Dr.WEB

vxCube

User Manual

Trademarks

Dr.Web, SpIDer Mail, SpIDer Guard, CureIt!, CureNet!, AV-Desk, KATANA and the Dr.WEB logo are trademarks and registered trademarks of Doctor Web in Russia and/or other countries. Other trademarks, registered trademarks and company names used in this document are the property of their respective owners.

Disclaimer

In no event shall Doctor Web and its resellers or distributors be liable for any errors or omissions, or for any loss of profit or any other damage caused or alleged to be caused directly or indirectly by this document, or by the use of or inability to use the information contained in this document.

Dr.Web vxCube
Version 1.6.0
User Manual
12/5/2024

Doctor Web Head Office

2-12A, 3rd str. Yamskogo polya, Moscow, Russia, 125124

Website: https://www.drweb.com/

Phone: +7 (495) 789-45-87

Refer to the official website for regional and international office information.

Doctor Web

Doctor Web develops and distributes Dr.Web information security solutions that provide effective protection against malicious software and spam.

Doctor Web customers include home users around the world, government agencies, small businesses, and nationwide corporations.

Since 1992, Dr.Web anti-virus solutions have been known for their continuous excellence in malware detection and compliance with international information security standards.

The state certificates and awards received by Dr.Web solutions, as well as the worldwide use of our products, are the best evidence of exceptional trust in the company products.

We thank all our customers for their support and devotion to Dr.Web products!

Table of Contents

1. Conventions 6

2. About Product 7

2.1. What is Special About Dr.Web vxCube? 7

2.2. How to Use Dr.Web vxCube 7

2.3. System Requirements 7

3. Signing in and out of Dr.Web vxCube 9

4. Settings 10

4.1. Changing an Interface Language 10

4.2. Default Analysis Settings 10

4.3. Managing a Password 11

4.3.1. How to Change a Password 11

4.3.2. How to Reset a Password 11

5. YARA Rules 14

5.1. How to Create a YARA Rule 14

5.2. How to Manage YARA Rules 15

5.3. The Rule Matches 17

5.4. The dr_sandbox Module 18

6. Analyzing Files 19

6.1. Supported Formats 20

6.2. Uploading Files to be Analyzed 22

6.3. Additional Settings 24

7. Reports 27

7.1. Opening a Report 27

7.2. Downloading a Report 27

7.3. The Retention Period of Reports 27

7.4. The Report Structure 28

7.4.1. General Information 29

7.4.2. Main Part 31

7.5. History 46

7.6. Tags 47

8. API 49

8.1. Authentication 49

8.2. Managing API Keys 49

8.3. Endpoints 50

8.3.1. analyses 50

8.3.2. formats 54

8.3.3. login 54

8.3.4. platforms 55

8.3.5. samples 56

8.3.6. sessions 57

8.3.7. tasks 58

8.3.8. ws/progress 62

8.4. Objects 63

8.4.1. Analysis 63

8.4.2. APIEvent 66

8.4.3. Call (optional) 67

8.4.4. Connection 67

8.4.5. Dump 68

8.4.6. Drop 68

8.4.7. Format 69

8.4.8. Intent (optional) 70

8.4.9. Message (optional) 70

8.4.10. Platform 71

8.4.11. Sample 71

8.4.12. Session 72

8.4.13. Task 73

8.5. Examples 76

8.5.1. Get an API Key 77

8.5.2. Upload a Sample to the Dr.Web vxCube Server 77

8.5.3. Start the Analysis 78

8.5.4. Get Information About the Analysis 79

8.5.5. Download a Report 80

9. Technical Support 81

10. Appendix A. List of Software on Virtual Machines 82

11. Appendix B. Functions of the dr_sandbox Module 88

1. Conventions

The following symbols and text conventions are used in this guide:

Convention	Comment
⚠	A warning about possible errors or important notes that require special attention.
Anti-virus network	A new term or an emphasis on a term in descriptions.
<IP-address>	Placeholders.
Save	Names of buttons, windows, menu items and other program interface elements.
CTRL	Names of keyboard keys.
`C:\Windows\`	Names of files and folders, code examples.
Appendix A	Cross-references to document chapters or internal hyperlinks to webpages.

2. About Product

Dr.Web vxCube is a service that analyzes potentially malicious files and generates detailed reports on their behavior in the selected environment.

Dr.Web vxCube uses *hardware virtualization* for the analysis. It allows Dr.Web vxCube to work fast and be invisible to the file you are running.

You can upload a file in a supported format to the analyzer, configure environment on a virtual machine, and influence the analysis process. After the analysis, you will receive a full technical report, as well as a video report showing the file’s behavior in the specified conditions.

2.1. What is Special About Dr.Web vxCube?

Below the specific features of Dr.Web vxCube are listed:

- Virtual machines connect to the internet through a private proxy server. This helps fully analyze the file behavior, especially if it depends on downloading data from the internet.
- The new analyzer works at the *hypervisor* level and does not use any additional software on the host operating system, such as drivers that hook functions. Thus, during the analysis, the sample cannot detect hooks or unhook.
- Events are logged at the hypervisor level—thus, the analyzer cannot be detected.
- It is possible to connect to an analyzed environment via VNC client and influence the analysis process.

2.2. How to Use Dr.Web vxCube

To scan the suspicious file for threats using Dr.Web vxCube, do the following:

1. Upload a file to be scanned to Dr.Web vxCube.
2. (Optional) Specify additional settings and start the analysis.
3. Review the report that Dr.Web vxCube produced based on the analysis results.

