ** → **drweb-traps** –> ヒストリページで利用できます。Zabbixは通知を収集するために、`net-snmp`パッケージの標準ツール`snmptt`と`snmptrapd`を使用します。Dr.Web SNMPDから*SNMPトラップ*通知を受信するためのツールの設定方法については、以下を参照してください。
5. 必要に応じて、Dr.Web SNMPDからの*SNMPトラップ*通知の受信時に状態を変更するトリガーを設定できます。状態の変更は、適切な通知を生成するためのイベントソースとして使用できます。以下の例は、トリガーの設定式を示しています。この式は**trigger expression**フィールドで指定されます。

- Zabbixバージョン*2.x*の場合：

```
({TRIGGER.VALUE}=0 &
{DRWEB:snmptrap[.*\.1\.3\.6\.1\.4\.1\.29690\..*].nodata(60)}=1 )|
({TRIGGER.VALUE}=1 &
{DRWEB:snmptrap[.*\.1\.3\.6\.1\.4\.1\.29690\..*].nodata(60)}=0)
```

- Zabbixバージョン*3.x*の場合：

```
({TRIGGER.VALUE}=0 and {drweb-host:snmptrap[".29690."].nodata(60)}=1 ) or
({TRIGGER.VALUE}=1 and {drweb-host:snmptrap[".29690."].nodata(60)}=0 )
```

Dr.Web SNMPDからの*SNMPトラップ*通知のログが1分以内に更新された場合、イベントがトリガーされます（値が1に設定されます）。ログが次の1分以内に更新されなかった場合、トリガーの値は再び0に設定されます。

深刻度では、このトリガーの通知タイプを*未分類*とは異なるものにすることをお勧めします（例：*警告*）。

ZabbixのSNMPトラップ通知の受信を設定する

1. モニタリング対象ホストのDr.Web SNMPDの設定（`TrapReceiver`パラメータ）で、Zabbixが動作しているホストで`snmptrapd`がリッスンするアドレスを指定する必要があります。次に例を示します。

```
SNMPD.TrapReceiver = 10.20.30.40:162
```

2. `snmptrapd`の設定ファイル（`snmptrapd.conf`）に、同じアドレスと、受信した*SNMPトラップ*通知を処理するアプリケーションを指定します（この例では`snmptthandler`、`snmptt`コンポーネント）。

```
snmpTrapdAddr 10.20.30.40:162
traphandle default /usr/sbin/snmptthandler
```

Dr.Web SNMPDによって送信された*SNMPトラップ*を*snmptt*が不明なものとして破棄しないように、ファイルに次の文字列を追加します。

```
outputOption n
```

3. `snmptthandler`コンポーネントは、Zabbixのホストテンプレートに設定されている正規表現（*アイテム drweb-traps*エレメント）に対応する指定された形式に従って、受信した*SNMPトラップ*通知をディスクのファイルに保存します。保存された通知の*SNMPトラップ*フォーマットは、*<opt_dir>*`/share/drweb-snmpd/connectors/zabbix/snmptt.drweb.zabbix.conf`.ファイルで指定します。ファイルは、`/etc/snmp`にコピーする必要があります。
4. さらに、フォーマットファイルへのパスを`snmptt.ini`に指定する必要があります。

```
[TrapFiles]
# A list of snmptt.conf files (this is NOT the snmptrapd.conf file).
# The COMPLETE path and filename. Ex: '/etc/snmp/snmptt.conf'
snmptt_conf_files = <<END
/etc/snmp/snmptt.conf
/etc/snmp/snmptt.drweb.zabbix.conf
END
```

その後、デーモンモードで起動している場合は`snmptt`を再起動します。

5. Zabbixサーバーの設定ファイル(`zabbix-server.conf`)で、次の設定を指定します(または、すでに指定されているかどうかを確認します)。

```
SNMPTrapperFile=/var/log/snmptt/snmptt.log
StartSNMPTrapper=1
```

ここで、`/var/log/snmptt/snmptt.log`は、受信したSNMPトラップ通知の情報を登録するために`snmptt`が使用するログファイルです。

Zabbixの公式マニュアルについては、https://www.zabbix.com/documentation/current/enを参照してください。

Nagiosモニタリングシステムとの統合

Dr.Web SNMPDとNagiosモニタリングシステム間の接続を確立するために必要なファイルとNagiosの設定例は、*<opt_dir>*`/share/drweb-snmpd/connectors/nagios`ディレクトリにあります。

ファイル	説明
`nagiosgraph/rrdopts.conf-sample`	RRD設定ファイルの例
`objects/drweb.cfg`	*drweb*オブジェクトを記述する設定ファイル
`objects/nagiosgraph.cfg`	Nagiosが使用するNagiosgraphによって使用されるグラフ描画のためのコンポーネントの設定ファイル
`plugins/check_drweb`	Dr.Web for UNIX Mail Serversがインストールされているホストからデータを収集するためのスクリプト
`plugins/eventhandlers/submit_check_result`	*SNMPトラップ*通知を処理するためのスクリプト
`snmp/snmptt.drweb.nagios.conf`	SNMPトラップハンドラーである*snmptt*ユーティリティの設定

Nagiosにホストを接続する

この説明では、Webサーバーとグラフィックツール Nagiosgraphの設定を含むNagiosモニタリングシステムがモニタリングサーバーにすでにデプロイされており、モニタリング対象ホストにはDr.Web SNMPDがインストール済みで機能している(コンポーネントはsnmpdとともにプロキシモードで機能する)と想定しています。さらに、モニタリング対象ホストから*SNMPトラップ*通知(保護対象サーバーでDr.Web for UNIX Mail Serversによって検出された脅威に関する通知を含む)を受信する場合は、モニタリングサーバーに`net-snmp`パッケージをインストールします(標準ツールの`snmptt`および`snmptrapd`が使用されます)。

現在のマニュアルでは、次のようなパスの規則が使用されています(実際のパスはオペレーティングシステムとNagiosのインストールによって異なります)。

- *<NAGIOS_PLUGINS_DIR>* - Nagiosプラグインを含むディレクトリ(例:`/usr/lib64/nagios/plugins`)。
- *<NAGIOS_ETC_DIR>* - Nagios設定を含むディレクトリ(例:`/etc/nagios`)。

- *<NAGIOS_OBJECTS_DIR>* - Nagiosオブジェクトを含むディレクトリ（例：`/etc/nagios/objects`）。
- *<NAGIOSGRAPH_DIR>* - Nagiosgraphディレクトリ（例：`/usr/local/nagiosgraph`）。
- *<NAGIOS_PERFDATA_LOG>* - Nagiosがサービスチェックの結果を記録するファイル（*<NAGIOSGRAPH_DIR>*`/etc/nagiosgraph.conf`の`perflog`ファイルと同じでなければなりません）。このファイルのレコードは*<NAGIOSGRAPH_DIR>*`/bin/insert.pl`スクリプトによって読み取られ、対応するRRAアーカイブRRDツールに記録されます。

Nagiosを設定する：

1. `check_drweb`ファイルを*<NAGIOS_PLUGINS_DIR>*ディレクトリに、`drweb.cfg`ファイルを*<NAGIOS_OBJECTS_DIR>*ディレクトリにコピーします。
2. 監視対象のDr.Web for UNIX Mail Serversがあるホストを*drweb*グループに追加します。ホストではDr.Web SNMPDが実行されている必要があります。デフォルトでは、このグループには*localhost*のみが追加されます。
3. 必要に応じて、`snmplwalk`ツールを介して*drweb*ホストのDr.Web SNMPDに接続するように指示する`check_drweb`コマンドを編集します。

   ```
   snmpwalk -c public -v 2c $HOSTADDRESS$:161
   ```

   SNMPプロトコルの適切なバージョンとパラメータ（「*コミュニティストリング*」や認証パラメータなど）の他、ポートを指定します。`$HOSTADDRESS$`変数をコマンドに含める必要があります（この変数は後でコマンドが呼び出されたときに、Nagiosによって正しいホストアドレスに自動的に置き換えられるためです）。このコマンドではOIDは必要ありません。また、実行ファイルへのフルパス（通常は`/usr/local/bin/snmpwalk`）を使用してコマンドを指定することをお勧めします。
4. 次の文字列をファイルに追加して、*<NAGIOS_ETC_DIR>*`/nagios.cfg`設定ファイルの*DrWeb*オブジェクトを接続します。

   ```
   cfg_file=<NAGIOS_OBJECTS_DIR>/drweb.cfg
   ```

5. *DrWebグラフィック用のRRDツール*設定を`rrdopts.conf-sample`ファイルから*<NAGIOSGRAPH_DIR>*`/etc/rrdopts.conf`ファイルに追加します。
6. Nagiosgraphがまだ設定されていない場合は、その設定に対して次の手順を実行します。
   - `nagiosgraph.cfg`ファイルを*<NAGIOS_OBJECTS_DIR>*ディレクトリにコピーし、`process-service-perfdata-for-nagiosgraph`コマンドで`insert.pl`スクリプトへのパスを編集します。たとえば、次のようになります。

     ```
     $ awk '$1 == "command_line" { $2 = "<NAGIOSGRAPH_DIR>/bin/insert.pl" }
     { print }' ./objects/nagiosgraph.cfg >
     <NAGIOS_OBJECTS_DIR>/nagiosgraph.cfg
     ```

   - 次の行を追加して、*<NAGIOS_ETC_DIR>*`/nagios.cfg`設定ファイルにこのファイルを接続します。

     ```
     cfg_file=<NAGIOS_OBJECTS_DIR>/nagiosgraph.cfg
     ```

7. *<NAGIOS_ETC_DIR>*`/nagios.cfg` 設定ファイルのNagiosパラメータの値を確認します。

```
check_external_commands=1
execute_host_checks=1
accept_passive_host_checks=1
enable_notifications=1
enable_event_handlers=1

process_performance_data=1
service_perfdata_file=/usr/nagiosgraph/var/rrd/perfdata.log
service_perfdata_file_template=$LASTSERVICECHECK$||$HOSTNAME$||$SERVICEDE
SC$||$SERVICEOUTPUT$||$SERVICEPERFDATA$
service_perfdata_file_mode=a
service_perfdata_file_processing_interval=30
service_perfdata_file_processing_command=process-service-perfdata-for-
nagiosgraph

check_service_freshness=1
enable_flap_detection=1
enable_embedded_perl=1
enable_environment_macros=1
```

NagiosのSNMPトラップ通知の受信を設定する

1. モニタリング対象ホストのDr.Web SNMPDの設定（`TrapReceiver`パラメータ）で、Nagiosが動作しているホストで`snmptrapd`がリッスンするアドレスを指定します。次に例を示します。

   ```
   SNMPD.TrapReceiver = 10.20.30.40:162
   ```

2. *SNMPトラップ*を受信したときに呼び出される *<NAGIOS_PLUGINS_DIR>*`/eventhandlers/submit_check_result`スクリプトがあるかどうかを確認します。スクリプトが見つからない場合は、*<opt_dir>*`/share/drweb-snmpd/connectors/nagios/plugins/eventhandlers/`ディレクトリの`submit_check_result`ファイルをこの場所にコピーします。このファイルで、`CommandFile`パラメータに指定されているパスを変更します。これは、*<NAGIOS_ETC_DIR>*`/nagios.cfg`ファイルの`command_file`パラメータと同じ値である必要があります。

3. `snmptt.drweb.nagios.conf`ファイルを`/etc/snmp/snmp/`ディレクトリにコピーします。このファイルで、パスを`submit_check_result`に変更します。たとえば、次のようなコマンドを使用します。

   ```
   $ awk '$1 == "EXEC" { $2 =
   <NAGIOS_PLUGINS_DIR>/eventhandlers/submit_check_result }{ print}'
   ./snmp/snmptt.drweb.nagios.conf > /etc/snmp/snmp/snmptt.drweb.nagios.conf
   ```

4. 「`/etc/snmp/snmptt.drweb.nagios.conf`」文字列を`/etc/snmp/snmptt.ini`ファイルに追加します。その後、デーモンモードで起動している場合は`snmptt`を再起動します。

Nagiosに必要な設定ファイルをすべて追加して編集したら、次のコマンドを使用してNagiosをデバッグモードで実行します。

```
# nagios -v <NAGIOS_ETC_DIR>/nagios.cfg
```

このコマンドを受け取ると、Nagiosは設定エラーをチェックします。エラーがなかった場合は、Nagiosを通常どおりに再起動できます（たとえば、`service nagios restart`コマンドを使用して）。

Nagiosの公式マニュアルについては、https://www.nagios.org/documentation/を参照してください。

Dr.Web SNMP MIB

SNMPプロトコルを介して外部モニタリングシステムから取得できるDr.Web for UNIX Mail Serversの動作パラメータの一覧を以下の表に示します。

パラメータ名	パラメータのOID	パラメータのタイプと説明
すべての名前に共通のプレフィックス: .iso.org.dod.internet.private.enterprises.drweb.drwebSnmpd すべてのOIDに共通のプレフィックス: .1.3.6.1.4.1.29690.2		
alert	イベントに関する非同期通知（SNMPトラップ）	
threatAlert	.1.1	脅威の検出に関する通知
threatAlertFile	.1.1.1	感染ファイル名（文字列）
threatAlertType	.1.1.2	脅威の種類（整数*）
threatAlertName	.1.1.3	脅威名（文字列）
threatAlertOrigin	.1.1.4	脅威を検出したコンポーネントの識別子（整数***）
threatAlertRemoteIp	.1.1.5	ファイルへのアクセスに使用されたリモートホストのIPアドレス（文字列）
threatAlertRemoteUser	.1.1.6	ファイルにアクセスしたリモートユーザーの名前（文字列）
threatAlertRemoteDomain	.1.1.7	ファイルへのアクセスに使用されたリモートホストの名前（文字列）
threatActionErrorAlert	.1.2	脅威を駆除しようとしたときに発生したエラーに関する通知
threatActionErrorAlertFile	.1.2.1	感染ファイル名（文字列）
threatActionErrorAlertType	.1.2.2	脅威の種類（整数*）
threatActionErrorAlertName	.1.2.3	脅威名（文字列）
threatActionErrorAlertOrigin	.1.2.4	脅威を検出したコンポーネントの識別子（整数***）
threatActionErrorAlertError	.1.2.5	エラーの説明（文字列）
threatActionErrorAlertErrorCode	.1.2.6	エラーコード（エラーカタログのコードに対応する整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
threatActionErrorAlertAction	.1.2.7	失敗したアクション（1 - 修復、2 - 隔離へ移動、3 - 削除、4 - 報告、5 - 無視）
componentFailureAlert	.1.3	コンポーネント障害に関する通知
componentFailureAlertName	.1.3.1	コンポーネント識別子（整数***）
componentFailureAlertExitCodeDescription	.1.3.2	コンポーネント終了コードの説明（文字列）
componentFailureAlertExitCode	.1.3.3	エラーコード（エラーカタログのコードに対応する整数）
infectedUrlAlert	.1.4	悪質なURLのブロックに関する通知（HTTP/HTTPS接続）
infectedUrlAlertUrl	.1.4.1	ブロックされたURL（文字列）
infectedUrlAlertDirection	.1.4.2	HTTPメッセージの方向（整数：1 - 要求、2 - 応答）
infectedUrlAlertType	.1.4.3	脅威の種類（整数*）
infectedUrlAlertName	.1.4.4	脅威名（文字列）
infectedUrlAlertOrigin	.1.4.5	脅威を検出したコンポーネントの識別子（整数***）
infectedUrlAlertSrcIp	.1.4.6	接続元のIPアドレス（文字列）
infectedUrlAlertSrcPort	.1.4.7	接続元のポート（整数）
infectedUrlAlertDstIp	.1.4.8	接続先のIPアドレス（文字列）
infectedUrlAlertDstPort	.1.4.9	接続先のポート（整数）
infectedUrlAlertSniHost	.1.4.10	接続先のSNI（SSL接続）（文字列）
infectedUrlAlertExePath	.1.4.11	接続を確立したプログラムの実行パス（文字列）
infectedUrlAlertUserName	.1.4.12	接続を確立するプログラムを実行する権限を持つユーザーの名前（文字列）
infectedEmailAttachmentAlert	.1.5	感染したメールの添付ファイルの検出に関する通知
infectedEmailAttachmentAlertType	.1.5.1	脅威の種類（整数*）
infectedEmailAttachmentAlertName	.1.5.2	脅威名（文字列）

パラメータ名	パラメータの OID	パラメータのタイプと説明
infectedEmailAttachmentAlertOrigin	.1.5.3	脅威を検出したコンポーネントの識別子（整数***）
infectedEmailAttachmentAlertSocket	.1.5.4	メールメッセージの送信元のIPアドレス（文字列）
infectedEmailAttachmentAlertMailFrom	.1.5.5	メールメッセージの送信者（文字列）
infectedEmailAttachmentAlertRcptTo	.1.5.6	メールメッセージの受信者（文字列）
infectedEmailAttachmentAlertMessageId	.1.5.7	メールメッセージのMessage-IDヘッダーの値（文字列）
infectedEmailAttachmentAlertAction	.1.5.8	メールメッセージ全体または感染した添付ファイルに適用されたアクション（整数：1 - 再圧縮、2 - 拒否、3 - 破棄、4 - 修復、5 - 隔離へ移動、6 - 削除）
infectedEmailAttachmentAlertDivert	.1.5.9	メールメッセージの方向（整数：1 - 受信、2 - 送信）
infectedEmailAttachmentAlertSrcIp	.1.5.10	接続元のIPアドレス（文字列）
infectedEmailAttachmentAlertSrcPort	.1.5.11	接続元のポート（整数）
infectedEmailAttachmentAlertDstIp	.1.5.12	接続先のIPアドレス（文字列）
infectedEmailAttachmentAlertDstPort	.1.5.13	接続先のポート（整数）
infectedEmailAttachmentAlertSniHost	.1.5.14	接続先のSNI（SSL接続）（文字列）
infectedEmailAttachmentAlertProtocol	.1.5.15	プロトコルの種類（整数：1 - SMTP、2 - POP3、3 - IMAP、4 - HTTP）
infectedEmailAttachmentAlertExePath	.1.5.16	接続を確立したプログラムの実行パス（文字列）
infectedEmailAttachmentAlertUserName	.1.5.17	接続を確立するプログラムを実行する権限を持つユーザーの名前（文字列）
categoryUrlAlert	.1.6	不要なカテゴリーに属するURLのブロックに関する通知
categoryUrlAlertUrl	.1.6.1	ブロックされたURL（文字列）
categoryUrlAlertCategory	.1.6.2	URLが属するWebリソースカテゴリー（整数**）

パラメータ名	パラメータのOID	パラメータのタイプと説明
categoryUrlAlertOrigin	.1.6.3	脅威を検出したコンポーネントの識別子（整数***）
categoryUrlAlertSrcIp	.1.6.4	接続元のIPアドレス（文字列）
categoryUrlAlertSrcPort	.1.6.5	接続元のポート（整数）
categoryUrlAlertDstIp	.1.6.6	接続先のIPアドレス（文字列）
categoryUrlAlertDstPort	.1.6.7	接続先のポート（整数）
categoryUrlAlertSniHost	.1.6.8	接続先のSNI（SSL接続）（文字列）
categoryUrlAlertExePath	.1.6.9	接続を確立したプログラムの実行パス（文字列）
categoryUrlAlertUserName	.1.6.10	接続を確立するプログラムを実行する権限を持つユーザーの名前（文字列）
categoryUrlEmailAttachmentAlert	.1.7	メールメッセージ内の不要なURLの検出に関する通知
categoryUrlEmailAttachmentAlertType	.1.7.1	URLが属するWebリソースカテゴリー（整数**）
categoryUrlEmailAttachmentAlertOrigin	.1.7.2	脅威を検出したコンポーネントの識別子（整数***）
categoryUrlEmailAttachmentAlertSocket	.1.7.3	メールメッセージの送信元のIPアドレス（文字列）
categoryUrlEmailAttachmentAlertMailFrom	.1.7.4	メールメッセージの送信者（文字列）
categoryUrlEmailAttachmentAlertRcptTo	.1.7.5	メールメッセージの受信者（文字列）
categoryUrlEmailAttachmentAlertMessageId	.1.7.6	メールメッセージのMessage-IDヘッダーの値（文字列）
categoryUrlEmailAttachmentAlertAction	.1.7.7	メールメッセージ全体または添付ファイルに適用されたアクション（整数：1 - 再圧縮、2 - 拒否、3 - 破棄、4 - 修復、5 - 隔離へ移動、6 - 削除）
categoryUrlEmailAttachmentAlertDivert	.1.7.8	メールメッセージの方向（整数：1 - 受信、2 - 送信）
categoryUrlEmailAttachmentAlertSrcIp	.1.7.9	接続元のIPアドレス（文字列）

パラメータ名	パラメータの OID	パラメータのタイプと説明
categoryUrlEmailAttachmentAlertSrcPort	.1.7.10	接続元のポート（整数）
categoryUrlEmailAttachmentAlertDstIp	.1.7.11	接続先のIPアドレス（文字列）
categoryUrlEmailAttachmentAlertDstPort	.1.7.12	接続先のポート（整数）
categoryUrlEmailAttachmentAlertSniHost	.1.7.13	接続先のSNI（SSL接続）（文字列）
categoryUrlEmailAttachmentAlertProtocol	.1.7.14	プロトコルの種類（整数：1 - SMTP、2 - POP3、3 - IMAP、4 - HTTP）
categoryUrlEmailAttachmentAlertExePath	.1.7.15	接続を確立したプログラムの実行パス（文字列）
categoryUrlEmailAttachmentAlertUserName	.1.7.16	接続を確立するプログラムを実行する権限を持つユーザーの名前（文字列）
spamEmailAlert	.1.8	メールメッセージのスパム認識に関する通知
spamEmailAlertOrigin	.1.8.1	脅威を検出したコンポーネントの識別子（整数***）
spamEmailAlertSocket	.1.8.2	メールメッセージの送信元のIPアドレス（文字列）
spamEmailAlertMailFrom	.1.8.3	メールメッセージの送信者（文字列）
spamEmailAlertRcptTo	.1.8.4	メールメッセージの受信者（文字列）
spamEmailAlertMessageId	.1.8.5	メールメッセージのMessage-IDヘッダーの値（文字列）
spamEmailAlertAction	.1.8.6	メールメッセージ全体または添付ファイルに適用されたアクション（整数：1 - 再圧縮、2 - 拒否、3 - 破棄、4 - 修復、5 - 隔離へ移動、6 - 削除）
spamEmailAlertDivert	.1.8.7	メールメッセージの方向（整数：1 - 受信、2 - 送信）
spamEmailAlertSrcIp	.1.8.8	接続元のIPアドレス（文字列）
spamEmailAlertSrcPort	.1.8.9	接続元のポート（整数）
spamEmailAlertDstIp	.1.8.10	接続先のIPアドレス（文字列）

パラメータ名	パラメータの OID	パラメータのタイプと説明
spamEmailAlertDstPort	.1.8.11	接続先のポート（整数）
spamEmailAlertSniHost	.1.8.12	接続先のSNI（SSL接続）（文字列）
spamEmailAlertProtocol	.1.8.13	プロトコルの種類（整数：1 - SMTP、2 - POP3、3 - IMAP、4 - HTTP）
spamEmailAlertExePath	.1.8.14	接続を確立したプログラムの実行パス（文字列）
spamEmailAlertUserName	.1.8.15	接続を確立するプログラムを実行する権限を持つユーザーの名前（文字列）
blockedConnectionAlert	.1.9	ネットワーク接続のブロックに関する通知
blockedConnectionAlertOrigin	.1.9.1	脅威を検出したコンポーネントの識別子（整数***）
blockedConnectionAlertDivert	.1.9.2	接続の方向（整数：1 - 受信、2 - 送信）
blockedConnectionAlertSrcIp	.1.9.3	接続元のIPアドレス（文字列）
blockedConnectionAlertSrcPort	.1.9.4	接続元のポート（整数）
blockedConnectionAlertDstIp	.1.9.5	接続先のIPアドレス（文字列）
blockedConnectionAlertDstPort	.1.9.6	接続先のポート（整数）
blockedConnectionAlertSniHost	.1.9.7	接続先のSNI（SSL接続）（文字列）
blockedConnectionAlertProtocol	.1.9.8	プロトコルの種類（整数：1 - SMTP、2 - POP3、3 - IMAP、4 - HTTP）
blockedConnectionAlertExePath	.1.9.9	接続を確立したプログラムの実行パス（文字列）
blockedConnectionAlertUserName	.1.9.10	接続を確立するプログラムを実行する権限を持つユーザーの名前（文字列）
stat	**Dr.Web for UNIX Mail Servers の動作に関する統計**	
threatCounters	.2.1	検出された脅威のカウンター
knownVirus	.2.1.1	検出された既知のウイルスの数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
suspicious	.2.1.2	検出された疑わしいオブジェクトの数（カウンター、整数）
adware	.2.1.3	検出されたアドウェアの数（カウンター、整数）
dialers	.2.1.4	検出されたダイアラーの数（カウンター、整数）
joke	.2.1.5	検出されたジョークプログラムの数（カウンター、整数）
riskware	.2.1.6	検出されたリスクウェアの数（カウンター、整数）
hacktool	.2.1.7	検出されたハッキングツールの数（カウンター、整数）
scanErrors	.2.2	ファイルのスキャン中に発生したエラーのカウンター
sePathNotAbsolute	.2.2.1	「パスが絶対パスではありません」エラーの発生回数（カウンター、整数）
seFileNotFound	.2.2.2	「ファイルが見つかりません」エラーの発生回数（カウンター、整数）
seFileNotRegular	.2.2.3	「ファイルは通常のファイルではありません」エラーの発生回数（カウンター、整数）
seFileNotBlockDevice	.2.2.4	「ファイルはブロックデバイスではありません」エラーの発生回数（カウンター、整数）
seNameTooLong	.2.2.5	「パスまたはファイル名が長すぎます」エラーの発生回数（カウンター、整数）
seNoAccess	.2.2.6	「パーミッションが拒否されました」エラーの発生回数（カウンター、整数）
seReadError	.2.2.7	「読み取りエラー」の発生回数（カウンター、整数）
seWriteError	.2.2.8	「書き込みエラー」の発生回数（カウンター、整数）
seFileTooLarge	.2.2.9	「ファイルサイズが大きすぎます」エラーの発生回数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
seFileBusy	.2.2.10	「ファイルがビジーです」エラーの発生回数（カウンター、整数）
seUnpackingError	.2.2.20	「アンパックエラー」の発生回数（カウンター、整数）
sePasswordProtecetd	.2.2.21	「パスワード保護」エラーの発生回数（カウンター、整数）
seArchCrcError	.2.2.22	「アーカイブのCRCエラー」の発生回数（カウンター、整数）
seArchInvalidHeader	.2.2.23	「無効なアーカイブヘッダーです」エラーの発生回数（カウンター、整数）
seArchNoMemory	.2.2.24	「アーカイブを処理するのに十分なメモリがありません」エラーの発生回数（カウンター、整数）
seArchIncomplete	.2.2.25	「不完全なアーカイブ」エラーの発生回数（カウンター、整数）
seCanNotBeCured	.2.2.26	「オブジェクトを修復できません」エラーの発生回数（カウンター、整数）
sePackerLevelLimit	.2.2.30	「圧縮されたオブジェクトが最大ネスティングレベルを超えました」エラーの発生回数（カウンター、整数）
seArchiveLevelLimit	.2.2.31	「アーカイブが最大ネスティングレベルを超えました」エラーの発生回数（カウンター、整数）
seMailLevelLimit	.2.2.32	「メールファイルが最大ネスティングレベルを超えました」エラーの発生回数（カウンター、整数）
seContainerLevelLimit	.2.2.33	「コンテナファイルが最大ネスティングレベルを超えました」エラーの発生回数（カウンター、整数）
seCompressionLimit	.2.2.34	「最大圧縮率を超えました」エラーの発生回数（カウンター、整数）
seReportSizeLimit	.2.2.35	「スキャン結果レポートの最大サイズを超えました」エラーの発生回数（カウンター、整数）
seScanTimeout	.2.2.40	「スキャンタイムアウトに達しました」エラーの発生回数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
seEngineCrash	.2.2.41	「Scan Engine のクラッシュが検出されました」エラーの発生回数（カウンター、整数）
seEngineHangup	.2.2.42	「Scan Engine が応答を停止しました」エラーの発生回数（カウンター、整数）
seEngineError	.2.2.44	「Scan Engine の内部エラー」の発生回数（カウンター、整数）
seNoLicense	.2.2.45	「有効なライセンスが見つかりません」エラーの発生回数（カウンター、整数）
seCuringLimitReached	.2.2.47	「修復試行限界に達しました」エラーの発生数（カウンター、整数）
seNonSupportedDisk	.2.2.50	「サポートされていないディスクです」エラーの発生回数（カウンター、整数）
seUnexpectedError	.2.2.100	「予期せぬエラーです」の発生数（カウンター、整数）
scanLoadAverage	.2.3	ファイルスキャン負荷の指標
filesScannedTable	.2.3.1	他のコンポーネントのリクエストによりスキャンされたファイルの平均数
filesScannedEntry	.2.3.1.1	製品のコンポーネント（テーブル行全体、レコード）
filesScannedIndex	.2.3.1.1.1	コンポーネントのインデックス（識別子、整数***）
filesScannedOrigin	.2.3.1.1.2	コンポーネントの名前
filesScanned1min	.2.3.1.1.3	1秒間にチェックされたファイルの平均（1分間の平均）数（文字列）
filesScanned5min	.2.3.1.1.4	1秒間にチェックされたファイルの平均（5分間の平均）数（文字列）
filesScanned15min	.2.3.1.1.5	1秒間にチェックされたファイルの平均（15分間の平均）数（文字列）
bytesScannedTable	.2.3.2	他のコンポーネントのリクエストにより実行されたスキャンの平均速度（1秒あたりのバイト数）
bytesScannedEntry	.2.3.2.1	製品のコンポーネント（テーブル行全体、レコード）