2.3. System Requirements

For the intended performance of web UI, we recommend that you use:

Parameter	**Requirement**
Browser	• Google Chrome 60.0 or later. • Mozilla Firefox 55.0 or later. • Safari 11.0 or later.

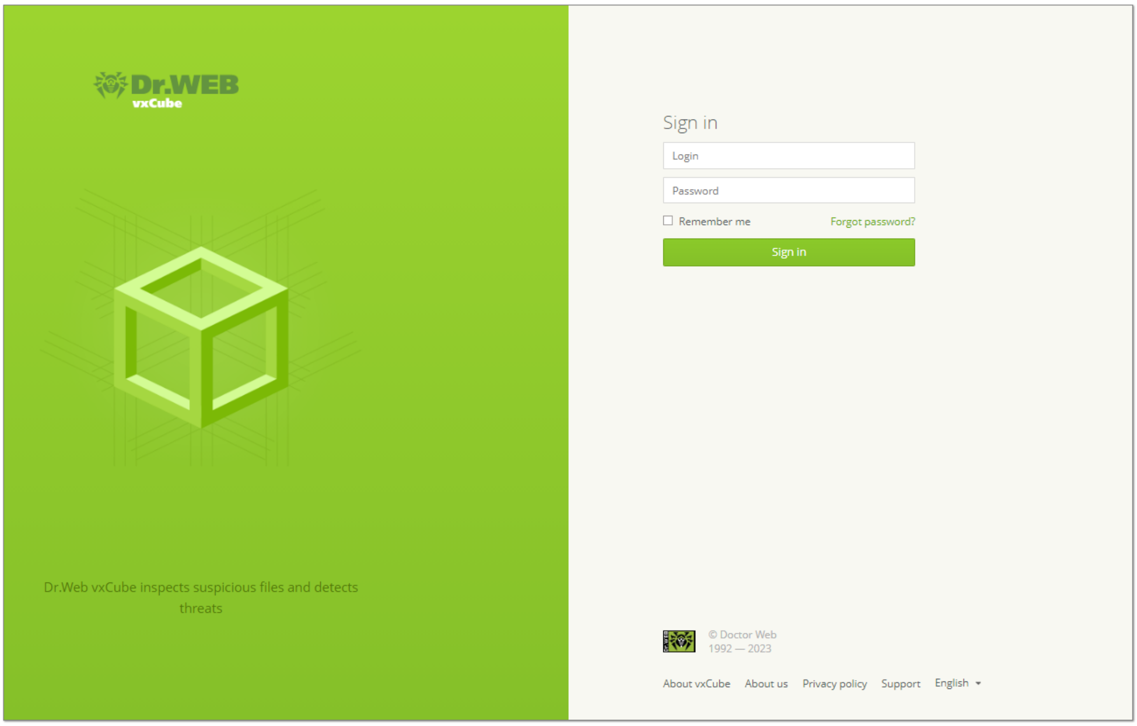
Parameter	Requirement
	• Opera 47.0 or later. We recommend that you use Google Chrome for Windows XP. We cannot guarantee that Mozilla Firefox will correctly display video on computers that run Windows XP.
Screen resolution	At least 1024x768.
Optional	If you want to manage emulation interactively, make sure that pop-ups are allowed in your browser.

3. Signing in and out of Dr.Web vxCube

Signing in to Dr.Web vxCube

Before you start working with Dr.Web vxCube, make sure that your computer meets the system requirements.

To start using Dr.Web vxCube, go to `http(s)://your address` and enter your login and password received from the service administrator. The first time you sign in, you will be prompted to accept the License Agreement.



Dr.Web vxCube inspects suspicious files and detects threats

© Doctor Web 1992 — 2023

Figure 1. Dr.Web vxCube sign-in page

Signing out of your Dr.Web vxCube account

To sign out of your Dr.Web vxCube account, in the top right-hand corner of the main screen click **Profile > Sign out**.

4. Settings

You can change an interface language of Dr.Web vxCube (currently supports Russian and English), set default file analysis settings, and manage your API keys and password.

4.1. Changing an Interface Language

Dr.Web vxCube is available in English and Russian. By default, Dr.Web vxCube display language matches the language preferences of your browser.

To change your display language

1. Scroll down the page.
2. Click the language selection menu at the bottom of the page.
3. Select English or Russian in the drop-down.

4.2. Default Analysis Settings

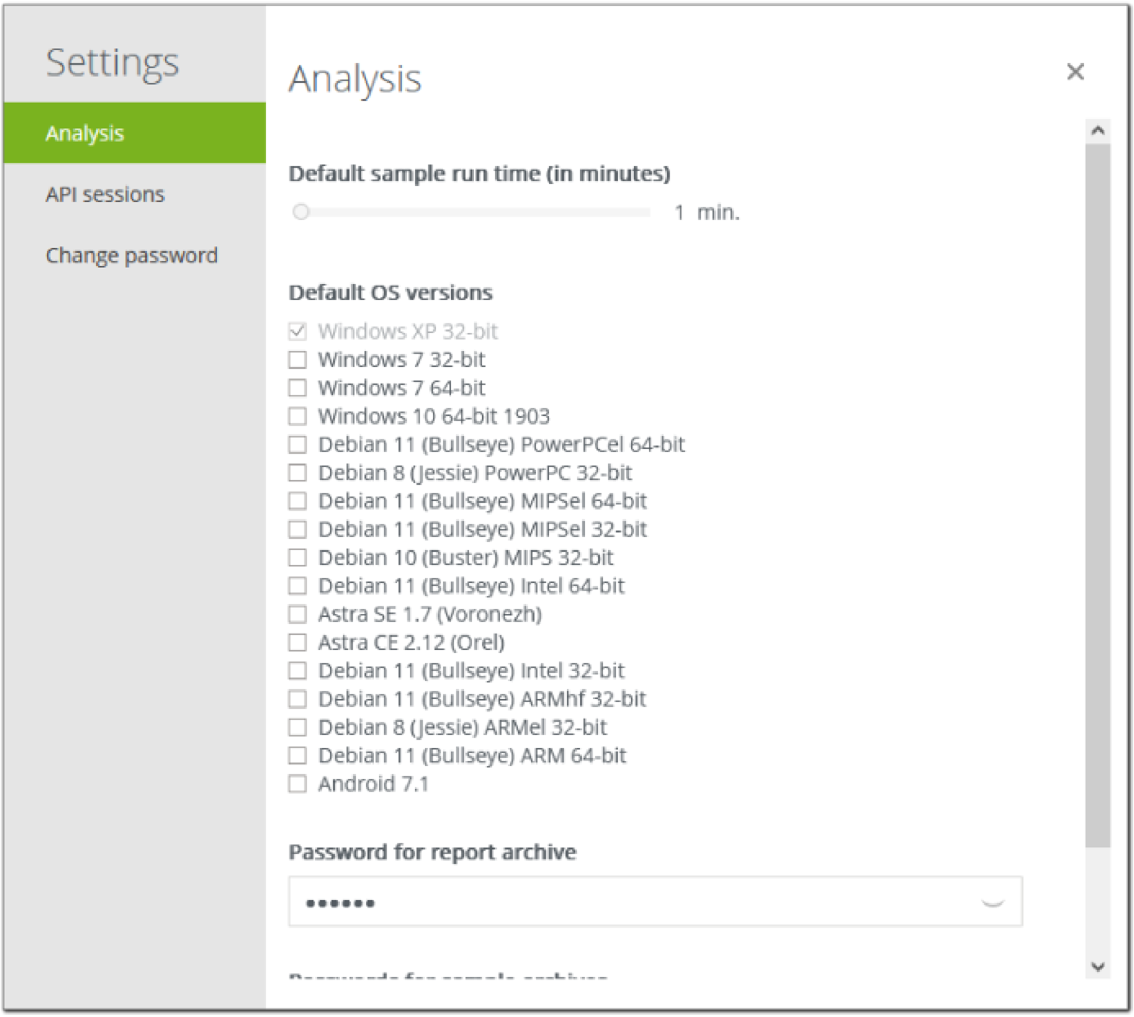
You can specify the default analysis settings such as sample run time on the virtual machine, OS versions to analyze on and the optional password for the report archive (if no password is set, the archive is sent without it).

In the **Passwords for sample archives** field, you can input passwords that will be used to analyze a password-protected archive.

> ⚠ In case a report archive is not password-protected, your local machine’s anti-virus can scan it and potentially detect it as malware, particularly if the report includes alloc function dumps.

To specify the default analysis settings

1. In the top right-hand corner of the main page, click 👤 **Profile > Settings**.
2. On the left, select the **Analysis** tab.
3. Specify the default settings for file analysis.



Analysis

Default sample run time (in minutes)

1 min.

Default OS versions

☑ Windows XP 32-bit
□ Windows 7 32-bit
□ Windows 7 64-bit
□ Windows 10 64-bit 1903
□ Debian 11 (Bullseye) PowerPCel 64-bit
□ Debian 8 (Jessie) PowerPC 32-bit
□ Debian 11 (Bullseye) MIPSel 64-bit
□ Debian 11 (Bullseye) MIPSel 32-bit
□ Debian 10 (Buster) MIPS 32-bit
□ Debian 11 (Bullseye) Intel 64-bit
□ Astra SE 1.7 (Voronezh)
□ Astra CE 2.12 (Orel)
□ Debian 11 (Bullseye) Intel 32-bit
□ Debian 11 (Bullseye) ARMhf 32-bit
□ Debian 8 (Jessie) ARMel 32-bit
□ Debian 11 (Bullseye) ARM 64-bit
□ Android 7.1

Password for report archive

••••••

Passwords for sample archives

Figure 2. Settings

4.3. Managing a Password

If you have forgotten your Dr.Web vxCube password, you can reset it. You can also change your password if it has been compromised.