パラメータ名	パラメータの OID	パラメータのタイプと説明
bytesScannedIndex	.2.3.2.1.1	コンポーネントのインデックス（識別子、整数***）
bytesScannedOrigin	.2.3.2.1.2	コンポーネントの名前
bytesScanned1min	.2.3.2.1.3	1秒間にスキャンされた平均バイト数（1分間の平均）（文字列）
bytesScanned5min	.2.3.2.1.4	1秒間にスキャンされた平均バイト数（5分間の平均）（文字列）
bytesScanned15min	.2.3.2.1.5	1秒間にスキャンされた平均バイト数（15分間の平均）（文字列）
cacheHitFilesTable	.2.3.3	コンポーネントのリクエストによりキャッシュから取得されたスキャンレポートの平均数
cacheHitFilesEntry	.2.3.3.1	製品のコンポーネント（テーブル行全体、レコード）
cacheHitFilesIndex	.2.3.3.1.1	コンポーネントのインデックス（識別子、整数***）
cacheHitFilesOrigin	.2.3.3.1.2	コンポーネントの名前
cacheHitFiles1min	.2.3.3.1.3	1秒間にキャッシュから取得されたレポートの平均（1分間の平均）数（文字列）
cacheHitFiles5min	.2.3.3.1.4	1秒間にキャッシュから取得されたレポートの平均（5分間の平均）数（文字列）
cacheHitFiles15min	.2.3.3.1.5	1秒間にキャッシュから取得されたレポートの平均（15分間の平均）数（文字列）
errorsTable	.2.3.4	コンポーネントのリクエストにより実行されたスキャン中の平均エラー数
errorsEntry	.2.3.4.1	製品のコンポーネント（テーブル行全体、レコード）
errorsIndex	.2.3.4.1.1	コンポーネントのインデックス（識別子、整数***）
errorsOrigin	.2.3.4.1.2	コンポーネントの名前
errors1min	.2.3.4.1.3	1秒間のスキャンエラーの平均（1分間の平均）数（文字列）

パラメータ名	パラメータの OID	パラメータのタイプと説明
errors5min	.2.3.4.1.4	1秒間のスキャンエラーの平均（5分間の平均）数（文字列）
errors15min	.2.3.4.1.5	1秒間のスキャンエラーの平均（15分間の平均）数（文字列）
net	.2.4	ネットワークアクティビティの統計
markedAsSpam	.2.4.1	スパムとしてマークされたメールメッセージの数（カウンター、整数）
blockedInfectionSource	.2.4.101	「感染源」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedNotRecommended	.2.4.102	「非推奨」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedAdultContent	.2.4.103	「アダルトコンテンツ」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedViolence	.2.4.104	「暴力」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedWeapons	.2.4.105	「武器」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedGambling	.2.4.106	「ギャンブル」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedDrugs	.2.4.107	「麻薬」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedObsceneLanguage	.2.4.108	「卑猥な表現」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedChats	.2.4.109	「チャット」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedTerrorism	.2.4.110	「テロリズム」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedFreeEmail	.2.4.111	「無料メールサービス」カテゴリーに属するブロックされたURLの数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
blockedSocialNetworks	.2.4.112	「ソーシャルネットワーク」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedOwnersNotice	.2.4.113	「著作権者の申し立て」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedOnlineGames	.2.4.114	「オンラインゲーム」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedAnonymizers	.2.4.115	「アノニマイザー」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedCryptocurrencyMiningPools	.2.4.116	「仮想通貨マイニングプール」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedJobs	.2.4.117	「求人情報サイト」カテゴリーに属するブロックされたURLの数（カウンター、整数）
blockedBlackList	.2.4.120	ユーザーのブラックリストからブロックされたURLの数（カウンター、整数）
info	**Dr.Web for UNIX Mail Servers の現在の状態に関する情報**	
components	.3.1	Dr.Web for UNIX Mail Servers コンポーネントのステータス
configd	.3.1.1	drweb-configd コンポーネントデータ
configdState	.3.1.1.1	コンポーネントの現在の状態（整数 ****）
configdExitCode	.3.1.1.2	前回の終了コード（エラーカタログのコードに対応する整数）
configdExitTime	.3.1.1.3	前回の終了時刻（*UNIX時間*）
configdInstalledApps	.3.1.1.101	インストールされているコンポーネントのリスト
configdAppEntry	.3.1.1.101.1	インストールされているコンポーネントに関する情報（テーブル行全体、レコード）
configdAppIndex	.3.1.1.101.1.1	インストールされているコンポーネントのインデックス（序数）（整数）

パラメータ名	パラメータのOID	パラメータのタイプと説明
configdAppName	.3.1.1.101.1.2	インストールされているコンポーネントの名前（文字列）
configdAppExePath	.3.1.1.101.1.3	コンポーネントの実行パス（文字列）
configdAppInstallTime	.3.1.1.101.1.4	コンポーネントがインストールされた時刻（*UNIX時間*）
configdAppIniSection	.3.1.1.101.1.5	設定ファイルにあるコンポーネントのパラメータを含むセクションの名前
scanEngine	.3.1.2	drweb-seコンポーネントデータ
scanEngineState	.3.1.2.1	コンポーネントの現在の状態（整数****）
scanEngineExitCode	.3.1.2.2	前回の終了コード（エラーカタログのコードに対応する整数）
scanEngineExitTime	.3.1.2.3	前回の終了時刻（*UNIX時間*）
scanEngineStatus	.3.1.2.101	Dr.Web Virus-Finding Engineの現在の状態（整数）
scanEngineVersion	.3.1.2.102	Dr.Web Virus-Finding Engineのバージョン（文字列）
scanEngineVirusRecords	.3.1.2.103	ウイルスレコードの数（整数）
scanEngineMaxForks	.3.1.2.104	スキャン対象の子プロセスの最大数（整数）
scanEngineQueues	.3.1.2.105	スキャンタスクのキュー
scanEngineQueuesLow	.3.1.2.105.1	優先度の低いタスクのキュー
scanEngineQueueLowOut	.3.1.2.105.1.1	キューから出て処理に転送された優先度の低いタスクの数（カウンター、整数）
scanEngineQueueLowSize	.3.1.2.105.1.2	キュー内で処理待ちになっている優先度の低いタスクの数（カウンター、整数）
scanEngineQueuesMedium	.3.1.2.105.2	通常優先度のタスクのキュー
scanEngineQueueMediumOut	.3.1.2.105.2.1	キューから出て処理に転送された通常優先度のタスクの数（カウンター、整数）
scanEngineQueueMediumSize	.3.1.2.105.2.2	キュー内で処理待ちになっている通常優先度のタスクの数（カウンタ

パラメータ名	パラメータのOID	パラメータのタイプと説明
		ー、整数）
scanEngineQueuesHigh	.3.1.2.105.3	優先度の高いタスクのキュー
scanEngineQueueHighOut	.3.1.2.105.3.1	キューから出て処理に転送された優先度の高いタスクの数（カウンター、整数）
scanEngineQueueHighSize	.3.1.2.105.3.2	キュー内で処理待ちになっている優先度の高いタスクの数（カウンター、整数）
scanEngineVirusBasesTable	.3.1.2.106	ウイルスデータベースのリスト
scanEngineVirusBasesEntry	.3.1.2.106.1	ウイルスデータベースに関する情報（テーブル行全体、レコード）
scanEngineVirusBaseIndex	.3.1.2.106.1.1	ウイルスデータベースのインデックス（整数）
scanEngineVirusBasePath	.3.1.2.106.1.2	ウイルスデータベースファイルへのパス（文字列）
scanEngineVirusBaseRecords	.3.1.2.106.1.3	ウイルスデータベース内のレコード数（整数）
scanEngineVirusBaseVersion	.3.1.2.106.1.4	ウイルスデータベースのバージョン（整数）
scanEngineVirusBaseTimestamp	.3.1.2.106.1.5	ウイルスデータベースのタイムスタンプ（*UNIX時間*）
scanEngineVirusBaseMD5	.3.1.2.106.1.6	MD5チェックサム（文字列）
scanEngineVirusBaseLoadResult	.3.1.2.106.1.7	このウイルスデータベースのダウンロード結果（文字列）
scanEngineQueuesTab	.3.1.2.107	スキャンタスクキューのリスト
scanEngineQueueEntry	.3.1.2.107.1	キューに関する情報（テーブル行全体、レコード）
scanEngineQueueIndex	.3.1.2.107.1.1	キューのインデックス（序数）（整数）
scanEngineQueueName	.3.1.2.107.1.2	キューの名前（文字列）
scanEngineQueueOut	.3.1.2.107.1.3	キューから出て処理に転送されたタスクの数（カウンター、整数）
scanEngineQueueSize	.3.1.2.107.1.4	キュー内で処理待ちになっているタスクの数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
fileCheck	.3.1.3	drweb-filecheck コンポーネントデータ
fileCheckState	.3.1.3.1	コンポーネントの現在の状態（整数 ****）
fileCheckExitCode	.3.1.3.2	前回の終了コード（エラーカタログのコードに対応する整数）
fileCheckExitTime	.3.1.3.3	前回の終了時刻（*UNIX 時間*）
fileCheckScannedFiles	.3.1.3.101	スキャン済みファイル数（カウンター、整数）
fileCheckScannedBytes	.3.1.3.102	スキャン済みバイト数（カウンター、整数）
fileCheckCacheHitFiles	.3.1.3.103	キャッシュから取得したスキャンレポートの数（カウンター、整数）
fileCheckScanErrors	.3.1.3.104	スキャンエンジンでのエラーの発生回数（カウンター、整数）
fileCheckScanStat	.3.1.3.105	クライアントのリスト
fileCheckClientEntry	.3.1.3.105.1	クライアントに関する情報（テーブル行全体、レコード）
fileCheckClientIndex	.3.1.3.105.1.1	クライアントのインデックス（序数）（整数）
fileCheckClientName	.3.1.3.105.1.2	クライアントコンポーネントの名前（文字列）
fileCheckClientScannedFiles	.3.1.3.105.1.3	このクライアントでスキャンされたファイル数（カウンター、整数）
fileCheckClientScannedBytes	.3.1.3.105.1.4	このクライアントでスキャンされたバイト数（カウンター、整数）
fileCheckClientCacheHitFiles	.3.1.3.105.1.5	このクライアントのキャッシュから取得したスキャンレポートの数（カウンター、整数）
fileCheckClientScanErrors	.3.1.3.105.1.6	このクライアントで動作しているときにスキャンエンジンで発生したエラーの数（カウンター、整数）
update	.3.1.4	drweb-update コンポーネントデータ
updateState	.3.1.4.1	コンポーネントの現在の状態（整数 ****）

パラメータ名	パラメータのOID	パラメータのタイプと説明
updateExitCode	.3.1.4.2	前回の終了コード（エラーカタログのコードに対応する整数）
updateExitTime	.3.1.4.3	前回の終了時刻（*UNIX時間*）
updateBytesSent	.3.1.4.101	送信したバイト数（カウンター、整数）
updateBytesReceived	.3.1.4.102	受信したバイト数（カウンター、整数）
esagent	.3.1.5	drweb-esagentコンポーネントデータ
esagentState	.3.1.5.1	コンポーネントの現在の状態（整数****）
esagentExitCode	.3.1.5.2	前回の終了コード（エラーカタログのコードに対応する整数）
esagentExitTime	.3.1.5.3	前回の終了時刻（*UNIX時間*）
esagentWorkStatus	.3.1.5.101	コンポーネントの現在の動作モード（整数：1 - スタンドアロンモード、2 - 接続中、3 - 接続待ち、4 - 接続承認済み）
esagentIsConnected	.3.1.5.102	サーバーに接続されているかどうか（整数、0 - いいえ、1 - はい）
esagentServer	.3.1.5.103	使用されている集中管理サーバーのアドレス（文字列）
netcheck	.3.1.6	drweb-netcheckコンポーネントデータ
netcheckState	.3.1.6.1	コンポーネントの現在の状態（整数****）
netcheckExitCode	.3.1.6.2	前回の終了コード（エラーカタログのコードに対応する整数）
netcheckExitTime	.3.1.6.3	前回の終了時刻（*UNIX時間*）
netcheckLocalSeForks	.3.1.6.101	ローカルで利用可能なスキャンエンジンプロセスの数（整数）
netcheckRemoteSeForks	.3.1.6.102	リモートで利用可能なスキャンエンジンプロセスの数（整数）
netcheckLocalFilesScanned	.3.1.6.103	ローカルでスキャンされたファイルの数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
netcheckNetworkFilesScanned	.3.1.6.104	リモートスキャンでスキャンされたファイルの数（カウンター、整数）
netcheckLocalBytesScanned	.3.1.6.105	ローカルでスキャンされたバイト数（カウンター、整数）
netcheckNetworkBytesScanned	.3.1.6.106	リモートスキャンでスキャンされたバイト数（カウンター、整数）
netcheckLocalBytesIn	.3.1.6.107	ローカルクライアントから受信したバイト数（カウンター、整数）
netcheckLocalBytesOut	.3.1.6.108	ローカルクライアントに送信したバイト数（カウンター、整数）
netcheckNetworkBytesIn	.3.1.6.109	リモートホストから受信したバイト数（カウンター、整数）
netcheckNetworkBytesOut	.3.1.6.110	リモートホストに送信したバイト数（カウンター、整数）
netcheckLocalScanErrors	.3.1.6.111	ローカルのスキャンエンジンプロセスでのエラーの発生回数（カウンター、整数）
netcheckNetworkScanErrors	.3.1.6.112	リモートのスキャンエンジンプロセスでのエラーの発生回数（カウンター、整数）
httpd	.3.1.7	drweb-httpdコンポーネントデータ
httpdState	.3.1.7.1	コンポーネントの現在の状態（整数****）
httpdExitCode	.3.1.7.2	前回の終了コード（エラーカタログのコードに対応する整数）
httpdExitTime	.3.1.7.3	前回の終了時刻（*UNIX時間*）
snmpd	.3.1.8	drweb-snmpdコンポーネントデータ
snmpdState	.3.1.8.1	コンポーネントの現在の状態（整数****）
snmpdExitCode	.3.1.8.2	前回の終了コード（エラーカタログのコードに対応する整数）
snmpdExitTime	.3.1.8.3	前回の終了時刻（*UNIX時間*）
clamd	.3.1.20	drweb-clamdコンポーネントデータ

パラメータ名	パラメータの OID	パラメータのタイプと説明
clamdState	.3.1.20.1	コンポーネントの現在の状態（整数****）
clamdExitCode	.3.1.20.2	前回の終了コード（エラーカタログのコードに対応する整数）
clamdExitTime	.3.1.20.3	前回の終了時刻（*UNIX時間*）
icapd	.3.1.21	drweb-icapdコンポーネントデータ
icapdState	.3.1.21.1	コンポーネントの現在の状態（整数****）
icapdExitCode	.3.1.21.2	前回の終了コード（エラーカタログのコードに対応する整数）
icapdExitTime	.3.1.21.3	前回の終了時刻（*UNIX時間*）
icapdConnectionsIn	.3.1.21.101	承認された受信接続の数（カウンター、整数）
icapdConnectionsCount	.3.1.21.102	現在開いている接続の数（カウンター、整数）
icapdOptions	.3.1.21.103	*OPTIONS*リクエストの数（カウンター、整数）
icapdReqmod	.3.1.21.104	*REQMOD*リクエストの数（カウンター、整数）
icapdRespmod	.3.1.21.105	*RESPMOD*リクエストの数（カウンター、整数）
icapdBad	.3.1.21.106	無効なリクエストの数（カウンター、整数）
smbspider	.3.1.40	drweb-smbspider-daemonコンポーネントデータ
smbspiderState	.3.1.40.1	コンポーネントの現在の状態（整数****）
smbspiderExitCode	.3.1.40.2	前回の終了コード（エラーカタログのコードに対応する整数）
smbspiderExitTime	.3.1.40.3	前回の終了時刻（*UNIX時間*）
smbspiderConnectionsIn	.3.1.40.101	開かれた接続の合計数（カウンター、整数）
smbspiderConnectionsCount	.3.1.40.102	現在開いている接続の数（カウンター、整数）

パラメータ名	パラメータの OID	パラメータのタイプと説明
smbspiderShareTable	.3.1.40.103	保護されたSamba共有リソースに関する統計
smbspiderShareEntry	.3.1.40.103.1	保護されているSamba共有リソースに関する情報（テーブル行全体、レコード）
smbspiderShareIndex	.3.1.40.103.1.1	保護されたSamba共有リソースのインデックス（序数）（整数）
smbspiderSharePath	.3.1.40.103.1.2	保護されたSamba共有リソースへのパス（文字列）
smbspiderShareConnectionsIn	.3.1.40.103.1.3	開かれた接続の合計数（カウンター、整数）
smbspiderShareConnectionsCount	.3.1.40.103.1.4	現在開いている接続の数（カウンター、整数）
gated	.3.1.41	drweb-gatedコンポーネントデータ
gatedState	.3.1.41.1	コンポーネントの現在の状態（整数****）
gatedExitCode	.3.1.41.2	前回の終了コード（エラーカタログのコードに対応する整数）
gatedExitTime	.3.1.41.3	前回の終了時刻（*UNIX時間*）
gatedInterceptedIn	.3.1.41.101	傍受した接続の数（カウンター、整数）
gatedInterceptedCount	.3.1.41.102	現在モニタリングされている接続の数（カウンター、整数）
maild	.3.1.42	drweb-maildコンポーネントデータ
maildState	.3.1.42.1	コンポーネントの現在の状態（整数****）
maildExitCode	.3.1.42.2	前回の終了コード（エラーカタログのコードに対応する整数）
maildExitTime	.3.1.42.3	前回の終了時刻（*UNIX時間*）
maildStat	.3.1.42.4	drweb-maildコンポーネントの動作の統計
maildStatNative	.3.1.42.4.1	コンポーネントの内部インターフェースdrweb-maildを介したメールスキャンの統計（傍受したSMTP、

パラメータ名	パラメータの OID	パラメータのタイプと説明
		POP3、IMAP 接続のスキャン中に SpIDer Gate が受信したメッセージ）
maildStatNativePassed	.3.1.42.4.1.1	受信できなかったメッセージの数（カウンター、整数）
maildStatNativeRepacked	.3.1.42.4.1.2	再圧縮されたメッセージの数（カウンター、整数）
maildStatNativeRejected	.3.1.42.4.1.3	拒否されたメッセージの数（カウンター、整数）
maildStatNativeFailed	.3.1.42.4.1.4	メッセージスキャンエラーの数（カウンター、整数）
maildStatNativeQueueSize	.3.1.42.4.1.5	キューライン、つまりインターフェースを介したスキャンを待機しているファイルの数（整数）
maildStatMilter	.3.1.42.4.2	drweb-maild コンポーネントのコンポーネントインターフェース *Milter* を介したメールスキャンの統計
maildStatMilterPassed	.3.1.42.4.2.1	受信できなかったメッセージの数（カウンター、整数）
maildStatMilterRepacked	.3.1.42.4.2.2	再圧縮されたメッセージの数（カウンター、整数）
maildStatMilterRejected	.3.1.42.4.2.3	拒否されたメッセージの数（カウンター、整数）
maildStatMilterFailed	.3.1.42.4.2.4	メッセージスキャンエラーの数（カウンター、整数）
maildStatMilterQueueSize	.3.1.42.4.2.5	キューライン、つまりインターフェースを介したスキャンを待機しているファイルの数（整数）
maildStatSpamc	.3.1.42.4.3	drweb-maild コンポーネントのコンポーネントインターフェース *Spamd* を介したメールスキャンの統計
maildStatSpamcPassed	.3.1.42.4.3.1	受信できなかったメッセージの数（カウンター、整数）
maildStatSpamcRepacked	.3.1.42.4.3.2	再圧縮されたメッセージの数（カウンター、整数）
maildStatSpamcRejected	.3.1.42.4.3.3	拒否されたメッセージの数（カウンター、整数）

パラメータ名	パラメータのOID	パラメータのタイプと説明
maildStatSpamcFailed	.3.1.42.4.3.4	メッセージスキャンエラーの数（カウンター、整数）
maildStatSpamcQueueSize	.3.1.42.4.3.5	キューライン、つまりインターフェースを介したスキャンを待機しているファイルの数（整数）
maildStatRspamc	.3.1.42.4.4	drweb-maildコンポーネントのコンポーネントインターフェース *Rspamd*を介したメールスキャンの統計
maildStatRspamcPassed	.3.1.42.4.4.1	受信できなかったメッセージの数（カウンター、整数）
maildStatRspamcRepacked	.3.1.42.4.4.2	再圧縮されたメッセージの数（カウンター、整数）
maildStatRspamcRejected	.3.1.42.4.4.3	拒否されたメッセージの数（カウンター、整数）
maildStatRspamcFailed	.3.1.42.4.4.4	メッセージスキャンエラーの数（カウンター、整数）
maildStatRspamcQueueSize	.3.1.42.4.4.5	キューライン、つまりインターフェースを介したスキャンを待機しているファイルの数（整数）
lookupd	.3.1.43	drweb-lookupdコンポーネントデータ
lookupdState	.3.1.43.1	コンポーネントの現在の状態（整数****）
lookupdExitCode	.3.1.43.2	前回の終了コード（エラーカタログのコードに対応する整数）
lookupdExitTime	.3.1.43.3	前回の終了時刻（*UNIX時間*）
antispam	.3.1.44	drweb-aseコンポーネントに関するデータ
antispamState	.3.1.44.1	コンポーネントの現在の状態（整数****）
antispamExitCode	.3.1.44.2	前回の終了コード（エラーカタログのコードに対応する整数）
antispamExitTime	.3.1.44.3	前回の終了時刻（*UNIX時間*）
cloudd	.3.1.50	drweb-clouddコンポーネントデータ

パラメータ名	パラメータのOID	パラメータのタイプと説明
clouddState	.3.1.50.1	コンポーネントの現在の状態（整数****）
clouddExitCode	.3.1.50.2	前回の終了コード（エラーカタログのコードに対応する整数）
clouddExitTime	.3.1.50.3	前回の終了時刻（*UNIX時間*）
meshd	.3.1.52	drweb-meshdコンポーネントデータ
meshdState	.3.1.52.1	コンポーネントの現在の状態（整数****）
meshdExitCode	.3.1.52.2	前回の終了コード（エラーカタログのコードに対応する整数）
meshdExitTime	.3.1.52.3	前回の終了時刻（*UNIX時間*）
lotus	.3.1.60	drweb-lotusコンポーネントデータ
lotusState	.3.1.60.1	コンポーネントの現在の状態（整数****）
lotusExitCode	.3.1.60.2	前回の終了コード（エラーカタログのコードに対応する整数）
lotusExitTime	.3.1.60.3	前回の終了時刻（*UNIX時間*）
macgui	.3.1.100	drweb-gui（macOS用）コンポーネントデータ
macguiState	.3.1.100.1	コンポーネントの現在の状態（整数****）
macguiExitCode	.3.1.100.2	前回の終了コード（エラーカタログのコードに対応する整数）
macguiExitTime	.3.1.100.3	前回の終了時刻（*UNIX時間*）
macspider	.3.1.102	drweb-spider（macOS用）コンポーネントデータ
macspiderState	.3.1.102.1	コンポーネントの現在の状態（整数****）
macspiderExitCode	.3.1.102.2	前回の終了コード（エラーカタログのコードに対応する整数）
macspiderExitTime	.3.1.102.3	前回の終了時刻（*UNIX時間*）

パラメータ名	パラメータのOID	パラメータのタイプと説明
macspiderWorkStatus	.3.1.102.101	コンポーネントの現在の動作モード（整数：0 - 未設定、1 - 読み込み中、2 - 実行中）
macfirewall	.3.1.103	drweb-firewall（macOS用）コンポーネントデータ
macfirewallState	.3.1.103.1	コンポーネントの現在の状態（整数****）
macfirewallExitCode	.3.1.103.2	前回の終了コード（エラーカタログのコードに対応する整数）
macfirewallExitTime	.3.1.103.3	前回の終了時刻（*UNIX時間*）
linuxgui	.3.1.200	drweb-gui（GNU/Linux用）コンポーネントデータ
linuxguiState	.3.1.200.1	コンポーネントの現在の状態（整数****）
linuxguiExitCode	.3.1.200.2	前回の終了コード（エラーカタログのコードに対応する整数）
linuxguiExitTime	.3.1.200.3	前回の終了時刻（*UNIX時間*）
linuxspider	.3.1.201	drweb-spider（GNU/Linux用）コンポーネントデータ
linuxspiderState	.3.1.201.1	コンポーネントの現在の状態（整数****）
linuxspiderExitCode	.3.1.201.2	前回の終了コード（エラーカタログのコードに対応する整数）
linuxspiderExitTime	.3.1.201.3	前回の終了時刻（*UNIX時間*）
linuxspiderWorkStatus	.3.1.201.101	コンポーネントの現在の動作モード（整数：0 - 未設定、1 - 読み込み中、2 - fanotify経由で実行中、3 - LKM経由で実行中）
linuxnss	.3.1.202	drweb-nss（GNU/Linux用）コンポーネントデータ
linuxnssState	.3.1.202.1	コンポーネントの現在の状態（整数****）
linuxnssExitCode	.3.1.202.2	前回の終了コード（エラーカタログのコードに対応する整数）
linuxnssExitTime	.3.1.202.3	前回の終了時刻（*UNIX時間*）

パラメータ名	パラメータの OID	パラメータのタイプと説明
linuxnssScannedFiles	.3.1.202.101	スキャン済みファイル数（カウンター、整数）
linuxnssScannedBytes	.3.1.202.102	スキャン済みバイト数（カウンター、整数）
linuxnssScanErrors	.3.1.202.103	スキャンエラーの発生回数（カウンター、整数）
linuxfirewall	.3.1.203	drweb-firewall（GNU/Linux 用）コンポーネントデータ
linuxfirewallState	.3.1.203.1	コンポーネントの現在の状態（整数 ****）
linuxfirewallExitCode	.3.1.203.2	前回の終了コード（エラーカタログのコードに対応する整数）
linuxfirewallExitTime	.3.1.203.3	前回の終了時刻（*UNIX時間*）
ctl	.3.1.300	drweb-ctl コンポーネントデータ
ctlState	.3.1.300.1	コンポーネントの現在の状態（整数 ****）
ctlExitCode	.3.1.300.2	前回の終了コード（エラーカタログのコードに対応する整数）
ctlExitTime	.3.1.300.3	前回の終了時刻（*UNIX時間*）
license	.3.2	ライセンスのステータス
licenseEsMode	.3.2.1	ライセンスが集中管理サーバーによって付与された（整数：0 - いいえ、1 - はい）
licenseNumber	.3.2.2	ライセンス番号（整数）
licenseOwner	.3.2.3	ライセンスの所有者（文字列）
licenseActivated	.3.2.4	ライセンスを有効化した日（*UNIX時間*）
licenseExpires	.3.2.5	ライセンス有効期限（*UNIX時間*）

*) 脅威のタイプ:

コード	脅威の種類
1	既知のウイルス（*known virus*）

コード	脅威の種類
2	疑わしいオブジェクト（*suspicious*）
3	アドウェア（*adware*）
4	ダイアラー（*dialer*）
5	ジョークプログラム（*joke program*）
6	リスクウェア（*riskware*）
7	ハッキングツール（*hacktool*）

**) URLのカテゴリー：

コード	脅威の種類
1	感染源（*infectionSource*）
2	非推奨（*notRecommended*）
3	アダルトコンテンツ（*adultContent*）
4	暴力（*violence*）
5	武器（*weapons*）
6	ギャンブル（*gambling*）
7	麻薬（*drugs*）
8	卑猥な表現など（*obsceneLanguage*）
9	チャット（*chats*）
10	テロリズム（*terrorism*）
11	無料メール（*freeEmail*）
12	ソーシャルネットワーク（*socialNetworks*）
13	著作権者からの申し立てによって登録されたURL（*ownerNotice*）
14	オンラインゲーム（*onlineGames*）
15	アノニマイザー（*anonymizers*）
16	仮想通貨マイニングプール（*cryptocurrencyMiningPools*）
17	求人情報サイト（*Jobs*）

コード	脅威の種類
20	ブラックリストに追加済み（*blackList*）

***) Dr.Webコンポーネントのコード:

コード	コンポーネント
1	Dr.Web ConfigD（`drweb-configd`）
2	Dr.Web Scanning Engine（`drweb-se`）
3	Dr.Web File Checker（`drweb-filecheck`）
4	Dr.Web Updater（`drweb-update`）
5	Dr.Web ES Agent（`drweb-esagent`）
6	Dr.Web Network Checker（`drweb-netcheck`）
7	Dr.Web HTTPD（`drweb-httpd`）
8	Dr.Web SNMPD（`drweb-snmpd`）
20	Dr.Web ClamD（`drweb-clamd`）
21	Dr.Web ICAPD（`drweb-icapd`）
40	SpIDer Guard for SMB（`drweb-smbspider-daemon`）
41	SpIDer Gate（`drweb-gated`）
42	Dr.Web MailD（`drweb-maild`）
43	Dr.Web LookupD（`drweb-lookupd`）
50	Dr.Web CloudD（`drweb-cloudd`）
52	Dr.Web MeshD（`drweb-meshd`）
60	Dr.Web for Lotus
100	`drweb-gui` for macOS
102	SpIDer Guard for macOS
103	Dr.Web Firewall for macOS
200	`drweb-gui` for GNU/Linux
201	SpIDer Guard（`drweb-spider`）

コード	コンポーネント
202	SpIDer Guard for NSS（drweb-nss）
203	Dr.Web Firewall for Linux（drweb-firewall）for GNU/Linux
300	Dr.Web Ctl（drweb-ctl）
400	Enterprise scanner（Dr.Web for UNIX Mail Serversのコンポーネントではありません）

****）コンポーネントの状態：

コード	ステータス
0	インストールされていません
1	インストールされているが開始されていない
2	起動中
3	実行中
4	終了中

変数の値を直接取得するには、snmpwalkユーティリティを使用します。

```
$ snmpwalk -Os -c <community> -v <SNMP version> <host address> <OID>
```

たとえば、ローカルホストで検出された脅威に関する統計を取得するには、次のコマンドを使用します（Dr.Web SNMPDの設定がデフォルト値に設定されている場合）。

```
$ snmpwalk -Os -c public -v 2c 127.0.0.1 .1.3.6.1.4.1.29690.2.2.1
```

Dr.Web MeshD

Dr.Web MeshDは、「ローカルクラウド」にインストールされたDr.Web for UNIX Mail Serversを持つホストを含むエージェントです。この製品は、インストール済みのDr.Web for UNIX製品にホストを接続します。このクラウドを利用すると、次のタスクを解決できます。

- 複数のクラウドホストによる他のファイルスキャンサービスの提供（スキャンコアサービス）。
- ウイルスデータベースの更新をクラウドホスト間で配布する。

インストール済みのDr.Web for UNIX製品にホストを接続するには、Dr.Web MeshDコンポーネントをすべてのホストにインストールする必要があります。このコンポーネントによってクラウドにホストが組み込まれます。クラウド内のホストの権限と、ホストが使用するクラウド機能は、Dr.Web MeshD設定で簡単に設定できます。

データは、保護されたSSHチャネルを介して他のクラウドホストと共有されます。

動作原理

このセクションの内容

- 接続タイプ
- 動作モード
- サービス

Dr.Web MeshDは、Dr.Web for UNIX Mail Serversがインストールされているホストと他のクラウドホスト間のインタラクションを調整します。

接続タイプ

Dr.Web MeshDでは次の接続タイプを使用します。

- *クライアント（サービス）* - Dr.Web MeshDが他のクラウドホストに接続するために使用されます。これらのホストは、指定したホストによって提供されるサービスのクライアントです。

  > ホスト上で動作し、同じホスト上で動作するDr.Web MeshDを介してクラウド提供のサービスにアクセスするDr.Web for UNIX Mail Serversのコンポーネントは、ローカルのUNIXソケットを介してクライアントに接続します。その場合、クライアント接続は使用されません。

- *パートナー（ピアツーピア）* - ピア（サービス内）パートナークラウドホストとのインタラクションのためにDr.Web MeshDによって使用されます。通常、このような横方向の接続は、クラウドでやり取りする際の負荷の軽減や分散の他、クラウドホストの同期に使用されます。
- *アップリンク* - このホスト（クライアント）をクラウドホスト（サービスプロバイダー）に接続する際に、Dr.Web MeshDによって使用されます（ウイルスデータベースの更新の配布、スキャンのためのファイル転送など）。

使用する3つの種類の接続はすべて、それぞれのクラウドサービスに対して個別に設定されます。さらに同じホストを、サービス内でクライアントの要求（最新の更新を提供するなど）を処理するサーバーとして、および別のサービス内（リモートファイルスキャンなど）のクライアントとして設定できます。

クラウド内では、ホストは安全なSSHプロトコルを介して承認済みのインタラクションを実行します。つまり、ホスト間通信のすべての側面が常に相互認証されます。認証には、RFC 4251に従って*ホストキー*が使用されます。ローカルコンポーネントからのクライアント接続は常に信頼済みと見なされます。

動作モード

Dr.Web MeshDはデーモンモードで動作させることも、ローカルホストにある他のDr.Web for UNIX Mail Serversコンポーネントからの要求で実行することもできます。Dr.Web MeshDがクライアント接続を提供するように設定されており（つまり、`ListenAddress`パラメータが空でない）、少なくとも1つのサービスがアクティブ化されている場合、Dr.Web MeshDはデーモンとして起動し、クライアントからの接続を待機します。またDr.Web MeshDは、リクエストに応じて、たとえば次のコマンドを実行するときに、ローカルホストで有効化できます。

```
$ drweb-ctl update --local-cloud
```

Dr.Web MeshDがクライアント接続を処理するように設定されておらず（`ListenAddress`パラメータが空で）、`IdleTimeLimit`パラメータで指定された期間中にDr.Web MeshDへのリクエストがない場合、コンポーネントは自動的に終了します。

サービス

更新を交換する（更新）

このサービスを使用すると、ホストはウイルスやその他のデータベースの更新をサブスクライブし、更新に関する通知を送信し、クラウドホスト間で更新ファイルをアップロードして共有できます。サービス設定は、`Update*`パラメータを使用して設定できます。

標準のサービス使用法では、Dr.Web MeshDがインストールされており、会社のローカルネットワーク内の複数のマシン（サービスのクライアント）で更新を取得する機能が有効になっていると想定しています。一般的なクライアント設定は次のとおりです。

```
…
[MeshD]
UpdateChannel = On
UpdateUplink = <server address>
ListenAddress =
…
[Update]
UseLocalCloud = Yes
…
```

更新を配布するローカルサーバーでは、次の設定が指定されています。

```
UpdateChannel = On
UpdateUplink =
ListenAddress = <address>:<port>
```

ここで、クライアントのアップリンク接続の*<server address>*は、サーバーホストがクライアント接続を管理するために使用する*<address>*および*<port>*を参照する必要があります。

ホストの1つが更新サーバー（ローカルクラウドの外部：Dr.Web GUS更新サーバーまたは集中管理サーバー）から更新されている場合、ホストは必要なすべてのクライアントに通知を送信し（ホストが更新交換サーバーとして設定されている場合）、サーバーホストに、ホストから配布可能なファイルの新しいリストを送信します。通知を受信すると、クライアントホストはサーバーからの更新のダウンロードをリクエストできます。次に、クライアントからファイルをリクエストしてローカルに保存するか、サーバーからファイルをリクエストした他のクライアントに送信できます。

このシナリオでは、クライアントが異なるタイミングでDr.Web GUSにリクエストを送信するため、更新の遅延が短縮します。そのため、最初に更新されたクライアントは、必要なすべてのクラウドホストに最新の更新ファイルをすぐに配信します。また、トラフィック量とDr.Web GUSの負荷も軽減されます。

ローカルクラウドを更新の配布に使用する場合、Dr.Web MeshDに加えて、ホストにはDr.Web Updaterコンポーネントが含まれている必要があることに注意してください。

リモートファイルスキャン（エンジン）

このサービスでは、Dr.Web Scanning Engineを使用してリモートファイルをスキャンできます。クライアントホストはスキャン用のファイルをサーバーホストに送信し、サーバーホストはファイルスキャン用のサービスを提供します。一般的なクライアント設定は次のとおりです。

```
…
[MeshD]
EngineChannel = On
EngineUplink = <server address>
ListenAddress =
…
```

ローカルスキャンサーバーでは、次の設定が指定されています。

```
EngineChannel = On
EngineUplink =
ListenAddress = <address>: <port>
```

ここで、クライアントのアップリンク接続の*<server address>*は、サーバーホストがクライアント接続を管理するために使用する*<address>*および*<port>*を参照する必要があります。

スキャンするファイルを送信する（ファイル）

この機能は使用されません（リモートスキャンは*Engine*サービス内で提供されます）。

URLチェック

このサービスでは、サーバーホストを使用して、潜在的に危険で推奨されないカテゴリーに属するURLをチェックできます。クライアントホストは、チェックするURLをサーバーホストに送信します。一般的なクライアント設定は次のとおりです。

```
…
[MeshD]
UrlChannel = On
UrlUplink = <server address>
ListenAddress =
…
```

URLチェックに使用されるローカルサーバーでは、次の設定が指定されています。

```
UrlChannel = On
UrlUplink =
ListenAddress = <address>:<port>
```

ここで、クライアントのアップリンク接続の*<server address>*は、サーバーホストがクライアント接続を管理するために使用する*<address>*および*<port>*を参照する必要があります。

コマンドライン引数

Dr.Web MeshDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-meshd [<parameters>]
```

Dr.Web MeshDは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了するように指示します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-meshd --help
```

このコマンドは、Dr.Web MeshDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-meshd` コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Servers の統合された設定ファイルの `[MeshD]` セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、`[Root]` セクションの `DefaultLogLevel` パラメータの値が使用されます。 デフォルト値：`Notice`
`Log` *{log type}*	コンポーネントのロギング方式。 デフォルト値：`Auto`
`ExePath` *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*`/bin/drweb-meshd` • GNU/Linux の場合：`/opt/drweb.com/bin/drweb-meshd` • FreeBSD の場合：`/usr/local/libexec/drweb.com/bin/drweb-meshd`
`DebugSsh` *{Boolean}*	ロギングレベルが `LogLevel = Debug` の場合、ホスト上で動作している Dr.Web MeshD が送受信した SSH プロトコルメッセージ（メッセージとデータの転送に使用）のロギングを実行します。 デフォルト値：`No`
`IdleTimeLimit` *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（`10s`）から30日（`30d`）まで。 `None` 値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、`SIGTERM` シグナルは送信されません。 デフォルト値：`30s`
`DnsResolverConfPath` *{path to file}*	ドメイン名パーミッション（DNS リゾルバ）のサブシステム設定ファイルへのパス。 デフォルト値：`/etc/resolv.conf`
`ListenAddress` *<IP address>:<port>*	コンポーネントがクラウドホストからの接続の受信を待機しているクライアント接続のネットワークソケット（アドレスとポート）。これらのホストは、このクラウドホストによって提供されるサービスのクライアントです。

パラメータ	説明
	コンポーネントがIPv6インターフェースをリッスンし、IPv6経由でクラウドクライアントホストを検出するためには、このパラメータを設定する必要があります。 値が指定されていない場合、コンポーネントはクライアントからの要求を受け取りません。 デフォルト値：*（未設定）*
`UpdateChannel` *{On / Off}*	このホスト上で動作するDr.Web MeshDコンポーネントを有効または無効にし、クラウドのホスト間でウイルスデータベースの更新を交換します（たとえば、他のクラウドホストからウイルスデータベースの更新を取得し、新しい更新をクラウドに送信します）。 このパラメータがOnに設定されている場合、コンポーネントDr.Web MeshDはDr.Web ConfigD設定デーモンによって自動的に起動されます。 デフォルト値：`On`
`UpdateUplink` *{address}*	このホストにURLチェックサービスを提供するサーバーとして機能するDr.Web MeshDの上位ホストのアドレス。 使用可能な値： • *指定なし* - サーバーがサービスに設定されておらず、Dr.Web MeshDはどこにも接続されません。 • *<IP address>:<port>* - Dr.Web MeshDは指定されたアドレスとポートでサーバーに接続します。 • `dns:`*<service name>*`[:`*<domain>*`]` - サーバーのアドレスとポートは、*<domain>*DNSドメインのSRVレコードを検索することで指定されます。*<domain>*が指定されていない場合、DNSリゾルバ設定ファイルのドメインが使用されます（パスは`ResolverConfPath`で指定されます）。このドメインは、最後に検出されたドメインに応じて、`search`または`domain`フィールドから取得されます。 • `discover` - *検出*メカニズムによって上位ホストを検索します。 デフォルト値：*（指定なし）*
`UpdateDebugIpc` *{Boolean}*	ロギングレベルが`LogLevel = Debug`の場合は、更新の交換サービスのログにデバッグ情報を出力します。 デフォルト値：`No`
`UpdateTraceContent` *{Boolean}*	ロギングレベルが`LogLevel = Debug`の場合は、更新の交換サービスのログに送信済みデータを出力します。 デフォルト値：`No`
`FileChannel` *{On / Off}*	このホストで動作するDr.Web MeshDコンポーネントがファイル交換に参加できるようにするオプションを有効または無効にします。 このパラメータがOnに設定されている場合、コンポーネントDr.Web MeshDはDr.Web ConfigD設定デーモンによって自動的に起動されます。 デフォルト値：`On`
`FileUplink` *{address}*	このホストのファイルをスキャンするサーバーとして機能するDr.Web MeshDの上位ホストのアドレス。

パラメータ	説明
	使用可能な値： • *指定なし* - サーバーがサービスに設定されておらず、Dr.Web MeshDはどこにも接続されません。 • *<IP address>:<port>* - Dr.Web MeshDは指定されたアドレスとポートでサーバーに接続します。 • `dns:<service name>[:<domain>]` - サーバーのアドレスとポートは、*<domain>*DNSドメインのSRVレコードを検索することで指定されます。*<domain>*が指定されていない場合、DNSリゾルバ設定ファイルのドメインが使用されます（パスは`ResolverConfPath`で指定されます）。このドメインは、最後に検出されたドメインに応じて、`search`または`domain`フィールドから取得されます。 • `discover` - *検出*メカニズムによって上位ホストを検索します。 デフォルト値：*（指定なし）*
`FileDebugIpc` *{Boolean}*	ロギングレベルが`LogLevel = Debug`の場合は、ファイル交換サービスのログにデバッグ情報を出力します。 デフォルト値：`No`
`EngineChannel` *{On / Off}*	このホストで動作するDr.Web MeshDコンポーネントがスキャンエンジンサービスの提供に参加できるようにするオプションを有効または無効にします。 このパラメータがOnに設定されている場合、コンポーネントDr.Web MeshDはDr.Web ConfigD設定デーモンによって自動的に起動されます。 デフォルト値：`On`
`EngineUplink` *{address}*	このホストにスキャンエンジンサービスを提供するスキャンサーバーとして機能するDr.Web MeshDの上位ホストのアドレス。 使用可能な値： • *指定なし* - サーバーがサービスに設定されておらず、Dr.Web MeshDはどこにも接続されません。 • *<IP address>:<port>* - Dr.Web MeshDは指定されたアドレスとポートでサーバーに接続します。 • `dns:<service name>[:<domain>]` - サーバーのアドレスとポートは、*<domain>*DNSドメインのSRVレコードを検索することで指定されます。*<domain>*が指定されていない場合、DNSリゾルバ設定ファイルのドメインが使用されます（パスは`ResolverConfPath`で指定されます）。このドメインは、最後に検出されたドメインに応じて、`search`または`domain`フィールドから取得されます。 • `discover` - *検出*メカニズムによって上位ホストを検索します。 デフォルト値：*（指定なし）*
`EngineDebugIpc` *{Boolean}*	ロギングレベルが`LogLevel = Debug`の場合は、スキャンサービスのログにデバッグ情報を出力します。 デフォルト値：`No`
`UrlChannel` *{On / Off}*	このホストで動作するDr.Web MeshDコンポーネントがURLチェックサービスの提供に参加できるようにするオプションを有効または無効にします。

パラメータ	説明
	デフォルト値：`On`
`UrlUplink` *{address}*	このホストにURLチェックサービスを提供するサーバーとして機能するDr.Web MeshDの上位ホストのアドレス。 使用可能な値： • *指定なし* - サーバーがサービスに設定されておらず、Dr.Web MeshDはどこにも接続されません。 • *<IP address>:<port>* - Dr.Web MeshDは指定されたアドレスとポートでサーバーに接続します。 • `dns:` *<service name>*`[:` *<domain>*`]` - サーバーのアドレスとポートは、*<domain>*DNSドメインのSRVレコードを検索することで指定されます。*<domain>*が指定されていない場合、DNSリゾルバ設定ファイルのドメインが使用されます（パスは`ResolverConfPath`で指定されます）。このドメインは、最後に検出されたドメインに応じて、`search`または`domain`フィールドから取得されます。 • `discover` - *検出*メカニズムによって上位ホストを検索します。 デフォルト値：*（指定なし）*
`DiscoveryResponderPort` *{port number}*	上位ホストがUDPプロトコルを介して設定されたクライアントの要求に応答するポート。 *検出*メカニズムは、`ListenAddress`値が設定されている場合にのみ有効になります。 デフォルト値：`18008`
`UrlDebugIpc` *{Boolean}*	ロギングレベルが`LogLevel = Debug`の場合、デバッグ情報をURLチェックサービスのログに出力します。 デフォルト値：`No`

Dr.Web for UNIX Mail Servers の現在のバージョンでは、*ファイル*送信サービスは使用されません。代わりに、*Engine*スキャンエンジンサービスを使用します。

Dr.Web URL Checker

Dr.Web URL Checkerは、URLが望ましくないカテゴリーや悪意のあるカテゴリーに属していないかどうかをチェックするために、他のコンポーネントが使用する補助コンポーネントです。

Dr.Web URL Checkerは次のコンポーネントによって使用されます。

- Dr.Web HTTPD
- Dr.Web MeshD
- SpIDer Gate
- Dr.Web MailD

望ましくないリンクや潜在的に危険なリンクがないかメールメッセージをチェックするために、このコンポーネントは標準の*Milter*、*Spamd*および*Rspamd*を使用してメールサーバー（MTA）と統合されます（これらのインターフェース

は通常、SpamAssassinフィルターを使用します）。チェックを有効にするには、Luaで適切なスクリプトを作成する必要があります（スクリプトの例については、Luaでのメール処理を参照してください）。

動作原理

Dr.Web URL Checkerコンポーネントは、URLが望ましくないカテゴリーや潜在的に危険なカテゴリーに属していないかチェックするために、他のコンポーネントが使用します。

このチェックは、専用のリンクベースを使用するか、Dr.Web CloudDサービスを使用して実行できます。Dr.Web CloudDサービスを使用するには、次のコマンドを実行します。

```
$ drweb-ctl cfset Root.UseCloud Yes
```

ユーザーがDr.Web URL Checkerを起動することはできません。他のコンポーネントのリクエストに応じて、Dr.Web ConfigD構成管理デーモンによって起動されます。

コマンドライン引数

Dr.Web URL Checkerを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-urlcheck [<parameters>]
```

Dr.Web URL Checkerは次のオプションを処理できます。

パラメータ	説明
`--help`	機能：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 短縮形：`-h` 引数：なし
`--version`	機能：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 短縮形：`-v` 引数：なし

例：

```
$ /opt/drweb.com/bin/drweb-urlcheck --help
```

このコマンドは、URL Checkerに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの

動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-urlcheck`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[Urlcheck]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのログレベル。 パラメータの値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。 デフォルト値：`Notice`
`Log` *{log type}*	コンポーネントのログ方式。 デフォルト値：`Auto`
`ExePath` *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*`/bin/drweb-urlcheck` • GNU/Linuxの場合：`/opt/drweb.com/bin/drweb-urlcheck` • FreeBSDの場合：`/usr/local/libexec/drweb.com/bin/drweb-urlcheck`
`RunAsUser` *{UID / user name}*	その権限によりコンポーネントを実行するユーザーの名前。このユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「`name:`」というプレフィックスを付けて指定します。たとえば、`RunAsUser = name:123456`です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：`drweb`
`IdleTimeLimit` *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（`10s`）から30日（`30d`）まで。 `None`値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、`SIGTERM`シグナルは送信されません。 デフォルト値：`30s`

Dr.Web CloudD

Dr.Web CloudDコンポーネントはDr.Web Cloud（Doctor Webのクラウドサービス）を参照します。Dr.Web Cloudサービスは、検出された脅威に関する最新情報をすべてのDr.Webアンチウイルスソリューションから収集し、ユーザーが望ましくないWebサイトにアクセスするのを防ぎ、Dr.Webウイルスデータベースに記述のない新しい脅威を含んだ感染ファイルからOSを保護します。さらに、Dr.Web Cloudサービスを使用すると、Dr.Web Scanning Engineスキャンエンジンやインターネットへのアクセスを監視するコンポーネントの誤検知の可能性が少なくなります。

動作原理

このコンポーネントは、指定ファイルのコンテンツにローカルのDr.Web Scanning Engineが把握していない脅威がないかスキャンし、指定URLがDoctor WebのWebリソースの定義済みカテゴリーに属するかどうかを確認するよう、Doctor Web Dr.Web Cloudサービスに指示します。またこのコンポーネントは、感染ファイルの検出に関する統計情報と、Dr.Web for UNIX Mail Serversを実行しているOSに関する情報を定期的にDr.Web Cloudに送信します。

Dr.Web CloudDは設定デーモンによって自動的に実行されます。このコンポーネントは、ユーザーまたはDr.Web for UNIX Mail Serversコンポーネントの1つからコマンドを受信したときに実行されます。

このコンポーネントは、ネットワークトラフィックとURL SpIDer Gate

さらに、このコンポーネントはコマンドラインDr.Web Ctl（`drweb-ctl`コマンドによって起動）からのDr.Web for UNIX Mail Servers管理ユーティリティのコマンドによるファイルのスキャン中に使用されます。脅威が検出されると、Dr.Web Scanning Engineスキャンエンジンがファイルに関するレポートをDr.Web Cloudに送信します。

コマンドライン引数

Dr.Web CloudDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-cloudd [<parameters>]
```