4.3.1. How to Change a Password

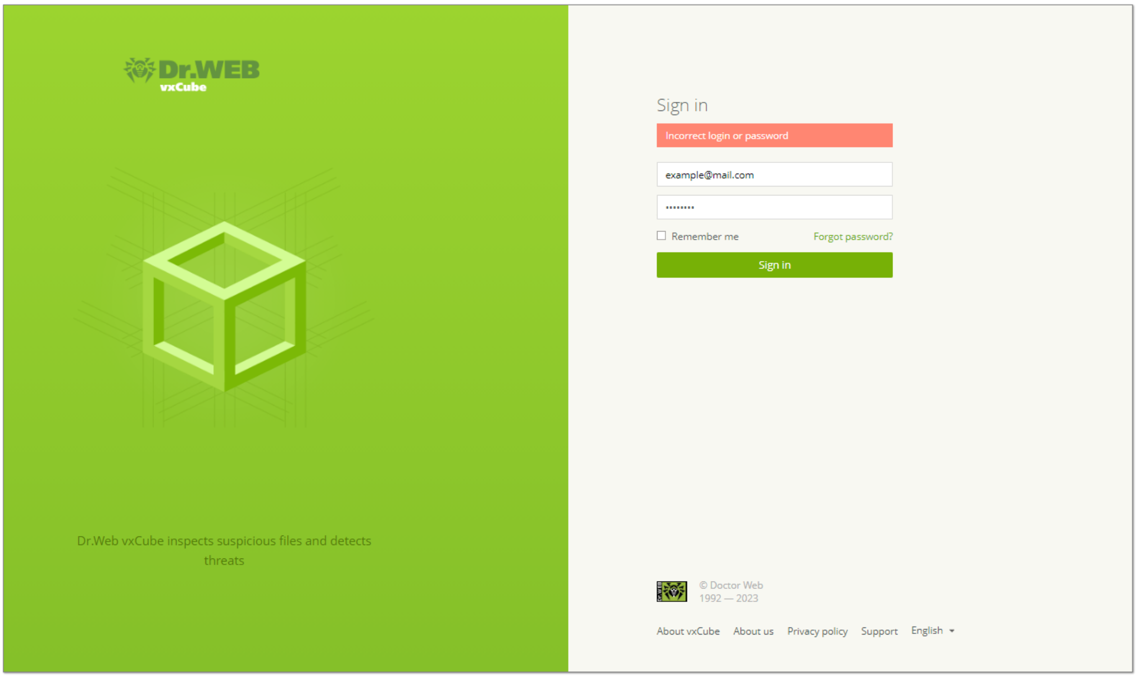
To change your password

1. In the top right-hand corner of the main page, click **Profile > Settings**.
2. On the left, select the **Change password** tab.
3. Enter your current password, then enter the new one twice and click **Save**.

4.3.2. How to Reset a Password

To reset your password

1. On the Dr.Web vxCube login page, click **Forgot password?**.



Sign in

Incorrect login or password

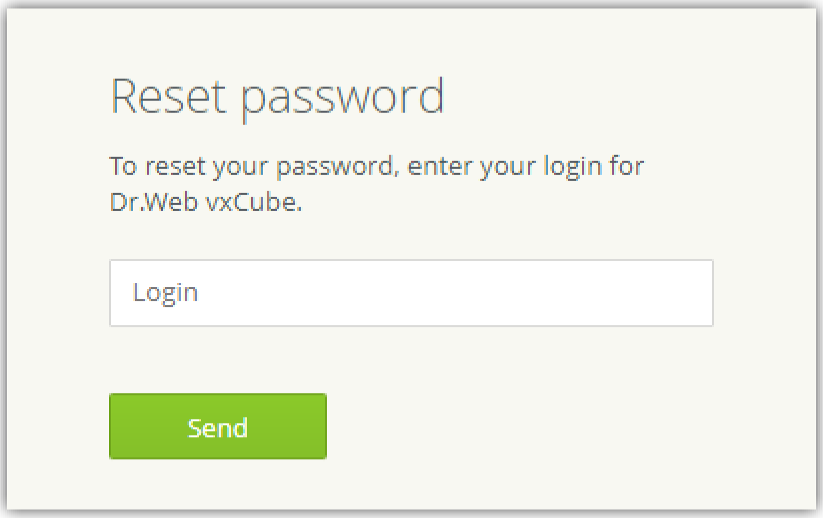
example@mail.com

••••••••

Figure 3. Failed to sign in to Dr.Web vxCube

2. On the **Reset password** page specify email address you used for registering in Dr.Web vxCube.
3. Select the **I'm not a robot** check box. It is only required if the CAPTCHA was enabled during the installation of Dr.Web vxCube.
4. Click **Send**.

   On this address, you will receive an email with a link for resetting your password. If you do not receive the email within 10 minutes, check the Spam folder or contact the server administrator.



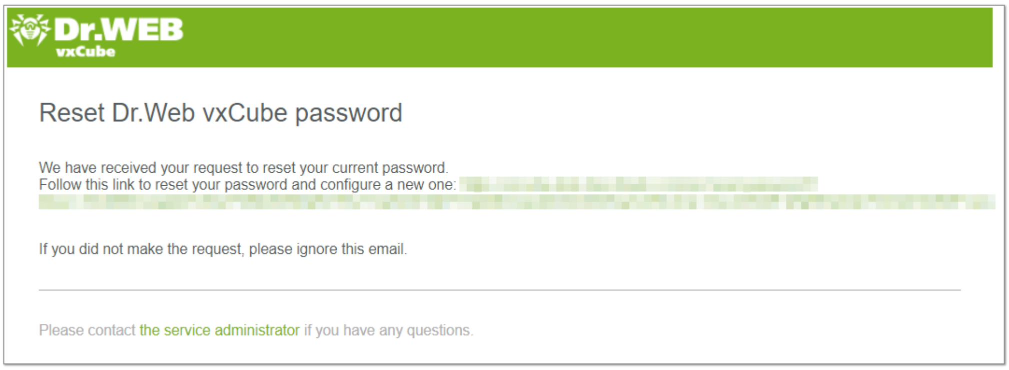
Reset password

To reset your password, enter your login for Dr.Web vxCube.

Figure 4. Requesting a password reset

5. Open the email you have received.
6. Follow the link to reset your password.

You will be redirected to Dr.Web vxCube.



Reset Dr.Web vxCube password

We have received your request to reset your current password.
Follow this link to reset your password and configure a new one: [redacted]

If you did not make the request, please ignore this email.

Please contact the service administrator if you have any questions.