Dr.Web CloudDは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-cloudd --help
```

このコマンドは、Dr.Web CloudDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じて、Dr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-cloudd`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[CloudD]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel`	コンポーネントのロギングレベル。

パラメータ	説明
{logging level}	パラメータの値が指定されていない場合は、[Root]セクションのDefaultLogLevelパラメータの値が使用されます。 デフォルト値：Notice
Log *{log type}*	コンポーネントのロギング方式。 デフォルト値：Auto
ExePath *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*/bin/drweb-cloudd • GNU/Linuxの場合：/opt/drweb.com/bin/drweb-cloudd • FreeBSDの場合：/usr/local/libexec/drweb.com/bin/drweb-cloudd
RunAsUser *{UID / user name}*	その権限によりコンポーネントを実行するユーザーの名前。このユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「name:」というプレフィックスを付けて指定します。たとえば、RunAsUser = name:123456です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：drweb
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：1h
PersistentCache *{On / Off}*	Dr.Web Cloud応答のキャッシュをディスクへ保存することを有効または無効にします。 デフォルト値：Off
DebugSdk *{Boolean}*	詳細なDr.Web Cloudメッセージをデバッグレベルでログファイルに含めるかどうかを示します（LogLevel = DEBUG）。 デフォルト値：No

Dr.Web LookupD

Dr.Web LookupDコンポーネントは、LDAPプロトコルを使用してデータを取得するために、外部ソース（LDAPプロトコルのインタラクションをサポートする、テキストファイル、リレーショナルデータベース、ディレクトリサービス）を参照するように設計されています。受信したデータはルールで使用され、それに従ってネットワーク接続がスキャンされます（ユーザーの承認のチェックなど）。このデータは、特定の条件が満たされた場合にURLへのアクセスをブロックするためにも使用されます。

コンポーネントの設定では、いくつかのデータソースに接続するためのパラメータを指定できます。Dr.Web LookupDは、Dr.Web for UNIX Mail Serversのいずれかのコンポーネントからデータリクエストを受信した場合にのみ、必要なデータソースに接続します。

Dr.Web LookupDでは次のデータソースへの参照がサポートされます。

- テキストファイル（*AllMatch*、*Mask*、*Regex*、*Cidr*モードで）。
- リレーショナルデータベース（MySQL、PostgreSQL、SQLite）。
- Redisデータストレージ。
- ディレクトリサービス（Active Directoryや、LDAP経由のアクセスを提供するその他のサービス、たとえばOpenLDAPなど）。

LDAPプロトコルによるデータの共有は、オープンチャネルまたは保護されたチャネル経由でSSL/TLSを適用することで実行できます。安全な接続を使用するには、Dr.Web LookupDに適切なSSL証明書とキーを提供する必要があります。SSLキーと証明書を生成する必要がある場合は、`openssl`ユーティリティを使用できます。`openssl`ユーティリティを使用して証明書とプライベートキーを生成する方法の例については、付録E. SSL証明書を生成するのセクションを参照してください。

動作原理

このコンポーネントは、テキストファイル、リレーショナルデータベース、ネットワークストレージ、LDAPプロトコルをサポートするディレクトリサービス（Active Directoryなど）にデータをリクエストするように設計されています。受信したデータ（ユーザーのIDや権限など）は、Dr.Web for UNIX Mail Serversのコンポーネントに転送され、さまざまなルールでスキャンに使用されます（ユーザーがリクエストしたURLなどにアクセスを許可するなど）。

このマニュアルではリレーショナルデータベース、Redisストレージ、ディレクトリサービス、LDAPプロトコルの動作原理については説明しません。必要に応じて、対応する参考資料を参照してください。

Dr.Web LookupDコンポーネントは、必要に応じて（データのリクエストを受信するときに）Dr.Web ConfigD設定デーモンによって自動的に起動されます。

特定のコンポーネントからのデータ受信に関するリクエストが到着すると、Dr.Web ConfigD設定デーモンがDr.Web LookupDを起動します（起動されていない場合）。次に、コンポーネントはリクエストされたデータソースからリクエストを実行し、レスポンスを返します。リクエストに応じて応答は異なり、所定の検索条件に従ってデータソースから取得された、特定の検索条件を満たす文字列のリスト、または所定の条件に一致する文字列が検索結果に含まれているかどうかを示す論理値（trueまたはfalse）で構成されます。

Dr.Web LookupDの設定で指定できるデータソースの数に制限はありません。クライアントコンポーネントは、データ取得リクエストを作成するときに、データソースを指定する必要があります。Dr.Web LookupDは起動後し

ばらくの間、新しいリクエストを待機します。それ以上リクエストがない場合は、待機期間が経過した後、Dr.Web LookupDは自動的にシャットダウンします。

Dr.Web for UNIX Mail Serversのその他のコンポーネントは、Dr.Web LookupDを使用するための基本の動作として、それらのコンポーネントの動作ルールで指定されている条件の妥当性を確認するために必要なデータを取得します。ルールの適用可能性と条件の妥当性をチェックするときに、Dr.Web LookupDに対するデータリクエストが自動的に実行されます。

テキストファイル処理の特性

1. テキストファイルを処理するとき、文字列の先頭と末尾のスペースは破棄されます。空白行と最初の文字が「#」で始まる行は無視されます。
2. テキストファイルは不変のデータソースと見なされ、その内容はメモリ内に完全にキャッシュされます。さらに、検証のためのこれらのファイルに対するリクエストの結果もキャッシュされます。そのため、ソースファイルが変更されている場合は、`drweb-ctl`の`reload`コマンドなどを使用してDr.Web ConfigDコンポーネントにHUPシグナルを送信することで、Dr.Web LookupDに設定を再読み込みさせる必要があります。

MySQL接続の状況

MySQL接続の前に、MySQLファイル設定の`[client]`セクションのパラメータがデフォルトで読み取られます（ファイル検索は、`/etc/my.cnf`、`/etc/mysql/my.cnf`、`/etc/alternatives/my.cnf`のパスで行われます）。

コマンドライン引数

Dr.Web LookupDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-lookupd [<parameters>]
```

Dr.Web LookupDは次のパラメータを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了するように指示します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了するように指示します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-lookupd --help
```

このコマンドは、Dr.Web LookupDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-lookupd`コマンドを使用します。

設定パラメータ

このセクションの内容

- コンポーネントパラメータ
- データソースセクション
- 新しいデータソース用のセクションの追加

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[LookupD]`セクションで指定されている設定パラメータを使用します。

コンポーネントパラメータ

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。 デフォルト値：`Notice`
`Log` *{log type}*	コンポーネントのロギング方式。 デフォルト値：`Auto`
`ExePath` *{path to file}*	コンポーネントの実行パス。 デフォルト値：*<opt_dir>*`/bin/drweb-lookupd` • GNU/Linuxの場合：`/opt/drweb.com/bin/drweb-lookupd` • FreeBSDの場合：`/usr/local/libexec/drweb.com/bin/drweb-lookupd`
`RunAsUser` *{UID / user name}*	このパラメータは、コンポーネントを実行するユーザー名を決定します。ユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「`name:`」というプレフィックスを付けて指定します。たとえば、`RunAsUser = name:123456`です。

パラメータ	説明
	ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：drweb
IdleTimeLimit *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（10s）から30日（30d）まで。 None値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、SIGTERMシグナルは送信されません。 デフォルト値：30s
DebugLibldap *{Boolean}*	libldapライブラリのデバッグメッセージもデバッグレベルでログファイルに含めるかどうかを示します（つまり、LogLevel = DEBUGの場合）。 デフォルト値：No
LdapCheckCertificate *{No / Allow / Try / Yes}*	SSL/TLS経由のLDAP接続の証明書検証モード。 使用可能な値： • No - サーバーの証明書を要求しない。 • Allow - サーバーの証明書を要求する。証明書が提供されない場合、セッションは通常どおりに続行されます。サーバーの証明書が提供されてもスキャンできない（対応するルート証明書が見つからない）場合、その証明書は無視され、セッションは通常どおりに続行されます。 • Try - サーバーの証明書を要求する。証明書が提供されない場合、セッションは通常どおりに続行されます。サーバーの証明書が提供されても確認できない（対応するルート証明書が見つからない）場合、セッションは終了します。 • Yes - サーバーの証明書を要求する。証明書が提供されないか、スキャンできない場合（対応するルート証明書が見つからない場合）、セッションは終了します。 LDAPデータソースの場合、この証明書検証モードは、ldaps://スキームまたはStartTLS拡張機能が使用されるときのURLの処理方法に影響します。ADデータソースの場合は、対応するセクションでUseSSL=Yesが指定されていると（以下参照）、サーバーへの接続に影響します。 デフォルト値：Yes
LdapCertificatePath *{path to file}*	安全なSSL/TLS接続を介したLDAPサーバーへの接続（Active Directory）に使用されるSSL証明書へのパス。 証明書ファイルとプライベートキーファイル（後述のパラメータで指定されます）は、一致するペアを形成する必要があります。 デフォルト値：*（未設定）*
LdapKeyPath *{path to file}*	安全なSSL/TLS接続を介したLDAPサーバーへの接続（Active Directory）に使用されるプライベートキーへのパス。 証明書ファイルとプライベートキーファイル（上記のパラメータで指定されます）は、一致するペアを形成する必要があります。

パラメータ	説明
	デフォルト値：*（未設定）*
`LdapCaPath` *{path}*	SSL/TLSを介したLDAPプロトコルによるデータの共有に信頼して使用できる、信頼できるルート証明書のシステムリストを含むディレクトリまたはファイルへのパス。 デフォルト値：*<path to the list of trusted certificates>*。パスは、お使いのGNU/Linuxディストリビューションに依存します。 • Astra Linux、Debian、Linux Mint、SUSE Linux、Ubuntuの場合、通常はパス`/etc/ssl/certs/`です。 • CentOSとFedoraの場合はパス`/etc/pki/tls/certs/ca-bundle.crt`です。 • 他のディストリビューションでは、コマンド`openssl version -d`の実行結果によってパスを定義できます。 • コマンドが使用できない場合、またはOSディストリビューションを特定できない場合は、値`/etc/ssl/certs/`が使用されます。
`DbIdleTimeout` *{time interval}*	データベース（またはRedisストレージ）への確立された接続がアイドル状態になっている場合に、その接続を切断するまでのタイムアウト期間。 デフォルト値：`5m`
`MysqlDefaultConn` *{URL}*	MySQLデータベースに接続するためのパラメータをデフォルトで設定するURI。 使用可能な値： • `tcp://[<user>[:<password>]@][<host>][:<port>][/<database name>][?<parameter>=<value>[&...]]` • `unix://[<user>[:<password>]@]<path to socket>[:<database name>][?<parameter>=<value>[&...]]` URI要件に注意してください。 デフォルト値：*（未設定）*
`PqDefaultConn` *{URL}*	PostgreSQLデータベースに接続するためのパラメータをデフォルトで設定するURI。 使用可能な値： • `tcp://[<user>[:<password>]@][<host>][:<port>][/<database name>][?<parameter>=<value>[&...]]` • `unix://[<user>[:<password>]@]<path to socket>[:<database name>][?<parameter>=<value>[&...]]` URI要件に注意してください。 デフォルト値：*（未設定）*
`SqliteDefaultConn` *{path to file}*	デフォルトのSQLiteデータベースファイルへのパス（`file://`スキームプレフィックスを指定）。 デフォルト値：*（未設定）*
`RedisDefaultConn` *{URL}*	Redisデータベースへの接続パラメータをデフォルトで設定するURLです。

パラメータ	説明
	使用可能な値： • `tcp://[<password>@][<host>][:<port>][/<database index>]` • `unix://[<password>@]<socket path>[:<database index>]` URI要件に注意してください。 デフォルト値：*（未設定）*

データベース接続のURI要件

1. `tcp:`および`unix:`スキームプレフィックスのみを使用します（ローカルUNIXソケットの場合）。データベース固有のプレフィックス（`postgresql:`および`mysql:`など）はサポートしていません。SQLiteデータベースファイルへのパスは、`file://`スキームプレフィックスで指定されます。
2. *<host>*フィールドがURIで指定されていないか、`localhost`ホストが指定されている場合、ホストアドレス`127.0.0.1`が置き換えられます。この場合、MySQLおよびPostgreSQLデータベースでは、ネットワーク接続が指定されているにもかかわらず、*ローカルUNIXソケットを介して*デフォルトで接続が確立されます。
3. URIフィールド（*<user>*、*<password>*、*<database name>*など）または接続パラメータ文字列に特殊文字（スペース、列など）が含まれている場合は、次のように16進コーディングを使用します。たとえば、
   - スペース - "%20"
   - ':' - "%3A"
   - '/' - "%2F"
   - '@' - "%40"
   - '%' - "%25"
4. MySQLの場合、接続パラメータ文字列には次のパラメータのみを含めることができます。

パラメータ名	データベースドキュメントの文字・記号	タイプ	説明
`init`	*MYSQL_INIT_COMMAND*	文字列	データベースへの接続後に実行するSQLコマンド
`compression`	*MYSQL_OPT_COMPRESS*	論理	データ圧縮を使用する
`connect-timeout`	*MYSQL_OPT_CONNECT_TIMEOUT*	整数	未使用の接続を切断するためのタイムアウト（秒）
`reconnect`	*MYSQL_OPT_RECONNECT*	論理	自動再接続を許可または拒否する
`read-timeout`	*MYSQL_OPT_READ_TIMEOUT*	整数	サーバーからのパケット受信のタイムアウト（秒）
`write-timeout`	*MYSQL_OPT_WRITE_TIMEOUT*	整数	パケットをサーバーに送信するためのタイムアウト（秒）

パラメータ名	データベースドキュメントの文字・記号	タイプ	説明
charset	*MYSQL_SET_CHARSET_NAME*	文字列	デフォルトの接続に使用される文字エンコーディングの名前
plugin-dir	*MYSQL_PLUGIN_DIR*	文字列	プラグインを格納するサーバーのディレクトリへのパス
nonblock	*MYSQL_OPT_NONBLOCK*	整数	ブロック以外のI/O操作のスタックサイズ
ssl-key	*MYSQL_OPT_SSL_KEY*	文字列	安全な接続を確立するために使用されるプライベートキー（PEM形式）へのパス
ssl-cert	*MYSQL_OPT_SSL_CERT*	文字列	安全な接続を確立するために使用されるパブリックキー証明書（PEM形式）へのパス
ssl-ca	*MYSQL_OPT_SSL_CA*	文字列	信頼できるCA証明書を含むファイル（PEM形式）へのパス
ssl-capath	*MYSQL_OPT_SSL_CAPATH*	文字列	信頼できるCA証明書を含むディレクトリへのパス（PEM形式）
ssl-cipher	*MYSQL_OPT_SSL_CIPHER*	文字列	安全な接続でサポートされている暗号化アルゴリズムのリスト
ssl-crl	*MYSQL_OPT_SSL_CRL*	文字列	失効した証明書を含むファイル（PEM形式）へのパス
ssl-crlpath	*MYSQL_OPT_SSL_CRLPATH*	文字列	失効した証明書を含むディレクトリへのパス（PEM形式）
ssl-fp	*MARIADB_OPT_SSL_FP*	文字列	有効なサーバー証明書のSHA1ハッシュ
ssl-fp-list	*MARIADB_OPT_SSL_FP_LIST*	文字列	有効なサーバー証明書のSHA1ハッシュを含むファイルへのパス
tls-passphrase	*MARIADB_OPT_TLS_PASSPHRASE*	文字列	パスワードで保護されたクライアントプライベートキーのパスワード

パラメータ名	データベースドキュメントの文字・記号	タイプ	説明
`tls-version`	*MARIADB_OPT_TLS_VERSION*	文字列	サポートされるTLSバージョンのリスト
`server-verify-cert`	*MYSQL_OPT_SSL_VERIFY_SERVER_CERT*	論理	サーバー証明書の検証を許可または禁止する
`server-public-key-path`	*MYSQL_SERVER_PUBLIC_KEY*	文字列	RSAサーバーパブリックキーを含むファイル（PEM形式）へのパス

データベースのドキュメントでのパラメータの詳細については、https://mariadb.com/kb/en/mysql_optionsv/を参照してください。

5. PostgreSQLデータベースについては、https://www.postgresql.org/docs/current/libpq-connect.html#LIBPQ-PARAMKEYWORDSも参照してください。

データソースセクション

設定ファイルには、一般的なセクションである`[LookupD]`の他、データソースへの接続を記述するセクション（各接続につき1つのセクション）もあります。これらのセクションには、`[LookupD.`*<type>*`.`*<name>*`]`というスキームを使用して名前が付けられます。ここで、

- *<type>* - 接続の種類：
  - `LDAP` - LDAPを使用するディレクトリサービス用。
  - `AD` - Active Directoryサービス。
  - `AllMatch` - *AllMatch*モードのテキストファイル用（フル認証）。
  - `Mask` - *Mask*モードのテキストファイル用（マスク認証）。
  - `Regex` - *Regex*モードのテキストファイル用（PCRE標準の正規表現に対する認証）。
  - `Cidr` - *Cidr*モードのテキストファイル用（IPアドレスまたはIPアドレス範囲認証）。
  - `Pq` - PostgreSQLデータベース用。
  - `Mysql` - Mysqlデータベース用。
  - `Sqlite` - Sqliteデータベース用。
  - `Redis` - Redisデータベース用。
- *<name>* - 接続の一意のID（タグ）。これを使用してルールから接続を参照できます。

たとえば、`[LookupD.LDAP.auth1]`とします。データソースのセクション内に含まれるパラメータのセットは、接続のタイプによって異なります。データソースセクションの数に制限はありません。

1. LDAPタイプのセクションで使用されるパラメータ

パラメータ	説明
`Url` *{string}*	使用されるLDAPサーバーと抽出されるデータを定義するURL。URLは、RFC 4516に従って次のスキームに基づいて構築されます。

<scheme>: / / *<host>*[: *<port>*]/ *<dn>*[? *<attrs>*[? *<scope>*[? *<filter>*[? *<extensions>*]]]]

各パラメータは次のとおりです。

<scheme> - サーバーへの接続方法（許可されるスキーム：`ldap`、`ldaps`、`ldapi`）。

<host>[:*<port>*] - リクエストを受信するLDAPサーバーアドレス。

<dn> - オブジェクトの識別名。このオブジェクトに関する情報が送信されました。

<attrs> - レコード属性の名前。この値はリクエスト内で受信する必要があります。

<scope> - 検索範囲（`base`、`one`、`sub`）。

<filter> - 抽出された属性の値に対するフィルター条件。

<extensions> - リクエストで使用されるLDAP拡張子のリスト。

機能

- 属性のリスト *<attrs>*では、「`*`」、「`+`」、「`1.1`」の任意の特殊文字を使用できます。
- 以下の自動的に解決されるプレースホルダーは、URLの *<dn>*および *<filter>*部分で使用できます。
  - `$u`は、クライアントコンポーネントによって送信されたユーザー名（`user`）に自動的に置き換えられます。
  - `$d`は、クライアントコンポーネントによって送信されたドメイン（`domain`）に自動的に置き換えられます。
  - `$D` - `dc=`*<subdomain>*,`dc=`*<domain>*に変更される *<subdomain>*.*<domain>*チェーン。
  - `$$` - 「$」文字。
- *<filter>*条件で特殊文字（「`*`」、「`(`」、「`)`」、「`\`」、コードが0の文字など）を通常の文字として使用する必要がある場合、それらの文字は`\XX`と記述する必要があります。さらに、URL LDAPの特殊文字は、シーケンス`%XX`を使用してエンコードされます。たとえば、「`/`」文字の`ldapi`スキームに従ってURLをローカルLDAPサーバーソケットへのパスの一部として使用する場合、この文字は`%2f`としてエンコードされます。
- *<extensions>*で許可される拡張機能として、`StartTLS`と`1.3.6.1.4.1.1466.20037`のみがサポートされます。これらの拡張機能では、保護されたスキームである`ldaps`を使用することが明示されていない場合でも、TLSメカニズム（つまり、LDAPサーバーとの保護された接続の確立）が使用されます。使用される拡張機能の名前の前に「`!`」文字がある場合は、TLSを使用する*必要があります*。つまり、安全な接続を確立できない場合には、リクエストは処理*されません*。それ以外の場合は、安全な接続が確立されていなくてもリクエストは処理されます。

指定された拡張機能は、保護された`ldaps`スキームでは使用できません。詳細については、RFC 4516または`man ldap_search_ext_s`を参照してください。

	例： `"ldaps://ds.example.com:990/$D?givenName,sn,cn?sub?(uid=$u)"` `"ldap://ldap.local/o=org,dc=nodomain?ipNetworkNumber?sub?(objectClass=ipNetwork)?!StartTLS"` デフォルト値：*（未設定）*
`BindDn` *{string}*	ユーザーが認証を受けなければならないLDAPディレクトリ内のオブジェクト。 例：`"cn=admin,dc=nodomain"` デフォルト値：*（未設定）*
`BindPassword` *{string}*	LDAPサーバー上での認証に使用するユーザーのパスワード。 デフォルト値：*（未設定）*
`ChaseReferrals` *{Boolean}*	現在のLDAPサーバーがリクエストへの応答として他のLDAPサーバーへの参照を返す場合、その参照に従うようコンポーネントに指示します。 デフォルト値：`No`

2. ADタイプのセクションで使用されるパラメータ

パラメータ	説明
`Host` *{string}*	接続するActive Directoryサービスのサーバーが稼働しているホストのドメイン名（FQDN）またはIPアドレス。 例：`"win2012.win.local"` デフォルト値：*（未設定）*
`Port` *{integer}*	Active Directoryサービスのサーバーがリッスンするホスト上のポート。 デフォルト値：`389`
`Dn` *{string}*	Active Directory内のオブジェクトの*DN*。これはLDAP URLの`dn`部分に似ています。 例：`"dc=win,dc=local"` デフォルト値：*（未設定）*
`User` *{string}*	識別に使用される、サーバー上のユーザーID。 例：`"Administrator@WIN.LOCAL"` デフォルト値：*（未設定）*
`Password` *{string}*	Active Directoryサーバー上での認証に使用されるユーザーのパスワード。 デフォルト値：*（未設定）*
`ChaseReferrals` *{Boolean}*	現在のActive Directoryサーバーがリクエストへの応答として他のLDAPサーバーへの参照を返す場合、その参照に従うようコンポーネントに指示します。 デフォルト値：`No`

UseSSL *{Boolean}*	Active Directoryサーバーへの接続にSSL/TLSを使用するよう指示します。 デフォルト値：No

3. AllMatch、Mask、Regex、Cidrタイプのセクションパラメータ

パラメータ	説明
File *{path}*	検索文字列を含むテキストファイルへのパス。 例："/etc/file1" デフォルト値：*（未設定）*

機能

- AllMatchタイプのセクションで指定したファイルの文字列は、大文字と小文字を区別しない完全に一致する文字列の検索に使用されます。
- Maskタイプのセクションで指定したファイルの文字列は、マスク（*ワイルドカード*）と見なされます。マスクは、標準文字と特殊文字を含む正規表現の簡略版と見なすことができます。文字列とマスクの照合は、大文字と小文字を区別しないで行われます。マスクには、次の特殊文字と式を含めることができます。

  * - 任意の文字シーケンス

  ? - 任意の1つの記号

  [*<character set>*] - セットの文字（たとえば、[bac]）

  [*<character set>*] - セットのどの記号にも一致しない文字（たとえば、[!cab]）

  [[:*<class>*:]] - 文字（*alnum*、*alpha*、*ascii*、*blank*、*cntrl*、*digit*、*graph*、*lower*、*print*、*punct*、*space*、*upper*、*xdigit*）のPOSIXクラスの文字

  サブストリングと一致するマスクには、「*」記号で囲まれたサブストリングが含まれている必要があります。（例：*host*）。いずれかの特殊文字を指定する必要がある場合は、バックスラッシュを使用して文字をエスケープする必要があります（\[、\]、*、\?）。必要に応じて、バックスラッシュをエスケープすることもできます（\\）。他の文字をエスケープしても意味はありません。たとえば文字列\a\b\c*\d\?\\は文字列abc*d?\に変換されます。マスクの例：

```
#Matches the “name” string exactly
name

#Matches the three-character strings where
#the first character is “c”, the second is any, and the third is “t”
#For example: “cat”, “cut”, “cct”
c?t

#Matches the strings: “user”, “users”, “us3rr”, “ussr1”, and so on
#(the [:alpha:] character class matches any alphabetical
#character, the special character “?” matches any character)
us[[:alpha:]34]r?

#Matches the strings: “.con”, “file.col”, “3...co!”, and so on
#(any character sequence before the “.co”, after—
#any character except “m” and “?”)
*.co[!m\?]

#Matches any string that contains “host”,
#For example: “host”, “localhost”, “hostel”, “ghosts”
*host*
```

- `Regex`タイプのセクションで指定したファイルの文字列は、PCRE（*Perl互換正規表現*）の通常の拡張機能として解釈されます。文字列と正規表現の照合は、大文字と小文字を区別しないで行われます。正規表現の例：

```
#IPv4
(\d{1,3}.){3}\d{1,3}

#Email address in the .com domain
\w+@\w+\.com
```

- *Cidr*タイプのセクションで指定したファイルの文字列は、IPアドレスまたはIPアドレス範囲として解釈されます。IPアドレスとIPアドレス範囲では、IPv4形式とIPv6形式が許可されます。サブネットマスクは、ビット（オクテット）形式とCIDR（*Classless Inter-Domain Routing*）表記で指定できます。例：

```
#IPv4
192.168.0.1
192.168.0.0/12
192.168.0.0/255.255.255.224

#IPv6
fe80::c7e8/32
fe80::c7e8/255.255.255.224
```

4. Pq、Mysql、Sqliteタイプのセクションで使用されるパラメータ

`Conn` *{string}*	データベース接続文字列。 使用可能な値： • `Mysql`（MySQL）、`Pq`（PostgreSQL）セクションの場合： `tcp://[<user>[:<password>]@]<host>[:<port>][/<database name>][?<parameter>=<value>[&...]]`

	unix://[<user>[: <password>]@]<path to socket>[: <database name>][?<parameter>=<value>[&...]] 例："tcp://user:pwd@localhost:1234/userdb"、"unix://user:pwd@/tmp/pgsql.sock:userdb" URI要件に注意してください。 • Sqlite（SQLite）セクションの場合： データベースファイルへのパス（file://スキームプレフィックスを指定）。 例："file:///home/user/users.db" デフォルト値：対応する*DefaultConnパラメータ値*によって定義*
Request *{string}*	データベースへのSQLクエリー文字列（SELECT）。ADおよびLDAPタイプのソースについては、以下の自動的に許可されるマーカーをクエリーで使用できます。 • $u、$Uは、クライアントコンポーネントによって送信されたユーザー名（user）に自動的に置き換えられます。 • $d、$Dは、クライアントコンポーネントによって送信されたドメイン（domain）に自動的に置き換えられます。 • $$は、「$」文字に置き換えられます。 例："SELECT username FROM users INNER JOIN domains ON users.domain = domains.id WHERE domains.name = $d AND users.name = $u" デフォルト値：*（未設定）*

SQLクエリーとして、SELECTタイプのクエリーのみ指定できます。置換を実行した後、クエリーは「そのまま」データベースに送信されます。クエリー結果に複数のカラムが含まれる場合、最初のカラムを除くすべてのカラムが無視されます。

5. Redisタイプのセクションで使用されるパラメータ

Conn *{string}*	Redisデータストレージとの接続文字列。 使用可能な値： • tcp://[*<password>*@]*<host>*[:*<port>*][/*<database index>*] • unix://[*<password>*@]*<socket path>*[:*<database index>*] URI要件に注意してください。 例："tcp://localhost:6379" デフォルト値：RedisDefaultConnパラメータ値によって定義
Request *{string}*	Redisストレージのクエリー文字列。クエリーでは、以下の自動的に許可されるマーカーを使用できます。 • $u、$Uは、クライアントコンポーネントによって送信されたユーザー名（user）に自動的に置き換えられます。 • $d、$Dは、クライアントコンポーネントによって送信されたドメイン（domain）に自動的に置き換えられます。

	• `$$`は、「`$`」文字に置き換えられます。 例：`"HVALS bad_users"` デフォルト値：*（未設定）*

クエリー結果に複数のカラムが含まれる場合、最初のカラムを除くすべてのカラムが無視されます。

新しいデータソース用のセクションの追加

Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctl（`drweb-ctl`コマンドでアクセス）を使用し、*<name>*タグを使って、サポートされるタイプの新しいデータソース用に新しいセクションを追加するには、次のコマンドを使用する必要があります。

```
# drweb-ctl cfset LookupD.<type> -a <name>
```

例：

```
# drweb-ctl cfset LookupD.AD -a WinAD1
# drweb-ctl cfset LookupD.AD.WinAD1.Host 192.168.0.20
```

最初のコマンドは`[LookupD.AD.WinAD1]`という名前のセクションを設定ファイルに追加し、2番目のコマンドはこのセクション内の`Host`パラメータの値を変更します。

あるいは次の例のように、新しいセクションを設定ファイルの末尾に直接書き込むこともできます。

```
[LookupD.AD.WinAD1]
Host = 192.168.0.20
```

どちらの方法でも効果は同じです。ただし設定ファイルを編集する場合は、`drweb-configd`コンポーネントに`SIGHUP`信号を送信して、変更した設定を適用する必要があります。これを行うには、次のコマンドを実行します。

```
# drweb-ctl reload
```

Dr.Web StatD

Dr.Web StatDコンポーネントは、Dr.Web for UNIX Mail Serversのコンポーネントの動作中に発生するイベントの統計を蓄積するために設計されています。イベントは永続的なリポジトリに保存され、リクエストに応じて取得できます。

動作原理

このコンポーネントは、Dr.Web for UNIX Mail Serversコンポーネントの操作中に取得されたイベントの蓄積と無期限の保存を確保します。次のタイプのイベントがログに記録されます。

- コンポーネントの緊急シャットダウン
- 脅威の検出（特にメールメッセージ）

Dr.Web StatDはデーモンモードで動作し、設定制御デーモンによって自動的に起動されます。イベントの表示と管理はDr.Web Ctlユーティリティの`events`コマンドによって確認できます。

コマンドライン引数

Dr.Web StatDを実行するには、コマンドラインに次のコマンドを入力します。

```
$ <opt_dir>/bin/drweb-statd [<parameters>]
```

Dr.Web StatDは次のオプションを処理できます。

パラメータ	説明
`--help`	**機能**：コマンドラインパラメータに関する簡単なヘルプ情報をコンソールまたはターミナルエミュレーターに出力し、完了時に終了します。 **短縮形**：`-h` **引数**：なし
`--version`	**機能**：このコンポーネントのバージョンに関する情報をコンソールまたはターミナルエミュレーターに出力し、完了後に終了します。 **短縮形**：`-v` **引数**：なし

例：

```
$ /opt/drweb.com/bin/drweb-statd --help
```

このコマンドは、Dr.Web StatDに関する簡単なヘルプ情報を出力します。

スタートアップノート

このコンポーネントは、自律モードで（他のコンポーネントから自律的に）OSのコマンドラインから直接起動することはできません。必要に応じてDr.Web ConfigD設定デーモンによって自動的に起動されます。コンポーネントの

動作を管理するには、Dr.Web for UNIX Mail Servers用のコマンドラインベース管理ツールDr.Web Ctlを使用できます（これは`drweb-ctl`コマンドを使用して呼び出されます）。

> コマンドラインから製品のこのコンポーネントに関するドキュメントを要求するには、`man 1 drweb-statd`コマンドを使用します。

設定パラメータ

このコンポーネントは、Dr.Web for UNIX Mail Serversの統合された設定ファイルの`[StatD]`セクションで指定されている設定パラメータを使用します。

セクションには以下のパラメータが含まれています。

パラメータ	説明
`LogLevel` *{logging level}*	コンポーネントのロギングレベル。 パラメータの値が指定されていない場合は、`[Root]`セクションの`DefaultLogLevel`パラメータの値が使用されます。 デフォルト値：`Notice`
`Log` *{log type}*	コンポーネントのロギング方式。 デフォルト値：`Auto`
`ExePath` *{path to file}*	コンポーネントのファイルへの実行パス。 デフォルト値：*<opt_dir>*`/bin/drweb-statd` • GNU/Linuxの場合：`/opt/drweb.com/bin/drweb-statd` • FreeBSDの場合：`/usr/local/libexec/drweb.com/bin/drweb-statd`
`RunAsUser` *{UID / user name}*	その権限によりコンポーネントを実行するユーザーの名前。このユーザー名は、ユーザーのUIDまたはユーザーのログインとして指定できます。ユーザー名が数字で構成されている場合（つまりUIDに似ている場合）は、「`name:`」というプレフィックスを付けて指定します。たとえば、`RunAsUser = name:123456`です。 ユーザー名が指定されていない場合、コンポーネント操作は開始後にエラーが発生して終了します。 デフォルト値：`drweb`
`IdleTimeLimit` *{time interval}*	コンポーネントの最大アイドル時間。指定された時間が経過すると、コンポーネントはシャットダウンします。 指定可能な値：10秒（`10s`）から30日（`30d`）まで。 `None`値が設定されている場合、コンポーネントは永続的に機能します。コンポーネントがアイドル状態になると、`SIGTERM`シグナルは送信されません。 デフォルト値：`30s`

パラメータ	説明
`MaxEventStoreSize` *{size}*	イベントデータベースの最大許容サイズ。`mb` で定義されます。例：`MaxEventStoreSize = 100mb`。 最小値：`50mb` デフォルト値：`1GB`

付録

付録A. コンピューター脅威の種類

本マニュアルでは、コンピューターやネットワークに対して潜在的または直接的な損害を与え、ユーザーの情報や権限を侵害するあらゆる種類のソフトウェアを*「脅威」*と定義します（悪意のあるソフトウェアやその他の不要なソフトウェア）。広義では、コンピューターまたはネットワークのセキュリティに対するあらゆる種類の潜在的な危険（すなわちハッカー攻撃につながる脆弱性）を指して「脅威」とする場合があります。

以下に記載するすべての種類のプログラムは、ユーザーのデータまたは機密情報を危険にさらすものです。姿を隠さないプログラム（スパム配信ソフトウェアやさまざまなトラフィックアナライザなど）は、状況によっては脅威と化す可能性はありますが、通常はコンピューター脅威と見なされません。

コンピューターウイルス

この種類のコンピューター脅威は、他のプログラム内にそのコードを埋め込む（これを感染と呼びます）ことができるという特徴を持っています。多くの場合、感染したファイルはそれ自体がウイルスのキャリアとなり、また埋め込まれたコードは必ずしもオリジナルのものと一致するとは限りません。ほとんどのウイルスは、システム内のデータを破損させる、または破壊する目的を持っています。

Doctor Webの分類では、ウイルスは感染させるオブジェクトの種類に応じて分けられます。

- *ファイルウイルス*は、OSのファイル（通常、実行ファイルおよびダイナミックライブラリ）を感染させ、そのファイルの起動と同時にアクティブになります。
- *マクロウイルス*は、Microsoft® Officeやマクロコマンドをサポートする他のアプリケーション（Visual Basicで書かれたものなど）が使用するドキュメントに感染するウイルスです。*マクロコマンド*は、完全に機能するプログラミング言語で書かれた一種の実装プログラム（マクロ）です。たとえば、Microsoft® Wordでは、文書を開くか、閉じるか、保存すると、マクロが自動的に開始されることがあります。
- *スクリプトウイルス*はスクリプト言語を使用して作成され、他のスクリプト（OSのサービスファイルなど）に感染します。これらはスクリプトを実行できる他のファイル形式に感染することができるため、Webアプリケーションのスクリプトの脆弱性を利用します。
- *ブートウイルス*は、ディスクやパーティションのブートレコード、またはハードドライブのマスターブートレコードに感染します。多くのメモリを必要とせず、システムのロールアウト、再起動、またはシャットダウンが実行されるまでタスクを実行し続けられます。

多くのウイルスは検出に対抗する何らかの手段を持ち、その手法は常時改良され続けていますが、それを打開する方法も常に開発されています。すべてのウイルスは、その使用する手法に応じて分類できます。

- *暗号化ウイルス*は、感染するたびにコードを暗号化して、ファイル、ブートセクター、またはメモリ内での検出を妨げます。このようなウイルスのすべてのコピーには、ウイルスのシグネチャとして使用される可能性がある小さな共通コードの一部（復号手順）しか含まれていません。
- *ポリモーフィック型ウイルス*も同様に自身のコードを暗号化しますが、ウイルスのコピーごとに異なる特別な復号プロシージャの生成も行います。つまり、この種のウイルスにはシグネチャバイトがありません。
- *ステルスウイルス*は、特定のアクションを実行して、感染したオブジェクトでの活動と存在を隠します。このようなウイルスは、オブジェクトを感染させる前にそのオブジェクトの特性を収集し、スキャナが変更されたファイルを探し出す際に誤認させるための「ダミー」特性を作り出します。

ウイルスは、書かれているプログラミング言語（ほとんどの場合アセンブラ、高級プログラミング言語、スクリプト言語など）、または感染させるOSに応じて分類することもできます。

コンピューターワーム

「コンピューターワーム」型の悪意のあるプログラムは、ウイルスやその他のマルウェアよりも多く見られるようになってきています。ウイルス同様、自身を複製し拡散できますが、他のオブジェクトを感染させることはありません。ネットワークを通じて（通常、メールの添付ファイルとして、またはインターネットから）侵入し、ネットワーク内にある他のコンピューターにコピーを拡散します。ユーザーのアクションに応じて、または攻撃するコンピューターを選択する自動モードで拡散を開始します。

ワームは1つのファイル（ワームの本体）のみで構成されているとは限りません。多くのワームが、メインメモリ（RAM）内に読み込んだ後にワームの本体を実行ファイルとしてネットワーク経由でダウンロードする感染部分（シェルコード）を持っています。シェルコードがシステム内に存在するだけであれば、システムを再起動することで（RAMが削除されリセットされます）ワームを削除できますが、ワームの本体がコンピューターに侵入してしまった場合はアンチウイルスプログラムのみが対処可能です。

ワームはその驚異的な拡散速度によって、ペイロードを持っていない（直接的な被害を与えない）場合であっても、ネットワーク全体の機能を破壊する能力を持っています。

Doctor Webの分類では、ワームはその拡散方法によって以下のように分けられます。

- *ネットワークワーム*は、さまざまなネットワークとファイル共有プロトコル経由で自身のコピーを拡散します。
- *メールワーム*は、メールプロトコル（POP3、SMTPなど）を使用して拡散します。
- *チャットワーム*は、広く使用されているメッセンジャーやチャットプログラム（ICQ、IM、IRCなど）のプロトコルを使用します。

トロイの木馬プログラム（トロイの木馬）

この種類のコンピューター脅威は自身を複製しません。トロイの木馬は頻繁に使用されるプログラムに成り代わり、その機能を実行します（または動作を模倣します）。同時に、システム内で悪意のあるアクション（データの破損または破壊、機密情報の送信など）を実行したり、ハッカーが許可なしにコンピューターにアクセス（たとえば第三者のコンピューターに損害を与えるために）したりすることを可能にします。

トロイの木馬の悪意のある機能は、ウイルスのそれに似ています。トロイの木馬は、ウイルスのコンポーネントである場合もあります。しかし、ほとんどのトロイの木馬は、ユーザーまたはシステムタスクによって起動される個別の実行ファイルとして配布されます（ファイル交換サーバー、リムーバブルストレージ、メール添付ファイルなどを介して）。

トロイの木馬は、よくウイルスやワームによって拡散されることや、他の種類の脅威によっても実行されうる、悪意のあるアクションの多くがトロイの木馬にも起因することから、その分類が難しくなっています。以下のトロイの木馬は、Doctor Webでは個別のクラスとして分類されています。

- *バックドア*は、侵入者がシステムにログオンしたり、既存のアクセスやセキュリティ対策を回避する特権機能を取得したりすることを可能にするトロイの木馬です。バックドアはファイルに感染しませんが、レジストリキーを変更して自身をレジストリに書き込みます。
- *ルートキット*は、自身を隠すためにOSのシステム機能を監視する際に使用されます。さらに、ルートキットは、他のプログラム（他の脅威）、レジストリキー、フォルダ、ファイルのプロセスを隠すことができます。ルートキットは独立したプログラムとして、または別の悪意のあるプログラムのコンポーネントとして拡散されます。動作モードに応じて2種類のルートキットがあります。ユーザーモードで動作（ユーザーモードライブラリの機能を監視）する

*ユーザーモードのルートキット（UMR）*とカーネルモードで動作する（システムのカーネルレベルで機能を監視して、悪意のあるプログラムを検出しにくくする）*カーネルモードのルートキット（KMR）*。

- *キーロガー*は、ユーザーがキーボードを使って入力したデータを記録するために使用されます。目的は、個人情報（ネットワークパスワード、ログイン、クレジットカードデータなど）を盗むことです。
- *クリッカー*は、Webサイトのトラフィックを増加させる目的で、またはDDoS攻撃を実行するためにハイパーリンクを別の（ときに有害な）アドレスにリダイレクトします。
- *プロキシ型トロイの木馬*は被害者のコンピューターを介して匿名でインターネットにアクセスします。

トロイの木馬は、Webブラウザのスタートページを変更したり特定のファイルを削除したりするなど、上記以外の悪意のあるアクションも実行することがあります。ただしそのような動作は、他の種類の脅威（ウイルスやワーム）によって実行される場合もあります。

ハッキングツール

ハッキングツールは、侵入者によるハッキングを可能にするプログラムです。最も一般的なものは、ファイアーウォールまたはその他のコンピューター保護システムコンポーネントの脆弱性を検出するポートスキャナです。それらのツールはハッカーだけではなく、管理者がネットワークのセキュリティを検査するためにも用いられます。ハッキングに使用することのできる一般的なソフトウェアや、ソーシャルエンジニアリングテクニックを使用するさまざまなプログラムもハッキングツールに含まれることがあります。

アドウェア

通常、ユーザーの画面に強制的に広告を表示させるフリーウェアプログラム内に組み込まれたプログラムコードを指します。ただしそのようなコードは、他の悪意のあるプログラム経由で配布されてWebブラウザ上に広告を表示させる場合もあります。アドウェアプログラムの多くは、スパイウェアによって収集されたデータを用いています。

ジョークプログラム

アドウェア同様、この種類の軽微な脅威はシステムに対して直接的な被害を与えることはありません。ジョークプログラムは通常、実際には起こっていないエラーに関するメッセージを表示させ、データの損失につながるアクションの実行を要求します。その目的はユーザーを驚かせたり不快感を与えたりすることにあります。

ダイアラー

幅広く電話番号をスキャンし、モデムとして応答するものを見つけるための特別なコンピュータープログラムです。その後、攻撃者がその番号を使用することによって被害者に通話料の請求書が送られます。または被害者が気づかぬうちに、モデム経由で高額な電話サービスに接続されます。

リスクウェア

これらのソフトウェアアプリケーションは悪意のある目的のために作成されたものではありませんが、コンピューターセキュリティに対する脅威となりうる特徴を持っています。リスクウェアプログラムはデータを破損または削除してしまう可能性があるのみならず、クラッカー（悪意のあるハッカー）や悪意のあるプログラムによって、システムに被害を与える目的で使用されることがあります。そのようなプログラムの中には、さまざまなリモートチャットおよび管理ツール、FTPサーバーなどがあります。

疑わしいオブジェクト

ヒューリスティックアナライザによって検出される潜在的なコンピューター脅威です。これらのオブジェクトは、あらゆるタイプの脅威（ITセキュリティの専門家も把握していないもの）である可能性があり、または誤検出の場合もあります。疑わしいオブジェクトを含むファイルを隔離に移動するよう選択することをお勧めします。解析のためにDoctor Webアンチウイルスラボにも送信する必要があります。

付録B. コンピューター脅威の駆除

この付録の内容

- 検出方法
- 脅威に関連したアクション

Doctor Webアンチウイルスソリューションは、悪意のあるソフトウェア検出に複数の手法を同時に使用します。それにより、感染が疑われるファイルに対する徹底的なスキャンを実行し、ソフトウェアの動作を管理できます。

検出方法

シグネチャ解析

スキャンはまず、ファイルコードセグメントを既知のウイルス署名と比較するシグネチャ解析で始まります。シグネチャはウイルスを特定するために必要かつ十分な、連続するバイトの有限なシーケンスです。シグネチャ辞書のサイズを抑えるため、Dr.Webアンチウイルスソリューションはシグネチャのシーケンス全体ではなくチェックサムを使用します。チェックサムはシグネチャを特定し、ウイルス検出および駆除の正確さを維持します。Dr.Webウイルスデータベースは、いくつかのエントリによって、特定のウイルスのみでなく脅威のクラス全体を検出できるよう設計されています。

Origins Tracing™

シグネチャ解析の完了後、Dr.Webアンチウイルスソリューションは既知の感染メカニズムを用いる新種・亜種ウイルスを検出するため、ユニークなテクノロジーOrigins Tracing™を使用します。それにより、Dr.Webユーザーはランサムウェアであるtrojan.Encoder.18（別名gpcode）のような悪質な脅威から保護されます。新種・亜種ウイルスの検出を可能にする他、Origins Tracing™はDr.Webヒューリスティックアナライザによる誤検出を劇的に減らします。Origins Tracing™アルゴリズムを使用して検出されたオブジェクトの名前には .Origin拡張子が付きます。

実行のエミュレーション

プログラムコードエミュレーションの技術は、チェックサムによる検索が直接適用できない場合、または実行するのが非常に困難な場合（安全な署名を構築することが不可能なため）に、ポリモーフィック型ウイルスと暗号化ウイルスの検出に使用されます。この方法は、*エミュレーター*、つまりプロセッサーとランタイム環境のプログラミングモデルによる解析コード実行のシミュレーションを意味します。エミュレーターは保護されたメモリ領域（*エミュレーションバッファ*）で動作し、解析されたプログラムの実行は命令ごとにモデル化されます。ただし、これらの命令は実際にはCPUによって実行されるものではありません。エミュレーターがポリモーフィック型ウイルスに感染したファイルを受信すると、エミュレーションの結果は復号されたウイルスコードになります。これは、シグネチャチェックサムを検索することで簡単に判別できます。

ヒューリスティック解析

ヒューリスティックアナライザの検出手法は、ウイルスコードに典型的な、または非常にまれな特徴（属性）に関する特定の情報に基づいています（*ヒューリスティック*）。各属性は、その深刻度および信頼度を定義する*重み*係数を持っています。属性が悪意のあるコードであることを示している場合には重み係数がプラスになり、コンピュー

ター脅威の特徴を示していない場合はマイナスになります。ヒューリスティックアナライザはファイルの重み付け合計値に応じて、未知のウイルスに感染している可能性を計算します。それらの合計が一定のしきい値を超えている場合、ヒューリスティックアナライザによって、オブジェクトは未知のウイルスに感染している可能性があると判定されます。

ヒューリスティックアナライザはファイル解凍の柔軟なアルゴリズムであるFLY-CODE™テクノロジーも使用します。このテクノロジーは、Dr.Webにとって既知のパッカーのみでなく、これまでに発見されていない未知のパッカーによって圧縮されたファイル内に悪意のあるオブジェクトが存在する可能性をヒューリスティックに検出します。Dr.Webアンチウイルスソリューションは圧縮されたオブジェクトのスキャン中に構造エントロピー解析も使用します。このテクノロジーはコードの配置を解析することで脅威を検出します。そのため、1つのウイルスデータベースから、同じポリモーフィックパッカーによって圧縮された他の多くの脅威を検出することが可能になります。

不確実な状況で仮説を扱うあらゆるシステム同様、ヒューリスティックアナライザもまたタイプIまたはタイプIIのエラーを生じさせる可能性があります（ウイルスを見逃す、または誤検知）。そのため、ヒューリスティックアナライザによって検出されたオブジェクトは「疑わしい」オブジェクトとして定義されます。

上記のスキャン手法に加え、Dr.Webアンチウイルスソリューションは既知の悪意のあるソフトウェアに関する最も新しい情報も使用します。Doctor Webアンチウイルスラボのエキスパートによって新しい脅威が発見されると、そのウイルスシグネチャ、振る舞い特性、属性を追加した更新が即座に配信されます。更新は1時間に数回行われる場合もあり、たとえ新種の悪意のあるプログラムがDr.Web常駐保護を通過してシステムに侵入した場合でも、更新後には検出され駆除されます。

クラウドベースの脅威検出テクノロジー

クラウドベースの検出方法では、あらゆるオブジェクト（ファイル、アプリケーション、ブラウザ拡張機能など）をハッシュ値によってスキャンします。ハッシュは、特定の長さの数字と文字からなる一意のシーケンスです。ハッシュ値による分析では、オブジェクトは既存のデータベースを使用してスキャンされ、カテゴリー別に分類されます（クリーン、疑わしい、悪意のある、など）。

このテクノロジーにより、ファイルスキャンの時間を最適化し、デバイスリソースを節約することができます。分析されるのはオブジェクトではなく、その固有のハッシュ値であるため、オブジェクトが悪意のあるものであるかどうかの決定はほとんど瞬時に行われます。Dr.Web Cloudサーバーに接続されていない場合、ファイルはローカルでスキャンされ、接続が復元されるとクラウドスキャンが再開されます。

Dr.Web Cloudサービスは多くのユーザーから情報を収集し、これまで未知であった脅威に関するデータを迅速に更新します。これにより、デバイス保護の効果を高めます。

アクション

コンピューター脅威を回避するために、Dr.Web製品は悪意のあるオブジェクトに対してさまざまなアクションを適用します。ユーザーはデフォルト設定を使用したり、自動的に適用するアクションを設定したり、あるいは検出のたびに手動でアクションを選択したりできます。使用可能なアクションは以下のとおりです。

- **Ignore** *(無視)* - いずれのアクションも実行せず、検出された脅威をスキップするように指示します。
- **Report** *(報告)* - 他のすべてのアクションを実行せず、検出された脅威について通知するように指示します。
- **Cure** *(修復)* - 感染したオブジェクトから悪意のあるコンテンツのみを削除し、修復するように指示します。ただし、すべての種類の脅威に対して適用できるわけではありません。

- **Quarantine** *(隔離)* - 検出された脅威を特別なディレクトリに移し、残りのシステムから隔離するように指示します。
- **Delete** *(削除)* - 感染したオブジェクトを永久に削除するように指示します。

> コンテナ（アーカイブ、メールメッセージなど）内のファイルで脅威が検出された場合は、削除アクションの代わりにコンテナの隔離への移動が実行されます。

Dr.Web MailDがメールメッセージをスキャンすると、次のアクションがメールメッセージに適用されます。

- **Pass** *(スキップ)* - いずれのアクションも実行せず、検出された脅威をスキップするように指示します。
- **Reject** *(拒否)* - メールメッセージを拒否し、受信者への配信を禁止するように指示します。
- **Tempfail** *(エラーの返信)* - メールメッセージの配信の代わりに、送信者または受信者にエラーメッセージを返すように指示します。
- **Discard** *(破棄)* - メールメッセージを受け入れ、受信者には配信しません。
- **Repack** *(リパック)* - メールメッセージを受信者に配信する前に、脅威を隔離に移動することによってメールメッセージを修正するように指示します。メールメッセージにアーカイブを添付し、脅威検出に関する通知を追加します。
- **Add Header** *(ヘッダーの追加)* - 受信者への配信時にメールメッセージにヘッダーを追加します。
- **Change Header** *(ヘッダーの変更)* - 受信者への配信中に、指示されたヘッダーの値を変更します。

付録C. テクニカルサポート

Dr.Web製品のインストールまたは使用中に問題が発生した場合、テクニカルサポートへのお問い合わせの前に以下のオプションをご利用ください:

- https://download.drweb.com/doc/ から最新のマニュアルやガイドをダウンロードして読む。
- https://support.drweb.com/show_faq/ で「よくあるご質問」を読む。
- https://forum.drweb.com/ でDr.Webフォーラムを見る。

問題が解決しなかった場合、サポートサイト https://support.drweb.com/ の該当するセクション内でwebフォームに必要事項を入力し、直接 Doctor Web テクニカルサポートまでお問い合わせください。

企業情報については、Doctor Web 公式サイト https://company.drweb.com/contacts/offices/ をご覧ください。

問題に対する円滑な対応を可能にするため、テクニカルサポートにご連絡いただく前に、インストールされた製品とその設定、およびシステム環境に関するデータセットを生成することをお勧めします。これは、Dr.Web for UNIX Mail Serversディストリビューションに含まれている特別なユーティリティを使用して行うことができます。

テクニカルサポートに提出するデータを収集するには、次のコマンドを使用します。