Figure 5. Confirming the request to reset your password

7. Type in your new password and confirm it.
8. Click **Create**.

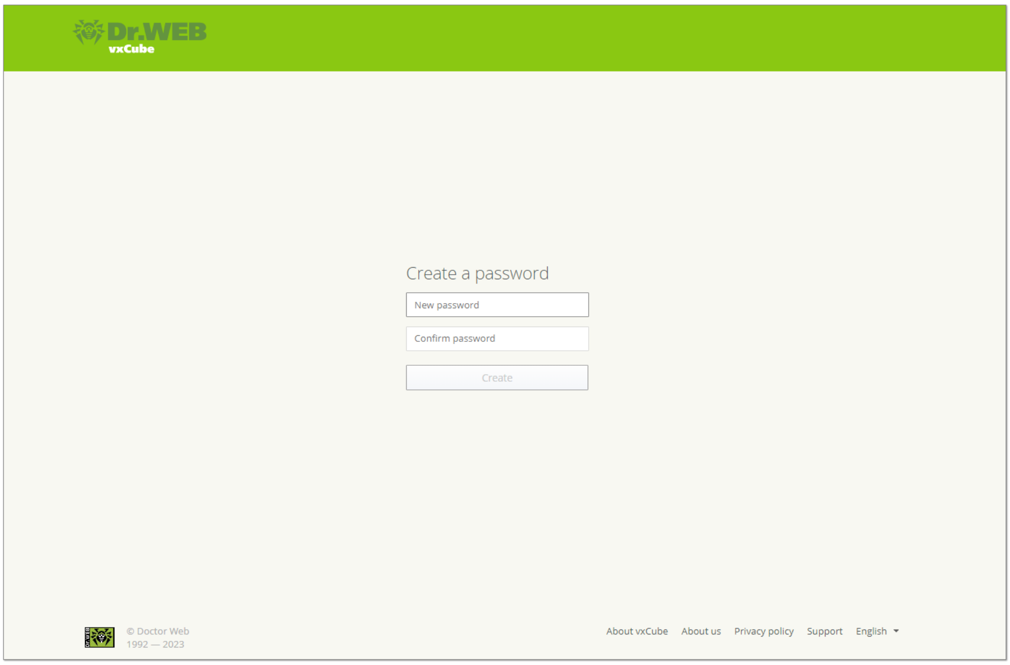


Figure 6. Creating a password

5. YARA Rules

Using YARA rules you can identify and classify malware samples: a rule triggers when the condition within it is met. The condition can refer to the specific file contents, behavior, or location. YARA rules can include strings, boolean expressions, wildcards, regular expressions, special operators, and many other features. For more information about YARA rules, go to official YARA documentation.

YARA rules used in Dr.Web vxCube have some special capabilities:

- In the `meta` rule section, the required `maliciousness` field is added. This field is used to specify a maliciousness type that will be added to the report if the rule triggers
- Using the exclusive dr_sandbox module, you can create rules triggered when specific behavior for a file is detected on a virtual machine.

All the YARA rules in Dr.Web vxCube are divided into two categories: *system* rules and *user* rules. System rules are created by the Dr.Web vxCube developers and used in file analysis by default. You can’t view or edit the contents of system rules, as well as delete them, but you have the option to disable these rules that are not needed. Additionally, you can create your own (user) rules. User rules can be edited, disabled, or deleted.

5.1. How to Create a YARA Rule

All the YARA rules in Dr.Web vxCube follow the standard format:

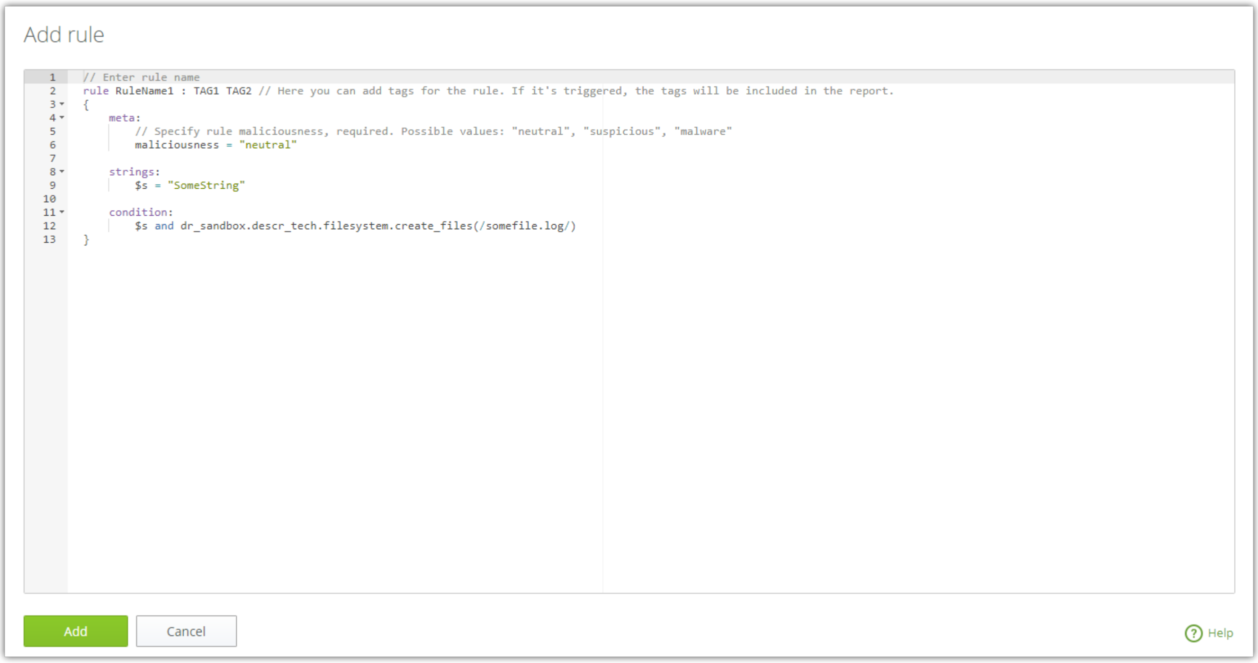
```
rule RuleName1 : TAG1 TAG2
{
    meta:
        maliciousness = "neutral"

    strings:
        $s = "SomeString"

    condition:
        $s
}
```