```
# <opt_dir>/bin/support-report.sh
```

*<opt_dir>*は、実行可能ファイルとライブラリを含むDr.Web for UNIX Mail Serversファイルのディレクトリです（GNU/Linuxの場合はデフォルトで`/opt/drweb.com`）。ディレクトリに使用される表記規則については、はじめにを参照してください。

テクニカルサポートに提出するデータを収集する際は、ユーティリティをスーパーユーザー権限（*root*ユーザーの権限）で起動することをお勧めします。権限を昇格するには、`su`コマンドで別のユーザーとしてログインするか、`sudo`コマンドで別のユーザーの権限でコマンドを実行します。

ユーティリティは次の情報を収集してアーカイブします。

- OSに関するデータ（名前、アーキテクチャ、`uname -a`コマンドの結果）
- Doctor Webパッケージを含む、システムにインストールされているパッケージのリスト
- ログの内容：
  - Dr.Web for UNIX Mail Serversのログ（コンポーネントごとに設定されている場合）
  - `syslog`システムデーモンのログ（`/var/log/syslog`, `/var/log/messages`）
  - システムパッケージマネージャーのログ（`apt`、`yum`など）
  - `dmesg`のログ
- 次のコマンドの出力：`df`、`ip a`（`ifconfig -a`）、`ldconfig -p`、`iptables-save`、`nft export xml`
- Dr.Web for UNIX Mail Serversの設定と構成に関する情報：
  - ダウンロードされたウイルスデータベースのリスト（`drweb-ctl baseinfo -l`）
  - Dr.Web for UNIX Mail Serversディレクトリにあるファイルのリストとそれらファイルの MD5ハッシュ値
  - Dr.Web Virus-Finding Engineスキャンエンジンのバージョンと MD5ハッシュ値
  - Dr.Web for UNIX Mail Serversの設定パラメータ（`drweb.ini`の内容、ルール、ルールで使用される値のファイル、Luaプロシージャなどを含む）
  - Dr.Web for UNIX Mail Serversがスタンドアロンモードで動作している場合、キーファイルから取得したユーザーの情報と権限

製品とそのシステム環境に関する情報を含むアーカイブは、ユーティリティを起動したユーザーのホームディレクトリに保存されます。ファイルの名前は次のようになります。

```
drweb.report.<timestamp>.tgz
```

*<timestamp>*は、レポート作成時の完全なタイムスタンプ（ミリ秒単位）です（例：`20190618151718.23625`）。

付録D. Dr.Web for UNIX Mail Servers設定ファイル

すべてのDr.Web for UNIX Mail Serversコンポーネントの設定パラメータは、特別な調整デーモンDr.Web ConfigDによって管理されます。これらのパラメータは`drweb.ini`ファイルに格納されています。デフォルトディレクトリは*<etc_dir>*（GNU/Linuxの場合は`/etc/opt/drweb.com`）です。

> テキスト設定ファイルには、デフォルト値とは値の異なるパラメータのみが格納されています。パラメータが設定ファイルにない場合は、そのデフォルト値が使用されます。
>
> *<opt_dir>*、*<etc_dir>*、*<var_dir>*の表記規則の詳細は、はじめにを参照してください。

設定ファイルに存在せず、かつデフォルト値を持つパラメータなど、利用可能なすべてのパラメータのリストは、次のコマンドを使用して表示できます。

```
$ drweb-ctl cfshow
```

パラメータの値は次のいずれかの方法で変更できます。

1. 設定ファイルでパラメータを指定（任意のテキストエディターでファイルを編集）し、変更を適用するためのSIGHUP信号を設定デーモン（`drweb-configd`コンポーネント）に送信します（これを行うには、以下のコマンドを実行します）。

   ```
   # drweb-ctl reload
   ```

2. このコマンドをコマンドラインに入力します。

   ```
   # drweb-ctl cfset <section>.<parameter> <new value>
   ```

   > このコマンドは、管理ツールDr.Web Ctlがスーパーユーザー権限で実行されている場合にのみ実行できます。スーパーユーザー権限を取得するには、`su`または`sudo`コマンドを使用します。

コマンドライン管理ツールDr.Web Ctl（`drweb-ctl`モジュール）の`cfshow`および`cfset`のコマンド構文の詳細については、Dr.Web Ctlのセクションを参照してください。

ファイル構造

設定ファイルの構造は以下のとおりです。

- ファイルの内容は、名前が付けられたセクションに分割されます。こうしたセクションで利用可能な名前は厳密に事前定義されており、変更できません。セクション名は角括弧で指定され、セクションパラメータを使用するDr.Web for UNIX Mail Serversコンポーネント名と似ています（設定デーモンDr.Web ConfigDのすべてのパラメータを保存する[Root]セクションは除く）。
- 設定ファイル内の「;」または「#」文字はコメントの始まりを示します。これらの文字に続くすべてのテキストは、設定パラメータの読み取り中にDr.Web for UNIX Mail Serversコンポーネントによってスキップされます。
- ファイル内の各行には、次のようにパラメータ値を1つだけ含めることができます。

```
<Parameter name> = <Value>
```

- すべてのパラメータ名は厳密に事前定義されており、変更できません。
- セクション名とパラメータ名はすべて大文字と小文字が区別されません。パラメータ値も、パスに含まれるディレクトリやファイルの名前を除き（UNIXのようなOSの場合）、大文字と小文字が区別されません。
- ファイル内のセクションの順序や、セクション内のパラメータの順序は重要ではありません。
- 設定ファイルのパラメータ値は引用符で囲むことができ、空白がある場合は引用符で囲む必要があります。
- 一部パラメータは複数の値を取ることができます。その場合、値はコンマで区切るか、設定ファイルの異なる行に複数指定します。前者の場合、コンマの前後の空白は無視されます。空白文字がパラメータ値の一部である場合は、その文字を引用符で囲む必要があります。

  次のようにして複数の値を指定できます。

  1) コンマ区切りのリストにする。

     ```
     Parameter = Value1, Value2,"Value 3"
     ```

  2) 設定ファイルで行を複数指定する。

     ```
     Parameter = Value2
     Parameter = Value1
     Parameter = "Value 3"
     ```

  値の順序は任意です。

  

  ファイルやディレクトリへのパスをコンマで区切る場合、それらを引用符で囲む必要があります。

  ```
  ExcludedPaths = "/etc/file1", "/etc/file2"
  ```

  一連のパスを複数行として表す場合、引用符は必要ありません。

  ```
  ExcludedPaths = /etc/file1
  ExcludedPaths = /etc/file2
  ```

- パラメータが複数の値を取り得る場合は、設定ファイルのコメントまたは本マニュアルの本文にその旨が記載されています。

設定ファイルのセクションの説明については、Dr.Web for UNIX Mail Serversコンポーネントの説明を参照してください。

パラメータタイプ

設定パラメータには以下のタイプがあります。

- *address* - *<IP address>:<port>*として指定されたネットワーク接続アドレス。
- *boolean* - パラメータの可能な値は、`Yes`または`No`の2つだけです。
- *integer* - 非負の整数。
- *fractional number* - 小数部分のある非負の整数。

- *time interval* - 非負の整数と時間単位を示すサフィックス（文字）で構成される時間間隔。以下のサフィックスを使用できます。
  - `w` - 週（`1w = 7d`）
  - `d` - 日（`1d = 24h`）
  - `h` - 時間（`1h = 60m`）
  - `m` - 分（`1m = 60s`）
  - `s`またはサフィックスなし - 秒

  時間間隔を秒で指定した場合、小数点の後にミリ秒を指定できます（区切り文字の後に3桁以内、0．5秒～500ミリ秒など）。異なる時間単位で複数の時間間隔を指定できます。この場合、間隔の合計が計算されます（実際には、値が設定に書き込まれる前に時間間隔は常にミリ秒に変換されます）。

  基本的には、すべての時間間隔はN_1`w`N_2`d`N_3`h`N_4`m`N_5[`.`N_6]`s`の形式で表現されます。ここでN_1, ..., N_6は、この間隔に含まれる時間単位の数です。たとえば、1年（365日）は`365d`、`52w1d`、`52w24h`、`51w7d24h`、`51w7d23h60m`、`8760h`、`525600m`、`31536000s`と表すことができます（すべてのレコードは同じ365日を表しています）。

  以下の例では、30分、2秒、500ミリ秒の間隔を指定する方法を示しています。

  1) 設定ファイルで指定する。

  ```
  UpdateInterval = 30m2.5s
  ```

  2) `drweb-ctl cfset`コマンドを使用する。

  ```
  # drweb-ctl cfset Update.UpdateInterval 1802.5s
  ```

  3) コマンドラインパラメータを使用する（例：コマンドライン引数の場合）。

  ```
  $ drweb-se --WatchdogInterval 1802.5
  ```

- *size* - パラメータ値はオブジェクト（ファイル、バッファ、キャッシュなど）のサイズで表すことができ、非負の整数と、単位を表すサフィックスで構成します。以下のサフィックスを使用できます。
  - `mb` - メガバイト（`1mb = 1024kb`）
  - `kb` - キロバイト（`1kb = 1024b`）
  - `b` - バイト

  サフィックスを省略した場合、サイズはバイト単位と見なされます。異なる単位で複数のサイズを指定できます。この場合、サイズの合計が計算されます（実際には、サイズ値は常にバイトに変換されます）。
- *path to a directory (file)* - パラメータ値は、ディレクトリ（ファイル）へのパスである文字列になります。

ファイルパスはファイル名で終わる必要があります。

UNIX系システムでは、ディレクトリとファイルの名前は大文字と小文字が区別されます。パラメータ記述で明示的に指定されていない場合、パスに特殊文字（`?`、`*`）のあるマスクを含めることはできません。

- *logging level* - Dr.Web for UNIX Mail Serversコンポーネントのイベントを記録するレベル。以下の値を使用できます。
  - `DEBUG` - 最も詳細なロギングレベル。すべてのメッセージとデバッグ情報が登録されます。
  - `INFO` - すべてのメッセージが登録されます。
  - `NOTICE` - すべてのエラーメッセージ、警告、通知が登録されます。
  - `WARNING` - すべてのエラーメッセージと警告が登録されます。
  - `ERROR` - エラーメッセージのみが登録されます。
- *log type* - Dr.Web for UNIX Mail Serversコンポーネントによるログの実行方法（ロギング方式）をパラメータ値で定義します。以下の値を使用できます。
  - `Stderr[:ShowTimestamp]` - メッセージは*stderr*（標準エラーストリーム）に表示されます。この値は設定デーモンの設定で*のみ*使用できます。バックグラウンドモードで動作する（「*デーモン化される*」）場合、つまりパラメータ`-d`を指定して起動した場合、バックグラウンドモードで動作するコンポーネントは端末のI/Oストリームにアクセスできないため、この値は*使用できません*。追加パラメータ`ShowTimestamp`は、すべてのメッセージにタイムスタンプを追加するように指示します。
  - `Auto` - ロギング対象のメッセージは設定デーモンDr.Web ConfigDに送られ、設定に基づいて一か所（`[Root]`セクションの`Log`パラメータ）に保存されます。この値は、*設定デーモンを除く*すべてのコンポーネントに指定され、デフォルト値として使用されます。
  - `Syslog[:<facility>]` - メッセージはシステムロギングサービス`syslog`に送信されます。
  - 追加オプション*<facility>*は、`syslog`のメッセージ登録レベルを指定するために使用します。次の値を使用できます。
    - `DAEMON` - デーモンのメッセージ
    - `USER` - ユーザープロセスのメッセージ
    - `MAIL` - メールプログラムのメッセージ
    - `LOCAL0` - ローカルプロセス0のメッセージ

    ...
    - `LOCAL7` - ローカルプロセス7のメッセージ
  - *<path>* - メッセージは指定されたログに直接保存されます。

  パラメータ値の指定方法の例：

  1) 設定ファイルで指定する。

  ```
  Log = Stderr:ShowTimestamp
  ```

  2) `drweb-ctl cfset`コマンドを使用する。

  ```
  # drweb-ctl cfset Root.Log /var/opt/drweb.com/log/general.log
  ```

  3) コマンドラインパラメータを使用する（例：コマンドライン引数の場合）。

  ```
  $ drweb-se --Log Syslog:DAEMON
  ```

- *action* - 特定の脅威または別のイベントが検出されたときにDr.Web for UNIX Mail Serversコンポーネントによって実行されるアクション。次の値を使用できます。
  - `Report` - 他のアクションは実行せず、検出された脅威についての通知のみをするよう指示します。
  - `Cure` - 脅威の修復を試みる（悪意のあるコンテンツのみを削除する）よう指示します。

- `Quarantine` - 感染したファイルを隔離に移動するよう指示します。
- `Delete` - 感染したファイルを削除するよう指示します。

> 一部のアクションは特定のイベントにのみ適用できます（たとえば「スキャンエラー」イベントでは`Cure`（修復）アクションをトリガーできません）。許可されたアクションは、常に*アクション*タイプのパラメータに記述されます。

他のパラメータタイプと可能な値は、パラメータの説明で指定されています。

付録E. SSL証明書を生成する

安全なSSL/TLSデータチャネルと、HTTPS、LDAPS、SMTPSなどのアプリケーションプロトコルを使用するDr.Web for UNIX Mail Serversコンポーネントの場合は、プライベートSSLキーと対応する証明書を提供する必要があります。一部のコンポーネントのキーと証明書は自動的に生成されます。それ以外の場合はDr.Web for UNIX Mail Serversユーザーが作成する必要があります。すべてのコンポーネントがPEN形式の証明書を使用します。

認証局(CA)の検証証明書や署名付き証明書など、SSL/TLSを介した接続に使用されるプライベートキーと証明書を生成するには、コマンドラインユーティリティ`openssl`(OpenSSL暗号化パッケージに含まれる)を使用できます。

プライベートキーとそれに対応するSSL証明書を、CA検証証明書によって署名されたSSL証明書とともに生成するために必要な一連のアクションを検討します。

プライベートSSLキーと証明書を生成する

1. プライベートキー(RSAアルゴリズム、キーの長さは2048ビット)を生成するには、以下のコマンドを実行します。

   ```
   $ openssl genrsa -out keyfile.key 2048
   ```

   キーをパスワードで保護する場合は、`-des3`オプションを使用します。生成されたキーは、カレントディレクトリの`keyfile.key`ファイルにあります。

   キーを表示するには、次のコマンドを使用します。

   ```
   $ openssl rsa -noout -text -in keyfile.key
   ```

2. 既存のプライベートキーに基づいて指定された期間(この場合は365日)の証明書を生成するには、以下のコマンドを実行します。

   ```
   $ openssl req -new -x509 -days 365 -key keyfile.key -out certificate.crt
   ```

   このコマンドは、認証オブジェクトを識別するためのデータ(名前、組織など)を要求します。生成された証明書は、`certificate.crt`ファイルに置かれます。

   生成された証明書の内容をスキャンするには、次のコマンドを使用します。

   ```
   $ openssl x509 -noout -text -in certificate.crt
   ```

証明書を信頼済みCA証明書として登録する

1. 証明書ファイルをシステムの信頼済み証明書ディレクトリ(Debian/Ubuntuの`/etc/ssl/certs/`)に移動またはコピーします。
2. 信頼済み証明書ディレクトリに、証明書へのシンボリックリンクを作成します。リンクの名前は証明書のハッシュ値です。
3. 証明書を含むシステムのディレクトリの内容にインデックスを付け直します。

以下の例では、これら3つすべてのアクションを実行します。ここでは、現在の証明書ディレクトリが信頼済み証明書ディレクトリ/etc/ssl/certs/であり、信頼済み証明書として登録されている証明書が/home/user/ca.crtファイルにあると想定しています。

```
# cp /home/user/ca.crt ./
# ln -s ca.crt `openssl x509 -hash -noout -in ca.crt`.0
# c_rehash /etc/ssl/certs/
```

署名付き証明書を作成する

1. 既存のプライベートキーに基づいて証明書に署名するためのリクエスト(*Certificate Signing Request (CSR)*)を生成します。キーが存在しない場合は、生成します。

   署名リクエストは次のコマンドで作成されます。

   ```
   $ openssl req -new -key keyfile.key -out request.csr
   ```

   このコマンドは、証明書を作成するコマンドと同様に、認証済みオブジェクトを識別するためのデータを要求します。このkeyfile.keyは、プライベートキーの既存のファイルです。受信したリクエストはrequest.csrファイルに保存されます。

   リクエストの作成結果を確認するには、次のコマンドを使用します。

   ```
   $ openssl req -noout -text -in request.csr
   ```

2. 次のコマンドを使用して、リクエストと既存のCA証明書に基づいて署名付き証明書を作成します。

   ```
   $ openssl x509 -req -days 365 -CA ca.crt -CAkey ca.key -set_serial 01 -in request.csr -out sigcert.crt
   ```

   

   署名付き証明書を作成するには、ルート証明書ca.crtとそのプライベートキーca.key(ca.crtとca.keyの代わりにcertificate.crt証明書とkeyfile.keyキーを使用することもできます。その場合、取得した証明書は自己署名されます)、署名リクエストのrequest.csrの3つのファイルが必要です。作成された署名付き証明書はsigcert.crtファイルに保存されます。

   結果を確認するには、次のコマンドを使用します。

   ```
   $ openssl x509 -noout -text -in sigcert.crt
   ```

作成する必要がある一意の証明書の数と同じだけこの手順を繰り返します。たとえば、スキャンクラスタ内の分散ファイルスキャンDr.Web Network Checkerのすべてのエージェントには、独自のキーと証明書が必要です。

署名付き証明書を変更する

一部のブラウザまたはメールクライアントでは、認証に使用される署名済み証明書をPKCS12形式に変更する必要があります。

次のコマンドを使用して証明書を変更できます。

```
# openssl pkcs12 -export -in sigcert.crt -out sigcert.pfx -inkey keyfile.key
```

Sigcert.crtは署名済み証明書の既存ファイルです。keyfile.keyは対応するプライベートキーのファイルです。変更した証明書はsigcert.pfxに保存されます。

付録F. 既知のエラー

この付録の内容

- エラーを特定するための推奨事項
- エラーコード
- コードのないエラー

本セクション内に記載されていないエラーが発生した場合は、テクニカルサポートまでご連絡ください。その際、エラーコードと、問題を再現するための手順をお伝えください。

エラーを特定するための推奨事項

- 考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって/var/log/syslogファイルまたは/var/log/messagesファイルにあります）。また、以下のコマンドを使用することもできます。

```
# drweb-ctl log
```

- エラーを特定するために、個別のファイルにログを記録するよう設定し、ログへの広範な情報の出力を有効にすることが推奨されます。そのために、以下のコマンドを実行してください。

```
# drweb-ctl cfset Root.Log <path to log file>
# drweb-ctl cfset Root.DefaultLogLevel DEBUG
```

- デフォルトのロギング方法とログの詳細レベルに戻すには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.Log -r
# drweb-ctl cfset Root.DefaultLogLevel -r
```

エラーコード

エラーメッセージ: *モニターチャネルに関するエラー*

エラーコード: x1

内部指定: EC_MONITOR_IPC_ERROR

説明: 1つまたは複数のコンポーネントが Dr.Web ConfigD 設定デーモンと接続できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 以下のコマンドを実行することで設定デーモンを再起動させてください。

   ```
   # service drweb-configd restart
   ```

2. PAMの認証メカニズムがインストール、設定されていて、正常に動作していることを確認します。そうでない場合は、インストール・設定します（詳細についてはお使いのOSディストリビューション向けの管理者ガイドとマニュアルを参照してください）。
3. PAMが正常に設定されていて、設定デーモンを再起動しても問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに復元してください。

   これを行うには、たとえば以下のコマンドを実行するなどして、*<etc_dir>*`/drweb.ini` ファイルのコンテンツを削除します（設定ファイル のバックアップを作成することが推奨されます）。

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に設定デーモンを再起動させます。
4. 設定デーモンを起動することができない場合は、`drweb-configd` パッケージを再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *操作はすでに実行中です*

エラーコード: `x2`

内部指定: `EC_ALREADY_IN_PROGRESS`

説明: 操作はすでに実行中です。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 操作が完了するまでお待ちください。必要に応じ、しばらく時間をおいて再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *操作は保留中です*

エラーコード: `x3`

内部指定: `EC_IN_PENDING_STATE`

説明: 要求された操作は保留中です。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにありま

す）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 操作が開始されるまでお待ちください。必要に応じ、しばらく時間をおいて再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ユーザーによって中断されました*

エラーコード: `x4`

内部指定: `EC_INTERRUPTED_BY_USER`

説明: アクションはユーザーによって終了されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *操作がキャンセルされました*

エラーコード: `x5`

内部指定: `EC_CANCELED`

説明: アクションはキャンセルされました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *IPC接続が切断されました*

エラーコード: `x6`

内部指定: `EC_LINK_DISCONNECTED`

説明: Dr.Web for UNIX Mail Serversコンポーネントの1つとのプロセス間通信（IPC）が切断されました長時間アイドル状態であるためにシャットダウンしたと考えられます。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 操作が完了していない場合は、しばらく時間をおいて再度操作を行ってください。そうでない場合、シャットダウンはエラーではありません。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なIPCメッセージサイズです*

エラーコード: x7

内部指定: EC_BAD_MESSAGE_SIZE

説明: コンポーネントのプロセス間通信（IPC）中に無効なサイズのメッセージを受信しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 以下のコマンドを入力し、Dr.Web for UNIX Mail Serversをリロードします。

```
# service drweb-configd restart
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なIPCメッセージフォーマットです*

エラーコード: x8

内部指定: EC_BAD_MESSAGE_FORMAT

説明: コンポーネントのプロセス間通信（IPC）中に無効なフォーマットのメッセージを受信しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 以下のコマンドを入力し、Dr.Web for UNIX Mail Serversをリロードします。

```
# service drweb-configd restart
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *準備が完了していません*

エラーコード: x9

内部指定: EC_NOT_READY

説明: 必要なコンポーネントまたはデバイスがまだ初期化されていないため、要求されたアクションを実行できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *コンポーネントがインストールされていません*

エラーコード: `x10`

内部指定: `EC_NOT_INSTALLED`

説明: 必要なコンポーネントがまだインストールされていないため、必要な操作を実行できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 必要なコンポーネントをインストールまたは再インストールしてください。コンポーネントの名前が分からない場合は、ログファイルを確認して特定してください。
2. 必要なコンポーネントをインストールまたは再インストールしても解決しない場合は、Dr.Web for UNIX Mail Serversを再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *予期せぬIPCメッセージです*

エラーコード: `x11`

内部指定: `EC_UNEXPECTED_MESSAGE`

説明: コンポーネントのプロセス間通信（IPC）中に予期せぬメッセージを受信しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 以下のコマンドを入力し、Dr.Web for UNIX Mail Serversをリロードします。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *IPCプロトコル違反です*

エラーコード: x12

内部指定: EC_PROTOCOL_VIOLATION

説明: コンポーネントのプロセス間通信(IPC)中にプロトコル違反が発生しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 以下のコマンドを入力し、Dr.Web for UNIX Mail Serversをリロードします。

```
# service drweb-configd restart
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *サブシステムの状態が未知です*

エラーコード: x13

内部指定: EC_UNKNOWN_STATE

説明: Dr.Web for UNIX Mail Serversの1つ以上のサブシステムの状態が不明であるため、必要な操作を実行できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 操作を繰り返します。
2. 引き続きエラーが発生する場合は、以下のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させてください。

```
# service drweb-configd restart
```

その後、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *パスは絶対パスでなければなりません*

エラーコード: x20

内部指定: EC_NOT_A_DIRECTORY

説明: ファイルまたはディレクトリへの絶対パスが必要ですが、相対パスが指定されています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. ファイルまたはディレクトリへのパスを絶対パスに変更します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *十分なメモリがありません*

エラーコード: x21

内部指定: EC_NO_MEMORY

説明: 要求された操作を完了するのに十分なメモリがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. Dr.Web for UNIX Mail Serversプロセスが使用可能なメモリのサイズを増やし（ulimit コマンドで上限を変更するなどして）、Dr.Web for UNIX Mail Serversを再起動して操作を繰り返してください。

   場合によっては、システムサービス systemd は指定した上限の変更を無視できます。この場合、ファイル /etc/systemd/system/drweb-configd.service.d/limits.conf を編集し（存在しない場合は作成）、変更後の制限値を指定します。たとえば、次のようになります。

   ```
   [Service]
   LimitDATA = 32767
   ```

   systemd の利用可能な上限のリストはドキュメント man systemd.exec で確認できます。

   以下のコマンドを入力し、Dr.Web for UNIX Mail Serversを再起動します。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *I/Oエラー*

エラーコード: x22

内部指定: EC_IO_ERROR

説明: 入出力（I/O）エラーが発生しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 必要なI/Oデバイスまたはファイルシステムのパーティションが使用可能であるかどうかを確認します。必要に応じ、それをマウントして操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *指定されたファイルまたはディレクトリがありません*

エラーコード: x23

内部指定: EC_NO_SUCH_ENTRY

説明: 指定された、ファイルシステムのオブジェクト(ファイルまたはディレクトリ)がありません。削除された可能性があります。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. パスを確認します。必要に応じ、それを変更して操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *パーミッションが拒否されました*

エラーコード: x24

内部指定: EC_PERMISSION_DENIED

説明: コンポーネントは、指定されたファイルまたはディレクトリにアクセスできません。アイテムにアクセスするために必要な権限がない可能性があります。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. パスが正しいかどうか、また、コンポーネントが要求される権限を持っているかどうかを確認します。オブジェクトにアクセスする必要がある場合、アクセス権限を変更するか、コンポーネントの権限を昇格させます。操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ディレクトリではありません*

エラーコード: x25

内部指定: EC_NOT_A_DIRECTORY

説明: ファイルシステムの、指定されたオブジェクトがディレクトリではありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. パスを確認します。正しいパスを指定して、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *データファイルが破損しています*

エラーコード: x26

内部指定: EC_NOT_A_DIRECTORY

説明: 要求されたデータが破損しています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 操作を繰り返します。
2. 引き続きエラーが発生する場合は、以下のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させてください。

   ```
   # service drweb-configd restart
   ```

   その後、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ファイルはすでに存在しています*

エラーコード: x27

内部指定: EC_FILE_EXISTS

説明: 指定された場所に同じ名前のファイルがすでに存在します。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. パスを確認します。それを変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *読み取り専用ファイルシステム*

エラーコード: x28

内部指定: EC_READ_ONLY_FS

説明: 変更しようとしているファイルシステムオブジェクト（ファイル、ディレクトリ、ソケット）は読み取り専用です。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. パスを確認します。ファイルシステムの書き込み可能なパーティションを指すようにパスを変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ネットワークエラー*

エラーコード: `x29`

内部指定: `EC_NETWORK_ERROR`

説明: ネットワークエラーが発生しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ネットワークが使用可能であること、ネットワーク設定が正しいことを確認します。必要に応じ、ネットワーク設定を変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ドライブではありません*

エラーコード: `x30`

内部指定: `EC_NOT_A_DRIVE`

説明: アクセスしたI/Oデバイスがドライブではありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ドライブ名を確認します。ドライブを指すようにパスを変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *予期せぬEOFがあります*

エラーコード: `x31`

内部指定: `EC_UNEXPECTED_EOF`

説明: データの読み込み中に、予期せずファイルの終わりに達しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ファイル名を確認します。必要に応じ、正しいファイルを指すようにパスを変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ファイルが変更されています*

エラーコード: `x32`

内部指定: `EC_FILE_WAS_CHANGED`

説明: スキャン中のファイルが変更されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 再スキャンしてください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *通常ファイルではありません*

エラーコード: `x33`

内部指定: `EC_NOT_A_REGULAR_FILE`

説明: アクセスされているオブジェクトは通常のファイルではありません（ディレクトリやソケットなどである可能性があります）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ファイル名を確認します。必要に応じ、通常のファイルを指すようにパスを変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *名前はすでに使用されています*

エラーコード: `x34`

内部指定: `EC_NAME_ALREADY_IN_USE`

説明: ファイルシステムに同じ名前のオブジェクトがあるため、オブジェクトを作成できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください

（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. パスを確認します。それを変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ホストがオフラインです*

エラーコード: `x35`

内部指定: `EC_HOST_OFFLINE`

説明: ネットワーク経由でリモートホストにアクセスできません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 必要なホストが使用可能であるかどうかを確認します。必要に応じ、ホストアドレスを変更して操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *リソースの上限に達しています*

エラーコード: `x36`

内部指定: `EC_LIMIT_REACHED`

説明: 特定のリソースの使用について設定された上限に達しています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 必要なリソースの使用可能状況を確認します。必要に応じ、このリソースの使用に関する上限を引き上げて、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *異なるマウントポイントです*

エラーコード: `x37`

内部指定: `EC_CROSS_DEVICE_LINK`

説明: ファイルを復元できません。復元は、2つの異なるマウントポイント間でファイルを移動を行います。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください

（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ファイルを復元する別のパスを選択し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *アンパックエラー*

エラーコード: `x38`

内部指定: `EC_UNPACKING_ERROR`

説明: アーカイブの展開に失敗しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. アーカイブが破損していないことを確認します。アーカイブがパスワード保護されている場合、正しいパスワードを入力することで保護を解除し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ウイルスデータベースが破損しています*

エラーコード: `x40`

内部指定: `EC_BASE_CORRUPTED`

説明: ウイルスデータベースが破損しています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ウイルスデータベースディレクトリへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの `[Root]` セクションにある `VirusBaseDir` パラメータ）。
   - パスを表示および変更するには、Webインターフェースの全般設定ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.VirusBaseDir
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir <new path>
   ```

パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースのメインページで**更新**をクリックします。
   - または、コマンドを実行します。

```
$ drweb-ctl update
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *サポートされていないバージョンのウイルスデータベースです*

エラーコード: x41

内部指定: EC_OLD_BASE_VERSION

説明: 現在のウイルスデータベースは以前のバージョンのプログラム向けのものです。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ウイルスデータベースディレクトリへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの `[Root]` セクションにある `VirusBaseDir` パラメータ）。
   - パスを表示および変更するには、Webインターフェースの**全般設定**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

```
$ drweb-ctl cfshow Root.VirusBaseDir
```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.VirusBaseDir <new path>
```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースのメインページで**更新**をクリックします。
   - または、コマンドを実行します。

```
$ drweb-ctl update
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ウイルスデータベースが空です*

エラーコード: x42

内部指定: EC_EMPTY_BASE

説明: ウイルスデータベースが空です。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. ウイルスデータベースディレクトリへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの [Root] セクションにある VirusBaseDir パラメータ）。
   - パスを表示および変更するには、Webインターフェースの**全般設定**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.VirusBaseDir
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir -r
   ```
2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースの**メイン**ページで**更新**をクリックします。
   - または、コマンド を実行します。

   ```
   $ drweb-ctl update
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *オブジェクトを修復できません*

エラーコード: x43

内部指定: EC_CAN_NOT_BE_CURED

説明: 修復不可能なオブジェクトに対して**修復** アクションが適用されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. オブジェクトに対して適用可能なアクションを選択し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *サポートされていないウイルスデータベースの組み合わせです*

エラーコード: `x44`

内部指定: `EC_INVALID_BASE_SET`

説明: 現在のウイルスデータベースの組み合わせはサポートされていません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ウイルスデータベースディレクトリへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの `[Root]` セクションにある `VirusBaseDir` パラメータ）。
   - パスを表示および変更するには、Webインターフェースの**全般設定**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.VirusBaseDir
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir -r
   ```
2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースの**メイン**ページで**更新**をクリックします。
   - または、コマンド `drweb-ctl update` を実行します。

   ```
   $ drweb-ctl update
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *スキャンの上限に達しています*

エラーコード: `x45`

内部指定: `EC_SCAN_LIMIT_REACHED`

説明: オブジェクトのスキャン中に、指定された上限を超えました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 以下のいずれかの方法で、スキャンにおける上限を変更します（コンポーネント設定内で）。
   - Webインターフェース のコンポーネント設定のあるページ（インストールされている場合）を使用。
   - `drweb-ctl cfshow` および `drweb-ctl cfset` コマンド を使用。
2. 設定の変更後、試みた操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *認証に失敗しました*

エラーコード: `x47`

内部指定: `EC_AUTH_FAILED`

説明: 認証に、無効なユーザー認証情報が使用されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 必要な権限を持ったユーザーの有効な認証情報を入力してください。再度、認証を実行してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *認証に失敗しました*

エラーコード: `x48`

内部指定: `EC_NOT_AUTHORIZED`

説明: ユーザーには、要求された操作を実行するための十分な権限がありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 必要な権限を持ったユーザーの有効な認証情報を入力してください。再度、認証を実行してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なアクセストークンです*

エラーコード: `x49`

内部指定: `EC_INVALID_TOKEN`

説明：Dr.Web for UNIX Mail Serversコンポーネントの1つが、昇格された権限を必要とする操作へのアクセスを試みる際に無効な認証トークンを提示しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 必要な権限を持ったユーザーの有効な認証情報を入力してください。再度、認証を実行してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*DBMSの一般的なエラーです*

エラーコード：`x50`

内部指定：`EC_DB_COMMON_ERROR`

説明：Dr.Web LookupDによるDBMSサーバーへのリクエストは成功しませんでした。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

DMBSサーバーログも参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*データベースを開くことができません*

エラーコード：`x51`

内部指定：`EC_DB_OPEN_ERROR`

説明：Dr.Web LookupDが接続しようとしているデータベースが利用できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

DMBSサーバーログも参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*DBMSによって接続が閉じられました*

エラーコード：`x52`

内部指定：`EC_DB_CONN_CLOSED`

説明：接続がDBMSサーバーによって閉じられました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

DMBSサーバーログも参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効な引数です*

エラーコード: x60

内部指定: EC_INVALID_ARGUMENT

説明: コマンドを実行しようとした際に無効な引数が使用されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 有効な引数を使用して、再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効な操作です*

エラーコード: x61

内部指定: EC_INVALID_OPERATION

説明: 無効なコマンドを実行しようとする試みが検出されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 有効なコマンドを使用して、再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *スーパーユーザー権限が必要です*

エラーコード: x62

内部指定: EC_ROOT_ONLY

説明: アクションを実行するには、スーパーユーザー権限が必要です。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 権限をroot権限に昇格させ、再度アクションを実行します。権限を昇格させるには、su または sudo コマンドを使用します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *集中管理モードでは許可されていません*

エラーコード: x63

内部指定: EC_STANDALONE_MODE_ONLY

説明: 要求されたアクションは、スタンドアロンモードで動作している場合のみ実行できます。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. Dr.Web for UNIX Mail Serversの動作モードをスタンドアロンモードに変更し、操作を繰り返します。
2. Dr.Web for UNIX Mail Serversをスタンドアロンモードに切り替えるには:
   - Webインターフェースがインストールされている場合は、Webインターフェースの**集中管理**の**集中管理モードを有効にする**チェックボックスをオフにします。
   - または、コマンドを実行します。

```
# drweb-ctl esdisconnect
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *サポートされていないOSです*

エラーコード: x64

内部指定: EC_NON_SUPPORTED_OS

説明: Dr.Web for UNIX Mail Serversは、ホスト上にインストールされているOSをサポートしていません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. システム要件のリスト内に記載されているOSをインストールします。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *実装されていない機能です*

エラーコード: x65

内部指定: EC_NOT_IMPLEMENTED

説明: 1つまたは複数のコンポーネントに必要な機能が、現在のバージョンのプログラムには備わっていません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください

（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. Dr.Web for UNIX Mail Serversの設定をデフォルトに復元します。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*`/drweb.ini`ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）。

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に、次のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させます。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *未知のオプションです*

エラーコード: `x66`

内部指定: `EC_UNKNOWN_SECTION`

説明: 設定ファイルに、未知のパラメータまたはDr.Web for UNIX Mail Serversの現在のバージョンでサポートされていないパラメータが含まれています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. いずれかのテキストエディターで *<etc_dir>*`/drweb.ini` ファイルを開き、無効なパラメータが含まれる行を削除します。ファイルを保存し、次のコマンドを実行することで Dr.Web ConfigD 設定デーモンを再起動させます。

   ```
   # service drweb-configd restart
   ```

2. 問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してしてください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*`/drweb.ini` ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）:

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に設定デーモンを再起動させます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *未知のセクションです*

エラーコード: x67

内部指定: EC_UNKNOWN_SECTION

説明: 設定ファイルに、未知のセクションまたはDr.Web for UNIX Mail Serversの現在のバージョンでサポートされていないセクションが含まれています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. いずれかのテキストエディターで *<etc_dir>*/drweb.ini ファイルを開き、未知の（サポートされていない）セクションを削除します。ファイルを保存し、次のコマンドを実行することで Dr.Web ConfigD 設定デーモンを再起動させます。

   ```
   # service drweb-configd restart
   ```

2. 問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してしてください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*/drweb.ini ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）:

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に設定デーモンを再起動させます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なオプション値です*

エラーコード: x68

内部指定: EC_INVALID_OPTION_VALUE

説明: 設定ファイル 内の1つまたは複数のパラメータに、無効な値が含まれています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 以下のいずれかの方法で、有効なパラメータ値を設定します。
   - Webインターフェース のコンポーネント設定のあるページ（インストールされている場合）を使用。
   - drweb-ctl cfshow および drweb-ctl cfset コマンド を使用。

   当該パラメータの有効な値が分からない場合は、そのパラメータを使用するコンポーネントのヘルプファイルを参照してください。パラメータ値をデフォルト値に戻すこともできます。
2. 設定ファイル *<etc_dir>*/drweb.ini を直接編集することも可能です。その場合は、いずれかのテキストエディターで設定ファイルを開き、無効なパラメータ値を含む行を見つけ、有効な値を設定してください。その後、ファイルを保存し、以下のコマンドを実行することで Dr.Web ConfigD 設定デーモンを再起動させます。

   ```
   # service drweb-configd restart
   ```

3. この手順で問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*/drweb.ini ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）：

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に設定デーモンを再起動させます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*無効な状態です*

エラーコード：x69

内部指定：EC_INVALID_STATE

説明：Dr.Web for UNIX Mail Serversまたはコンポーネントの1つが無効な状態であるため、必要な操作を完了できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度アクションを実行します。
2. 引き続きエラーが発生する場合は、以下のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させてください。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*使用可能な値は1つのみです*

エラーコード：x70

内部指定：EC_NOT_LIST_OPTION

説明：設定ファイルでは、値のリストは単一値パラメータに関連付けられています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 以下のいずれかの方法で、有効なパラメータ値を設定します。
   - Webインターフェースのコンポーネント設定のあるページ（インストールされている場合）を使用。
   - drweb-ctl cfshow および drweb-ctl cfset コマンドを使用。

   当該パラメータの有効な値が分からない場合は、そのパラメータを使用するコンポーネントのヘルプファイルを参照してください。パラメータ値をデフォルト値に戻すこともできます。

2. 設定ファイル *<etc_dir>*/drweb.ini を直接編集することも可能です。その場合は、いずれかのテキストエディターで設定ファイルを開き、無効なパラメータ値を含む行を見つけ、有効な値を設定してください。その後、ファイルを保存し、以下のコマンドを実行することで Dr.Web ConfigD 設定デーモンを再起動させます。

```
# service drweb-configd restart
```

3. この手順で問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*/drweb.ini ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）：

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```

   設定ファイルのコンテンツを削除した後に設定デーモンを再起動させます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*無効なタグ値です*

エラーコード：x71

内部指定：EC_INVALID_TAG

説明：Dr.Web LookupDコンポーネントが連携するデータソースの名前に、正しくない、または無効な（存在しない）タグが使用されています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. タグのスペルが正しいかどうかを確認します。誤りを見つけた場合は、設定ファイル の該当するセクションを編集してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*レコードが見つかりません*

エラーコード：x80

内部指定：EC_RECORD_NOT_FOUND

説明：脅威のレコードがありません（他のDr.Web for UNIX Mail Serversコンポーネントによってすでに処理されている可能性があります）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. しばらくしてから脅威のリストを更新してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *レコードは現在処理中です*

エラーコード: x81

内部指定: EC_RECORD_BUSY

説明: レコードはすでに別のコンポーネントによって処理されています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. しばらくしてから脅威のリストを更新してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ファイルはすでに隔離済みです*

エラーコード: x82

内部指定: EC_RECORD_BUSY

説明: 検出された脅威を含むファイルを隔離に移動しようとした際に、ファイルがすでに隔離されていることが明らかになりました（脅威が別のコンポーネントによって処理された可能性があります）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. しばらくしてから脅威のリストを更新してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *更新ゾーンはクラウドで提供されていません*

エラーコード: x83

内部指定: EC_NO_ZONE_IN_CLOUD

説明: Dr.Web Cloudを使用して更新しようとしましたが、失敗しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度アクションを実行します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *更新ゾーンはディスクで提供されません*

エラーコード: x84

内部指定: EC_NO_ZONE_ON_DISK

説明: オフラインモードでウイルスベースを更新しようとしましたが、失敗しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 更新に使用するデバイスへのパスが正しいことを確認します。
2. ベースを更新しようとするユーザーが、更新を含むディレクトリに対して必要な読み取り権限を持っていることを確認します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *更新前にバックアップを行うことができません*

エラーコード: x89

内部指定: EC_BACKUP_FAILED

説明: 更新サーバーから更新をダウンロードする前に対象となるファイルのバックアップコピーを作成しようとする試みが失敗しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 更新されたファイルのバックアップコピーを保存するディレクトリへのパスを確認します。必要に応じてパスを変更します（設定ファイルの [Update] セクションにある BackupDir パラメータ）。
   - パスを表示および変更するには、Webインターフェースの**Updater**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Update.BackupDir
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Update.BackupDir <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Update.BackupDir -r
   ```
2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースのメインページで**更新**をクリックします。

- または、コマンド を実行します。

```
$ drweb-ctl update
```

3. 引き続きエラーが発生する場合は、Dr.Web Updaterコンポーネントを実行しているアカウントのユーザーが、BackupDir で指定されたディレクトリへの書き込み権限を持っているかどうかを確認してください。このユーザーの名前は、RunAsUser パラメータで指定されます。必要に応じて、RunAsUser パラメータで指定されているユーザーを変更するか、ディレクトリのプロパティで足りない権限を付与します。
4. それでもエラーが続く場合は、drweb-update パッケージを再インストールします。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なDRLファイルです*

エラーコード: x90

内部指定: EC_BAD_DRL_FILE

説明: 更新サーバーのリストが含まれているファイルの1つの整合性が侵害されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. サーバーのリストを含むファイルへのパスを確認し、必要に応じてパスを変更します（設定ファイルの [Update] セクションにある *DrlDir を持つパラメータ）。
   - パスを表示および変更するには、Webインターフェースの**Updater** ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を表示するには、コマンドを使用します（*<*DrlDirPath>* は、指定されたパラメータ名に置き換える必要があります。パラメータ名が不明な場合は、セクション内のパラメータ値を参照します。角括弧内のコマンド部分は省略します。

   ```
   $ drweb-ctl cfshow Update[.<*DrlDir>]
   ```

   新しいパラメータ値を設定するには、コマンドを実行します（*<*DrlDir>* は、指定されたパラメータ名に置き換える必要があります。

   ```
   # drweb-ctl cfset Update.<*DrlDir> <new path>
   ```

   パラメータ値をデフォルトに戻すには、コマンドを実行します（*<*DrlDir>* は、指定されたパラメータ名に置き換える必要があります。

   ```
   # drweb-ctl cfset Update.<*DrlDir> -r
   ```

2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェース の メイン ページで **更新** をクリックします。

- または、コマンド を実行します。

```
$ drweb-ctl update
```

3. それでもエラーが続く場合は、`drweb-update` パッケージを再インストールします。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なLSTファイルです*

エラーコード: `x91`

内部指定: `EC_BAD_LST_FILE`

説明: 更新されたウイルスデータベースのリストが含まれているファイルの整合性が侵害されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度ウイルスデータベースを更新します:
   - Webインターフェースがインストールされている場合は、Webインターフェース の メイン ページで **更新** をクリックします。
   - または、コマンド を実行します。

```
$ drweb-ctl update
```

2. それでもエラーが続く場合は、`drweb-update` パッケージを再インストールします。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効な圧縮ファイルです*

エラーコード: `x92`

内部指定: `EC_BAD_LZMA_FILE`

説明: 更新が含まれているダウンロードされたファイルで、整合性侵害が検出されました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度ウイルスデータベースを更新します:
   - Webインターフェースがインストールされている場合は、Webインターフェース の メイン ページで **更新** を

クリックします。

- または、コマンド を実行します。

```
$ drweb-ctl update
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *プロキシ認証エラーです*

エラーコード: x93

内部指定: EC_PROXY_AUTH_ERROR

説明: プログラムは、設定内で指定されたプロキシサーバーを使用して更新サーバーに接続できませんでした。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. プロキシサーバーへの接続に使用されているパラメータを確認します（設定ファイルの [Update] セクションの Proxy パラメータで設定されています）。
   - 接続パラメータを表示および設定するには、Webインターフェース の **Updater** ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Update.Proxy
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Update.Proxy <new parameters>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Update.Proxy -r
   ```
2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェース の メイン ページで **更新** をクリックします。
   - または、コマンド を実行します。

   ```
   $ drweb-ctl update
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *使用可能な更新サーバーがありません*

エラーコード: x94

内部指定：EC_NO_UPDATE_SERVERS

説明：プログラムは、いずれの更新サーバーにも接続できませんでした。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. ネットワークが使用可能であるかどうかを確認します。必要に応じ、ネットワーク設定を変更します。
2. プロキシサーバーのみを使用してネットワークにアクセスできる場合は、プロキシサーバーに接続するためのパラメータを設定します（設定ファイルの [Update] セクションの Proxy パラメータで設定できます）。
   - 接続パラメータを表示および設定するには、Webインターフェース の **Updater** ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Update.Proxy
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Update.Proxy <new parameters>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Update.Proxy -r
   ```
3. ネットワーク接続パラメータ（プロキシサーバーのパラメータを含む）が正しくてもエラーが発生する場合は、利用可能な更新サーバーのリストを使用していることを確認してください。使用されている更新サーバーのリストは、設定ファイルの [Update] セクションのパラメータ *DrlDir で表示されます。

   > *CustomDrlDir パラメータが既存の正しいサーバーリストのファイルを示す場合、標準的な更新ゾーンのサーバーではなく、リスト内で指定されたサーバーが使用されます（対応する*DrlDir パラメータで指定されている値は無視されます）。

   - 接続パラメータを表示および設定するには、Webインターフェース の **Updater** ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を表示するには、コマンドを使用します（*<*DrlDirPath>* は、指定されたパラメータ名に置き換える必要があります。パラメータ名が不明な場合は、セクション内のパラメータ値を参照します。角括弧内のコマンド部分は省略します。

   ```
   $ drweb-ctl cfshow Update[.<*DrlDir>]
   ```

   新しいパラメータ値を設定するには、コマンドを実行します（*<*DrlDir>* は、指定されたパラメータ名に置き換える必要があります。

   ```
   # drweb-ctl cfset Update.<*DrlDir> <new path>
   ```

パラメータ値をデフォルトに戻すには、コマンドを実行します（*<*DrlDir>* は、指定されたパラメータ名に置き換える必要があります。

```
# drweb-ctl cfset Update.<*DrlDir> -r
```

4. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェース の メイン ページで **更新** をクリックします。
   - または、コマンド を実行します。

```
$ drweb-ctl update
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *キーファイルのフォーマットが無効です*

エラーコード: x95

内部指定: EC_BAD_KEY_FORMAT

説明: キーファイルのフォーマットがサポートされていません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. キーファイルを持っているかどうか、また、キーファイルへのパスを確認します。キーファイルへのパスは、設定ファイル の [Root] セクション の KeyPath パラメータで指定できます。
   - キーファイルへのパスを表示および設定するには、Webインターフェースの **全般設定** ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

```
$ drweb-ctl cfshow Root.KeyPath
```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.KeyPath <path to file>
```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.KeyPath -r
```

2. キーファイルをお持ちでない場合や、使用しているキーファイルが破損している場合は、キーファイルを購入してインストールしてください。キーファイル、購入、インストールに関する詳細については ライセンス のセクションを参照してください。
3. キーファイルをインストールするには、Webインターフェースのメインページの下部にあるライセンス有効化フォームを使用できます（インストールされている場合）。
4. また、https://support.drweb.com/get+cabinet+link/のユーザーのWebページ**My Dr.Web**で、現在のライセンスオプションを確認できます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ライセンスの有効期間が満了しています*

エラーコード: x96

内部指定: EC_EXPIRED_KEY

説明: ライセンスの有効期限が切れています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 新しいライセンスを購入して取得したキーファイルをインストールします。ライセンスの購入方法とキーファイルのインストールの詳細については、ライセンス のセクションを参照してください。
2. 購入したキーファイルをインストールするには、Webインターフェースのメインページの下部にあるライセンス有効化フォームを使用できます（インストールされている場合）。
3. また、https://support.drweb.com/get+cabinet+link/のユーザーのWebページ**My Dr.Web**で、現在のライセンスオプションを確認できます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ネットワークオペレーションがタイムアウトしました*

エラーコード: x97

内部指定: EC_NETWORK_TIMEDOUT

説明: ネットワークオペレーションがタイムアウトしました（リモートホストが予期せず応答を停止したか、必要な接続に失敗した可能性があります）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. ネットワークが使用可能であること、ネットワーク設定が正しいことを確認します。必要に応じ、ネットワーク設定を変更し、操作を繰り返します。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なチェックサムです*

エラーコード: x98

内部指定: EC_BAD_CHECKSUM

説明: 更新プログラムでダウンロードしたファイルのチェックサムが破損しています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください

（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. しばらく時間をおいて再度ウイルスデータベースを更新します：
   - Webインターフェースがインストールされている場合は、Webインターフェースのメインページで**更新**をクリックします。
   - または、コマンドを実行します。