Every rule begins with the keyword `rule` followed by a rule name that should be entered using latin letters, digits, or underscore. Then, after a colon, you could specify tags. They will be included in the report if this rule is triggered during the file analysis. The rule body can contain three sections:

- The required `meta` section specifies the maliciousness type (the `maliciousness` field) that will be set for the file if the rule is triggered. The possible values for the field: `maliciousness`: `neutral`, `suspicious`, `malware`.
- In the required `condition` section, a condition is set. If the condition is met, the rule will be triggered.
- In the optional `strings` section, the strings that used in the rule are specified.

To create a YARA rule

1. At the top of the Dr.Web vxCube main page, click **YARA rules**.
2. Click ⊕ **Add**. The window containing a rule example code appears.
3. Edit the code to include the rule options you want.
4. Click **Add**.



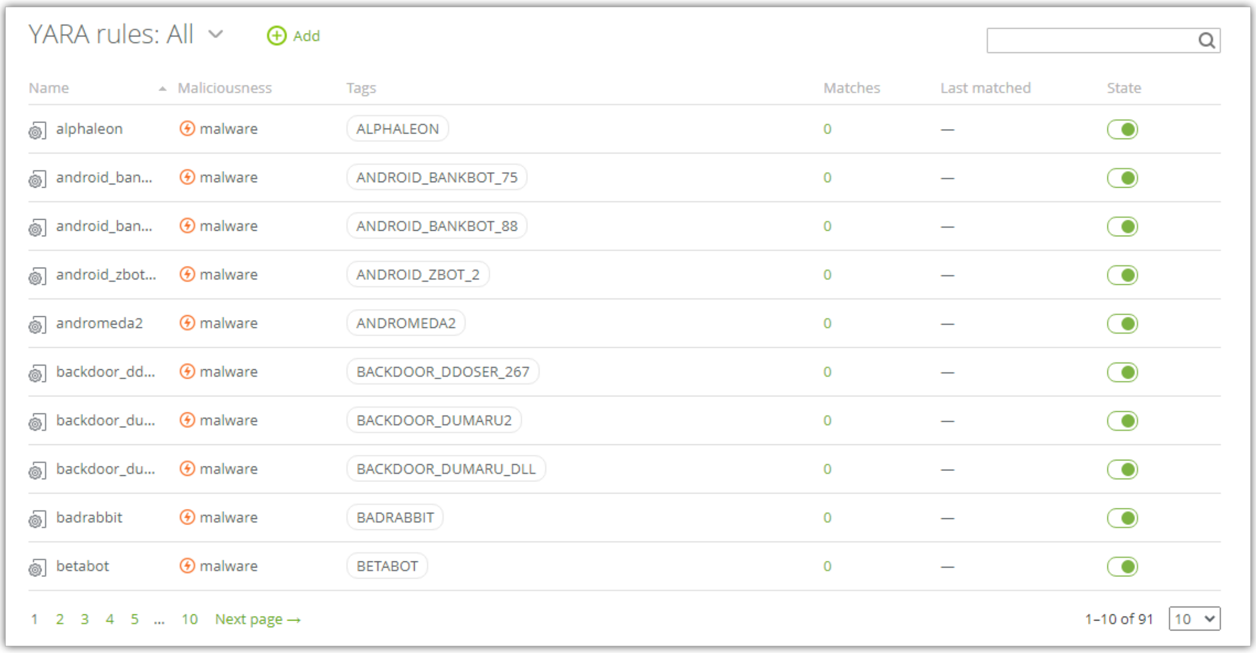
```
// Enter rule name
rule RuleName1 : TAG1 TAG2 // Here you can add tags for the rule. If it's triggered, the tags will be included in the report.
{
    meta:
        // Specify rule maliciousness, required. Possible values: "neutral", "suspicious", "malware"
        maliciousness = "neutral"

    strings:
        $s = "SomeString"

    condition:
        $s and dr_sandbox.descr_tech.filesystem.create_files(/somefile.log/)
}
```

Figure 7. Add rule window

5.2. How to Manage YARA Rules

Click **YARA rules** at the top of the Dr.Web vxCube main page to see all YARA rules available for your account. The YARA rule list that opens includes the following information for each rule:

- The rule type (👤 for user rules and ⚙ for system rules).
- **Name:** The rule name.
- **Maliciousness:** The maliciousness level specified in the rule.
- **Tags:** Tags specified in the rule.
- **Matches:** The total amount of matches for the particular rule.
- **Last matched:** The date when the rule was last triggered. If the trigger occurred today, the time will be shown instead of the date.
- **State:** The current state of the rule (enabled/disabled).



Name ▲	Maliciousness	Tags	Matches	Last matched	State
alphaleon	malware	ALPHALEON	0	—	●
android_ban...	malware	ANDROID_BANKBOT_75	0	—	●
android_ban...	malware	ANDROID_BANKBOT_88	0	—	●
android_zbot...	malware	ANDROID_ZBOT_2	0	—	●
andromeda2	malware	ANDROMEDA2	0	—	●
backdoor_dd...	malware	BACKDOOR_DDOSER_267	0	—	●
backdoor_du...	malware	BACKDOOR_DUMARU2	0	—	●
backdoor_du...	malware	BACKDOOR_DUMARU_DLL	0	—	●
badrabbit	malware	BADRABBIT	0	—	●
betabot	malware	BETABOT	0	—	●

Figure 8. The list of YARA rules

In the list of YARA rules, you can:

- Search for rules by their names and tags
- Filter rules by type (system/user)
- Sort rules
- View information about rule matches (the name of the file that the rule was triggered on, the date of triggering, OS)
- Edit, delete, and enable/disable rules

To search for a rule

- To find specific rule(s), type their name or tags (or a portion of them) in the search box located at the top right of the rule list.

To filter rules by type

- Next to the header of the rule list, click ⌄ and choose the filter option: **YARA rules: All**, **YARA rules: System**, or **YARA rules: User**.

To sort rules

- Click the header of the column you want to sort by. At the left of the header ▲ or ▼ will appear. To change the sorting direction, click the header again.

To view information about the rule matches

- In the **Matches** column, click the number of matches for the required rule. The page of reports on matches for this rule opens.

To edit a rule

- Hover over the row of the rule and click ✎ on the right.

To delete a rule

- Hover over the row of the rule and click 🗑 on the right.

To disable or enable a rule

- In the row of the rule, turn the 🟩 switcher on or off.

To set the number of rules displayed per page

- At the bottom right, select the required value (10, 25, 50, or 100) from the drop-down.

5.3. The Rule Matches

You can view information about all the matches of the particular YARA rule. To do this:

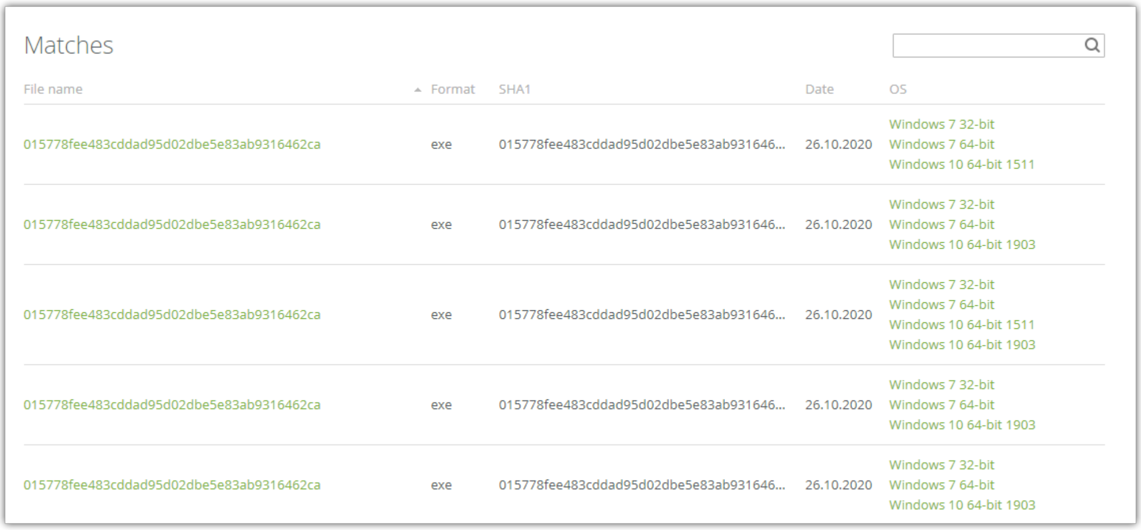
1. At the top of the Dr.Web vxCube main page, click **YARA rules**.
2. In the **Matches** column for the required rule, click the number.

The full list of matches for this rule opens. For each match, the following information is displayed:

- **File name:** The name of the file that the match occurred on.
- **Format:** The format of the file that the match occurred on.
- **SHA1:** The hash of the file.
- **Date:** The date when the match was occurred.
- **OS:** The list of operating systems that the analysis has been done for.

From the rule match report, you can go to the analysis report related to the particular match:

- To go to the report main page, click the file name in the corresponding row.
- To go to the report page for the specific platform, click the OS name in the corresponding row.



Matches

File name	Format	SHA1	Date	OS
015778fee483cddad95d02dbe5e83ab9316462ca	exe	015778fee483cddad95d02dbe5e83ab931646...	26.10.2020	Windows 7 32-bit Windows 7 64-bit Windows 10 64-bit 1511
015778fee483cddad95d02dbe5e83ab9316462ca	exe	015778fee483cddad95d02dbe5e83ab931646...	26.10.2020	Windows 7 32-bit Windows 7 64-bit Windows 10 64-bit 1903
015778fee483cddad95d02dbe5e83ab9316462ca	exe	015778fee483cddad95d02dbe5e83ab931646...	26.10.2020	Windows 7 32-bit Windows 7 64-bit Windows 10 64-bit 1511 Windows 10 64-bit 1903
015778fee483cddad95d02dbe5e83ab9316462ca	exe	015778fee483cddad95d02dbe5e83ab931646...	26.10.2020	Windows 7 32-bit Windows 7 64-bit Windows 10 64-bit 1903
015778fee483cddad95d02dbe5e83ab9316462ca	exe	015778fee483cddad95d02dbe5e83ab931646...	26.10.2020	Windows 7 32-bit Windows 7 64-bit Windows 10 64-bit 1903