   ```
   $ drweb-ctl update
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*無効なトライアルライセンスです*

エラーコード：`x99`

内部指定：`EC_BAD_TRIAL_KEY`

説明：デモキーファイルが無効です（別のコンピューターから受け取った場合など）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 該当するコンピューターの新しい試用期間のリクエストを送信するか、新しいライセンスを購入して、受け取るキーファイルをインストールしてください。ライセンスの購入とキーファイルのインストールに関する詳細については ライセンス のセクションを参照してください。
2. 購入したキーファイルをインストールするには、Webインターフェースのメインページの下部にあるライセンス有効化フォームを使用できます（インストールされている場合）。
3. また、https://support.drweb.com/get+cabinet+link/のユーザーのWebページ**My Dr.Web**で、現在のライセンスオプションを確認できます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*ブロックされているライセンスキーです*

エラーコード：`x100`

内部指定：`EC_BLOCKED_LICENSE`

説明：使用中のライセンスがブロックされています（Dr.Web for UNIX Mail Serversの利用規約違反の可能性があります）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 新しいライセンスを購入して取得したキーファイルをインストールします。ライセンスの購入方法とキーファイルのインストールの詳細については、ライセンス のセクションを参照してください。

2. 受領したキーファイルをインストールするには、Webインターフェースのメイン ページの下部にあるライセンス有効化フォームを使用できます（インストールされている場合）。
3. また、https://support.drweb.com/get+cabinet+link/のユーザーのWebページ**My Dr.Web**で、現在のライセンスオプションを確認できます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なライセンスです*

エラーコード: x101

内部指定: EC_BAD_LICENSE

説明: ライセンスが別の製品のものであるか、Dr.Web for UNIX Mail Serversコンポーネントの動作がライセンスで許可されていません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 新しいライセンスを購入して取得したキーファイルをインストールします。ライセンスの購入方法とキーファイルのインストールの詳細については、ライセンス のセクションを参照してください。
2. 受領したキーファイルをインストールするには、Webインターフェースのメイン ページの下部にあるライセンス有効化フォームを使用できます（インストールされている場合）。
3. また、https://support.drweb.com/get+cabinet+link/のユーザーのWebページ**My Dr.Web**で、現在のライセンスオプションを確認できます。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効な設定です*

エラーコード: x102

内部指定: EC_BAD_CONFIG

説明: 設定が正しくないため、1つ以上のDr.Web for UNIX Mail Serversコンポーネントが動作できません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. エラーが発生しているコンポーネントの名前が分からない場合は、ログファイルを確認して特定してください。
2. このエラーがDr.Web Firewall for Linuxによって引き起こされる場合、別のファイアウォールとの競合が発生している可能性があります。たとえば、Dr.Web Firewall for LinuxはFedora、CentOS、Red Hat Enterprise LinuxのFirewallDと競合することが確認されています（起動するたびに、FirewallDはDr.Web Firewall for Linuxで指定されるトラフィックルーティングルールを無効にします）。

このエラーを解決するには、以下のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させてください。

```
# service drweb-configd restart
```

または

```
# drweb-ctl reload
```

FirewallDの動作を許可する場合、Dr.Web Firewall for LinuxのエラーはFirewallDが再起動（OSの再起動を含む）する度に繰り返し発生する可能性があります。FirewallDを無効にすることで、このエラーを解決することができます（お使いのOSに付属しているFirewallDのマニュアルを参照してください）。

3. エラーが別のコンポーネントによって発生している場合、以下のいずれかの方法によって、そのコンポーネントの設定をデフォルトに復元してください。
   - Webインターフェース のコンポーネント設定のあるページ（インストールされている場合）を使用。
   - `drweb-ctl cfshow` および `drweb-ctl cfset` コマンド を使用。
   - 手動で 設定ファイル を編集（コンポーネントセクションからすべてのパラメータを削除してください）。
4. この手順で問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*`/drweb.ini` ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）：

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に、次のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させます。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*無効な実行ファイルです*

エラーコード：`x104`

内部指定：`EC_BAD_EXECUTABLE`

説明：コンポーネントの実行ファイルが破損しています。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. エラーが発生しているコンポーネントの名前が分からない場合は、ログファイルを確認して特定してください。

2. 以下のコマンドを実行することで（*<component section>*を、設定ファイルの該当するセクション名に変更します）、Dr.Web for UNIX Mail Servers設定ファイル内でコンポーネントの実行可能パスを確認してください（コンポーネントセクションの `ExePath` パラメータ）。

   ```
   $ drweb-ctl cfshow <component section>.ExePath
   ```

3. 以下のコマンドを実行することで（*<component section>*を、設定ファイルの該当するセクション名に変更します）、パスをデフォルトに復元します。

   ```
   # drweb-ctl cfset <component section>.ExePath -r
   ```

4. この手順で問題が解決しない場合は、該当するコンポーネントのパッケージを再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *コアエンジンは使用できません*

エラーコード: `x105`

内部指定: `EC_NO_CORE_ENGINE`

説明: Dr.Web Virus-Finding Engineのファイルが見つからないか、使用できません（脅威の検出に必要です）。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb32.dll` スキャンエンジンファイルへのパスを確認します。必要に応じてパスを変更します（設定ファイルの `[Root]` セクションにある `CoreEnginePath` パラメータ）。
   - パスを表示および変更するには、Webインターフェースの**全般設定**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.CoreEnginePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.CoreEnginePath <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.CoreEnginePath -r
   ```

2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースの**メイン**ページで**更新**をクリックします。

- または、コマンド を実行します。

```
$ drweb-ctl update
```

3. パスが正しく、ウイルスデータベースを更新した後もエラーが続く場合は、`drweb-bases` パッケージを再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ウイルスデータベースがありません*

エラーコード: `x106`

内部指定: `EC_NO_VIRUS_BASES`

説明: ウイルスデータベースが見つかりません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. ウイルスデータベースディレクトリへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの `[Root]` セクションにある `VirusBaseDir` パラメータ）。
   - パスを表示および変更するには、Webインターフェースの**全般設定**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.VirusBaseDir
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.VirusBaseDir -r
   ```

2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェース の メイン ページで **更新** をクリックします。
   - または、コマンド を実行します。

   ```
   $ drweb-ctl update
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *プロセスはシグナルによって中断されました*

エラーコード: x107

内部指定: EC_APP_TERMINATED

説明: コンポーネントがシャットダウンしました(ユーザーコマンドによって、またはアイドル状態であるためなど)。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 操作が完了していない場合は、再度開始してください。そうでない場合、シャットダウンはエラーではありません。
2. コンポーネントが度々シャットダウンする場合は、以下のいずれかの方法によって、そのコンポーネントの設定をデフォルトに復元します。
   - Webインターフェース のコンポーネント設定のあるページ(インストールされている場合)を使用。
   - drweb-ctl cfshow および drweb-ctl cfset コマンド を使用。
   - 手動で 設定ファイル を編集(コンポーネントセクションからすべてのパラメータを削除してください)。
3. 問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してしてください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*/drweb.ini ファイルのコンテンツを削除します(設定ファイルのバックアップを作成することが推奨されます):

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に、次のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させます。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *予期せぬプロセスの中断です*

エラーコード: x108

内部指定: EC_APP_CRASHED

説明: 不具合によってコンポーネントがシャットダウンしました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. 中断された操作を繰り返します。
2. コンポーネントが度々異常にシャットダウンする場合は、以下のいずれかの方法によって、そのコンポーネントの設定をデフォルトに復元します。
   - Webインターフェース のコンポーネント設定のあるページ(インストールされている場合)を使用。

- `drweb-ctl cfshow` および `drweb-ctl cfset` コマンドを使用。
- 手動で 設定ファイル を編集（コンポーネントセクションからすべてのパラメータを削除してください）。

3. 問題が解決しない場合は、Dr.Web for UNIX Mail Servers設定をデフォルトに戻してしてください。

   そのために、たとえば以下のコマンドを実行するなどして、*<etc_dir>*`/drweb.ini` ファイルのコンテンツを削除します（設定ファイルのバックアップを作成することが推奨されます）：

   ```
   # cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
   # echo "" > /etc/opt/drweb.com/drweb.ini
   ```

   設定ファイルのコンテンツを削除した後に、次のコマンドを実行することでDr.Web for UNIX Mail Serversを再起動させます。

   ```
   # service drweb-configd restart
   ```

4. Dr.Web for UNIX Mail Servers設定を復元した後もエラーが続く場合は、コンポーネントパッケージを再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ：*互換性のないソフトウェアが検出されました*

エラーコード：`x109`

内部指定：`EC_INCOMPATIBLE`

説明：Dr.Web for UNIX Mail Serversの1つまたは複数のコンポーネントが正しく動作できません。コンポーネントのシステムでの動作を妨げているソフトウェアがあります。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. このエラーがSpIDer Gateによって発生した場合は、互換性のないプログラムがOSで実行されている可能性があります。このプログラムはNetFilterシステムのファイアウォールのルールを生成し、SpIDer Gateの正しい動作を妨げている可能性があります。おそらく、ShorewallまたはSuseFirewall2がシステムにインストールされています（SUSE Linux OSの場合）。NetFilterシステムのファイアウォールを設定するアプリケーションは、指定されたルールシステムの整合性をチェックし、それを書き換えることがあります。これが、このようなアプリケーションとSpIDer Gateが競合する主な原因と考えられます。

   SpIDer Gateの動作に干渉しないように、互換性のないソフトウェアを再設定してください。それができない場合は、OSの起動時に読み込まれないようにソフトウェアを無効にしてください。次の手順によって、SuseFirewall2アプリケーション（SUSE Linux OS）を設定できます。

   1) SuseFirewall2の設定ファイル（デフォルトでは `/etc/sysconfig/SuSEfirewall2` ファイルです）を開きます。
   2) 以下のラインを見つけます。

   ```
   # Type: yesno
   #
   # Install NOTRACK target for interface lo in the raw table. Doing so
   ```

```
# speeds up packet processing on the loopback interface. This breaks
# certain firewall setups that need to e.g. redirect outgoing
# packets via custom rules on the local machine.
#
# Defaults to "yes" if not set
#
FW_LO_NOTRACK=""
```

3) FW_LO_NOTRACK パラメータ値に "no"を指定します:

```
FW_LO_NOTRACK="no"
```

4) SuseFirewall2を再起動します。

```
# rcSuSEfirewall2 restart
```

SuseFirewall2 設定 に FW_LO_NOTRACK オプションがない場合は、アプリケーションを停止し、システム起動時のアプリの自動起動を無効にします。

競合アプリケーションを再設定または無効にした後、SpIDer Gateを再起動してください。

2. エラーが別のコンポーネントによって発生している場合、互換性のないソフトウェアを無効にするか、再設定し、それがDr.Web for UNIX Mail Serversの動作を妨げないようにしてください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *無効なライブラリです*

エラーコード: x110

内部指定: EC_BAD_VADERETRO_LIB

説明: アンチスパムライブラリのファイルが見つからないか、使用できないか、あるいは破損しています(メールのスキャンに必要です)。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください(デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります)。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. ライブラリファイルへのパスを確認してください。必要に応じてパスを変更します(設定ファイル の [Root] セクション にある AntispamCorePath パラメータ)。
   - パスを表示および変更するには、Webインターフェースの全般設定ページに移動します(インストールされている場合)。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.AntispamCorePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.AntispamCorePath <new path>
   ```

パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset Root.AntispamCorePath -r
```

2. ウイルスデータベースを更新します。
   - Webインターフェースがインストールされている場合は、Webインターフェースのメインページで**更新**をクリックします。
   - または、コマンドを実行します。

   ```
   $ drweb-ctl update
   ```

3. パスが正しく、ウイルスデータベースを更新した後もエラーが続く場合は、`drweb-maild`パッケージを再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *Webリソースデータベースがありません*

エラーコード: `x112`

内部指定: `EC_NO_DWS_BASES`

説明: Webリソースカテゴリーのデータベースがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. Webリソースカテゴリーディレクトリのデータベースへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの[Root]セクションにあるDwsDirパラメータ）。
   - パスを表示および変更するには、Webインターフェースの**全般設定**ページに移動します（インストールされている場合）。
   - または、コマンドライン管理ツールのコマンドを使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Root.DwsDir
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.DwsDir <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Root.DwsDir -r
   ```

2. Webリソースカテゴリーのデータベースを更新:
   - Webインターフェースがインストールされている場合は、Webインターフェースのメインページで**更新**を

クリックします。

- または、コマンド を実行します。

```
$ drweb-ctl update
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *MeshDは使用できません*

エラーコード: x114

内部指定: EC_NO_MESHD

説明: ファイルスキャン時の負荷分散に必要なDr.Web MeshDコンポーネントがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. drweb-meshd 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの [MeshD] セクション にある ExePath パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow MeshD.ExePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset MeshD.ExePath <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset MeshD.ExePath -r
   ```

2. 設定にDr.Web MeshDコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、drweb-meshd パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *LookupDは使用できません*

エラーコード: x115

内部指定: EC_NO_LOOKUPD

説明: 外部ソースからデータを取得するために必要なDr.Web LookupDコンポーネントがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb-lookupd` 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイル の `[LookupD]` セクション にある `ExePath` パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow LookupD.ExePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset LookupD.ExePath <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset LookupD.ExePath -r
   ```

2. 設定にDr.Web LookupDコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、`drweb-lookupd` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *UrlCheckは使用できません*

エラーコード: `x116`

内部指定: `EC_NO_URL_CHECK`

説明: 潜在的に危険な可能性があるカテゴリーや不要なカテゴリーに属しているURL接続をチェックするために必要な、Dr.Web URL Checkerコンポーネントが見つかりません

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb-urlcheck` 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの `[URLCheck]` セクションにある `ExePath` パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow URLCheck.ExePath
   ```

新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset URLCheck.ExePath <new path>
```

パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset URLCheck.ExePath -r
```

2. 設定にDr.Web URL Checkerコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、`drweb-urlcheck` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *GateDは使用できません*

エラーコード: `x117`

内部指定: `EC_NO_GATED`

説明: ネットワーク接続のスキャンに必要なSpIDer Gateコンポーネントがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb-gated` 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの [`GateD`] セクション にある `ExePath` パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

```
$ drweb-ctl cfshow GateD.ExePath
```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset GateD.ExePath <new path>
```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset GateD.ExePath -r
```

2. 設定にSpIDer Gateコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、`drweb-gated` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *MailDは使用できません*

エラーコード: x118

内部指定: EC_NO_MAILD

説明: メールスキャンに必要なDr.Web MailDコンポーネントが見つかりません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. drweb-maild 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの [MailD] セクション にある ExePath パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow MailD.ExePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset MailD.ExePath <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset MailD.ExePath -r
   ```

2. 設定にDr.Web MailDコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、drweb-maild パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ScanEngineは使用できません*

エラーコード: x119

内部指定: EC_NO_SCAN_ENGINE

説明: 脅威の検出に必要なDr.Web Scanning Engineコンポーネントが見つからないか、起動に失敗しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. drweb-se 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイル の

[ScanEngine] セクション にある ExePath パラメータ）。

または、コマンドライン管理ツールの コマンド を使用することもできます。

現在のパラメータ値を確認するには、以下のコマンドを実行します。

```
$ drweb-ctl cfshow ScanEngine.ExePath
```

新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset ScanEngine.ExePath <new path>
```

パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset ScanEngine.ExePath -r
```

2. 正しいパスを入力した後もエラーが継続する場合は
   - 次のコマンドを実行します。

   ```
   $ drweb-ctl rawscan /
   ```

   「エラー: 有効なライセンスが指定されていません」が出力された場合、有効なキーファイルがありません。Dr.Web for UNIX Mail Serversを登録してライセンスを取得します。ライセンスを取得したら、キーファイルが使用可能かどうかを確認し、必要に応じてインストールします。
   - お使いのOSにてSELinuxを有効化している場合、drweb-se モジュールに対するセキュリティポリシーを設定します（管理者マニュアルの SELinuxのセキュリティポリシーを設定する を参照してください）。
3. 設定にコンポーネントの設定が含まれていない場合、またはこれまでの手順で問題が解決しない場合は、drweb-se パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *FileCheckは使用できません*

エラーコード: x120

内部指定: EC_NO_FILE_CHECK

説明: 脅威の検出に必要なDr.Web File Checkerコンポーネントが見つからないか、起動に失敗しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. drweb-filecheck 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの [FileCheck] セクション にある ExePath パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow FileCheck.ExePath
   ```

新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset FileCheck.ExePath <new path>
```

パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset FileCheck.ExePath -r
```

2. 正しいパスを入力した後もエラーが継続する場合は
   - お使いのOSにてSELinuxを有効化している場合、`drweb-filecheck` モジュールに対するセキュリティポリシーを設定します（管理者マニュアルの SELinuxのセキュリティポリシーを設定する を参照してください）。
3. 設定にコンポーネントの設定が含まれていない場合、またはこれまでの手順で問題が解決しない場合は、`drweb-filecheck` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *ESAgentは使用できません*

エラーコード: `x121`

内部指定: `EC_NO_ESAGENT`

説明: 集中管理サーバーへの接続に必要なDr.Web ES Agentコンポーネントがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb-esagent` 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの `[ESAgent]` セクション にある `ExePath` パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

```
$ drweb-ctl cfshow ESAgent.ExePath
```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

```
# drweb-ctl cfset ESAgent.ExePath <new path>
```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset ESAgent.ExePath -r
```

2. 設定にDr.Web ES Agentコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、`drweb-esagent` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインスト

ールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *Firewallは使用できません*

エラーコード: x122

内部指定: EC_NO_FIREWALL

説明: ネットワーク接続のスキャンに必要なDr.Web Firewall for Linuxコンポーネントがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにあります）。また、コマンド drweb-ctl log を使用することもできます。

エラーの解決

1. drweb-firewall 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの [LinuxFirewall] セクション にある ExePath パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow LinuxFirewall.ExePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset LinuxFirewall.ExePath <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset LinuxFirewall.ExePath -r
   ```

2. 設定にDr.Web Firewall for Linuxコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、drweb-firewall パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *NetCheckerは使用できません*

エラーコード: x123

内部指定: EC_NO_NET_CHECK

説明: ダウンロードしたファイルのスキャンに必要なDr.Web Network Checkerコンポーネントがありません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって /var/log/syslog ファイルまたは /var/log/messages ファイルにありま

す）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb-netcheck` 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの`[Netcheck]`セクションにある `ExePath` パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow Netcheck.ExePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Netcheck.ExePath <new path>
   ```

   パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset Netcheck.ExePath -r
   ```

2. 設定にDr.Web Network Checkerコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、`drweb-netcheck` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *CloudDは使用できません*

エラーコード: `x124`

内部指定: `EC_NO_CLOUDD`

説明: Dr.Web Cloudサービスへの要求に必要なDr.Web CloudDが見つかりません。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. `drweb-cloudd` 実行ファイルへのパスを確認してください。必要に応じてパスを変更します（設定ファイルの`[CloudD]`セクションにある `ExePath` パラメータ）。

   または、コマンドライン管理ツールの コマンド を使用することもできます。

   現在のパラメータ値を確認するには、以下のコマンドを実行します。

   ```
   $ drweb-ctl cfshow CloudD.ExePath
   ```

   新しいパラメータ値を設定するには、以下のコマンドを実行します。

   ```
   # drweb-ctl cfset CloudD.ExePath <new path>
   ```

パラメータ値をデフォルトに復元するには、以下のコマンドを実行します。

```
# drweb-ctl cfset CloudD.ExePath -r
```

2. 設定にDr.Web CloudDコンポーネントの設定が含まれていない場合、または正しいパスを入力した後もエラーが継続する場合は、`drweb-cloudd` パッケージをインストールまたは再インストールしてください。

   Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については Dr.Web for UNIX Mail Serversをインストールする および Dr.Web for UNIX Mail Serversをアンインストールする のセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

エラーメッセージ: *予期せぬエラーです*

エラーコード: `x125`

内部指定: `EC_UNEXPECTED_ERROR`

説明: いずれかのコンポーネントの動作に、予期せぬエラーが発生しました。

考えられる原因やエラーの状況を特定するには、Dr.Web for UNIX Mail Serversのログを参照してください（デフォルトでは、OSによって `/var/log/syslog` ファイルまたは `/var/log/messages` ファイルにあります）。また、コマンド `drweb-ctl log` を使用することもできます。

エラーの解決

1. 以下のコマンドを入力し、Dr.Web for UNIX Mail Serversを再起動します。

   ```
   # service drweb-configd restart
   ```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡のうえ、エラーコードをお伝えください。

コードのないエラー

症状: Dr.Web ClamD、SpIDer Gate、Dr.Web MailDなどのコンポーネントがメッセージをスキャンしない。Dr.Web for UNIX Mail Serversのログにメッセージ`Too many open files`が表示される。

説明: データスキャンの負荷が大きいため、Dr.Web Network Checkerが利用可能なファイル記述子の制限を超えました。

エラーの解決

1. ulimit -nコマンドを使用して、アプリケーションで使用可能なオープンファイルの記述子数の上限を引き上げます（Dr.Web for UNIX Mail Serversの記述子数のデフォルト上限は16384です）。

場合によっては、システムサービスsystemdは、指定した上限の変更を無視できます。

この場合、ファイル/etc/systemd/system/drweb-configd.service.d/limits.confを編集し(存在しない場合は作成し)、変更後の制限値を次のように指定します。

```
[Service]
LimitNOFILE=16384
```

systemdの利用可能な上限のリストはドキュメントman systemd.execで確認できます。

2. 上限を変更したら、次のコマンドを実行してDr.Web for UNIX Mail Serversを再起動します。

```
# service drweb-configd restart
```

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡ください。

症状：WebブラウザがDr.Web管理Webインターフェースへの接続を確立できない。Dr.Webアンチウイルスソリューションのコンポーネントが、実行中のプロセスのリスト(ps ax | grep drweb)に含まれていない。drweb-ctl rawscan以外のdrweb-ctl *<command>*を実行しようとすると、次のいずれかのエラーが発生する。
Error: connect: No such file or directory: "*<path>*/.com.drweb.public"
または
Error: connect: Connection refused: "*<path>*/.com.drweb.public".

説明：設定デーモンDr.Web ConfigDが使用できないため、Dr.Web for UNIX Mail Serversを起動できません。

エラーの解決

1. 次のコマンドを実行します。

```
# service drweb-configd restart
```

これによってDr.Web ConfigDとDr.Web for UNIX Mail Serversが再起動します。

2. このコマンドがエラーメッセージを返した場合、または効果がない場合は、drweb-configdコンポーネント(パッケージ)を個別にインストールしてください。

これは、PAM認証がシステムで使用されていないことを意味する場合もあります。その場合は、PAMをインストールして設定します(PAMがないとDr.Web for UNIX Mail Serversは正しく動作できません)。

3. エラーが解決しない場合は、Dr.Web for UNIX Mail Serversを削除してから再度インストールしてください。

Dr.Web for UNIX Mail Serversまたはそのコンポーネントのインストールとアンインストールの方法については、Dr.Web for UNIX Mail Serversをインストールするおよび Dr.Web for UNIX Mail Serversをアンインストールするのセクションを参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡ください。

症状

1. SpIDer Gateを無効にした後、すべてのネットワーク接続が切断され（SSHおよびFTPプロトコル経由での送信と、場合によっては受信も）、再確立することができない。
2. NetFilter（`iptables`）ルールを検索するために

   ```
   # iptables-save | grep "comment --comment --comment"
   ```

   を使用すると、空以外の結果が返ってくる。

説明：このエラーは、1.4.15より以前のバージョンのNetFilter（`iptables`）の異常動作に関連しています。この内部エラーのため、SpIDer Gateが一意のラベル（コメント）が付いたルールをルールのリストに追加すると、そのルールは誤って追加されます。その結果、SpIDer Gateはシャットダウン時に接続のリダイレクトルールを削除できません。

エラーの解決

1. SpIDer Gateモニターを再度有効にしてください。
2. SpIDer Gateを無効にする必要がある場合は、以下のコマンドを使用してNetFilter（`iptables`）の不正なルールを削除してください。

   ```
   # iptables-save | grep -v "comment --comment --comment" | iptables-restore
   ```

   

   `iptables-save`および`iptables-restore`コマンドにはroot権限が必要です。権限を昇格するには`su`または`sudo`コマンドを使用できます。

   このコマンドは、正しくないコメントを持つすべてのルール（例：同じくトラフィックのリダイレクトを実行する、他のアプリケーションによって追加されたものなど）を削除します。

追加情報

- この問題の発生を防ぐため、お使いのOSをアップグレードすることを推奨します（または、少なくともNetFilterをバージョン1.4.15以降にすることを推奨します）。
- また、`iptables`ユーティリティを使用して必要なルールを指定することで、接続を手動でSpIDer Gateへリダイレクトさせる場合は、Dr.Web Firewallの設定でSpIDer Gateへの接続のリダイレクトを手動モードに切り替えることができます（この方法は推奨されません）。
- 詳細は、マニュアル`man: drweb-firewall(1)`、`drweb-gated(1)`、`iptables(8)`を参照してください。

引き続きエラーが発生する場合は、テクニカルサポートまでご連絡ください。

付録G. 略語のリスト

本マニュアルでは、次の略語を詳しく解説することなく使用しています。

表記規則	説明
AD	Microsoft Active Directory（マイクロソフトアクティブディレクトリ）
DN	（LDAP）Distinguished Name（識別名）
FQDN	Fully Qualified Domain Name（完全修飾ドメイン名）
GID	Group ID（システムユーザーグループID）
GNU	GNUプロジェクト（GNU is Not Unix）
HTML	HyperText Markup Language（ハイパーテキストマークアップ言語）
HTTP	HyperText Transfer Protocol（ハイパーテキスト転送プロトコル）
HTTPS	HyperText Transfer Protocol Secure（ハイパーテキスト転送プロトコルセキュア）（SSL/TLS経由）
ID	Identifier（識別子）
IMAP	Internet Message Access Protocol（インターネットメッセージアクセスプロトコル）（メールプロトコル）
IP	Internet Protocol（インターネットプロトコル）
LDAP	Lightweight Directory Access Protocol（ライトウェイトディレクトリアクセスプロトコル）
MBR	Master Boot Record（マスターブートレコード）
MDA	Mail Delivery Agent（メール配信エージェント）
MTA	Mail Transfer Agent（メール転送エージェント）（メールサーバー）
MUA	Mail User Agent（メールユーザーエージェント）（メールクライアント）
OID	（SNMP）Object ID（オブジェクトID）
OS	Operating System（オペレーティングシステム）
PID	Process ID（システムプロセスID）
PAM	Pluggable Authentication Modules（プラガブル認証モジュール）
POP	Post Office Protocol（ポストオフィスプロトコル）（メールプロトコル）
RPM	Red Hat Package Manager（Red Hatパッケージマネージャー）

表記規則	説明
RRA	Round-Robin Archive（ラウンドロビンアーカイブ）
RRD	Round-Robin Database（ラウンドロビンデータベース）
SMTP	Simple Mail Transfer Protocol（シンプルメールトランスファープロトコル）（メールプロトコル）
SNI	Server Name Indication（サーバー名表示）
SNMP	Simple Network Management Protocol（シンプルネットワークマネージメントプロトコル）
SP	Service Pack（サービスパック）
SSH	Secure Shell（セキュアシェル）
SSL	Secure Sockets Layer（セキュアソケットレイヤー）
TCP	Transmission Control Protocol（伝送制御プロトコル）
TLS	Transport Layer Security（トランスポート層セキュリティ）
UID	User ID（システムユーザーID）
URI	Uniform Resource Identifier（ユニフォームリソースアイデンティファイア）
URL	Uniform Resource Locator（ユニフォームリソースロケータ）
VBR	Volume Boot Record（ボリュームブートレコード）