Figure 9. Report on YARA rule matches

5.4. The dr_sandbox Module

The dr_sandbox module is an exclusive YARA module of Doctor Web. With dr_sandbox, you can create rules based on the following information:

- File behavior on a virtual machine
- Types of created files (`src`, `dump`, `drop`, `alloc` etc.)
- Details regarding detected threats
- The name of the analyzed file

The rule example that includes the `connect_to` function of dr_sandbox:

```
rule bad_file
{
    condition:
        dr_sandbox.descr_tech.network.connect_to(/http:\/\/someplace\.badsite\.com/)
}
```

You can find the full list of the dr_sandbox module functions in Appendix B. Functions of dr_sandbox module.

6. Analyzing Files

To analyze a file

1. Make sure Dr.Web vxCube supports the format of the file you want to analyze.
2. Browse for the file you want to check and upload it to the application.

   If Dr.Web vxCube cannot identify the file format automatically, you will be able to select it manually.
3. Select an environment for the analysis—an operating system version or an application version.

   You can select multiple OS versions or application versions.
4. (Optionally) Specify additional settings for analyzing the file.
5. Click **Analyze**.

   > ⚠ Files can be also analyzed using API.

Analysis

When you start the analysis, one or several virtual machines with pre-installed software will be run. The number of virtual machines depends on the number of OS versions or application versions you have selected.

All events related to file behavior on a virtual machine are monitored to detect any suspicious activity. All processes on a guest OS are logged to the API Log. The analyzer uses a list of rules to categorize these processes.

The Dr.Web vxCube analyzer interacts with a *hypervisor* and does not use any additional software in the guest operating system (for example, drivers that hook functions). Thus, during analysis, the sample cannot detect or remove hooks.

Virtual machines connect to the internet through a dedicated proxy server. This helps fully analyze the virus behavior, especially if its functioning depends on downloading data from the internet.

In order to log events, Dr.Web vxCube interacts with a hypervisor, not with virtual machines. It means the analyzer cannot be detected.

You can connect to a virtual machine through a VNC (Virtual Network Computing) client and influence the analysis. Note that this can only be done when the virtual machine is operating.

Once the analysis is complete, you will receive a detailed report and be able to review the history of previously analyzed files.

Sometimes analysis of the same file may have different results if the file behavior depends on external conditions, for example, current date or availability of remote resources.

Additionally, results of analysis using VNC may differ from those obtained without VNC if the analyzed file uses an injection method unknown to Dr.Web vxCube, or the control is transferred to processes indirectly.

6.1. Supported Formats

Dr.Web vxCube supports the following formats:

File type	File format
Windows executable files	CPL, DLL, EXE, MSI, NATIVE APP, SYS
Android packages	APK
Microsoft Office documents	MHT, RTF, DOC, DOCX, DOCM, DOTM, DOTX, WPS, XLL, XLS, XLSX, XLSM, XLSB, XLAM, XTLX, XTLM, SLK, IQY, PPT, PPTX, PPTM, PPSX, PPSM, SLDX, SLDM, PPA, PPAM, THMX, POTX, POTM, XML, ACCDB, PUB, ODT, ODS, ODP
Acrobat Reader files	PDF
Java executable files	CLASS, JAR
Script files	BAT, JS, JSE, PL, PS1, PY, SCT, SH, VBE, VBS, WSF, XSL
*nix executable files	ELF
Other	7Z, ACE, ARJ, BZ2, CAB, CHM, DOCKER, EML, GZ, HTA, LNK, MOF, RAR, TAR, XZ, ZIP

Files with the ZIP, ARJ, XZ, ACE, TAR, BZ2, CAB, GZ, RAR, 7Z, or EML extensions can only be uploaded for analysis using API.

The file size cannot exceed 1000 MB.

File processing

For different formats, Dr.Web vxCube uses different ways of file processing and running.

If you choose a Microsoft Office, Acrobat Reader, or Java file for analysis, you will be prompted to select an app version to run the file instead of an OS version. For example, for a PDF file, you will need to choose between three versions of Acrobat Reader: 10.1, 11.0, or 15.10.

File formats and methods to launch them

File format	Launching
EXE	*%sample%*
DLL	regsvr32 /s *%sample%*
CPL	rundll32 shell32.dll, Control_RunDLL "*%sample%*"
SYS	sc create *%random_name%* type= kernel start= demand error= ignore binpath= "*%sample%*" DisplayName= *%random_name%* sc start *%random_name%*
NATIVE APP	rtlrun *%sample%*
MSI	msiexec.exe /i *%sample%*
MHT	winword *%sample%*
XML	msoxmled.exe
RTF, DOC, DOCX, DOCM, DOTM, DOTX, WPS, ODT	winword.exe
XLS, XLSX, XLSM, XLSB, XLAM, XTLX, XTLM, SLK, IQY, ODS	excel.exe
PPT, PPTX, PPTM, PPSX, PPSM, SLDX, SLDM, PPA, PPAM, THMX, POTX, POTM, ODP	powerpnt.exe
ACCDB	msaccess.exe
PUB	mspub.exe
PDF	acrord32.exe
JAR	javaw -jar *%sample%*
CLASS	java *%sample%*
JS, VBS, WSF, JSE, VBE	wscript /b /nologo *%sample%*
PS1	powershell -file *%sample%*

File format	Launching
BAT	cmd /c *%sample%*
SCT	regsvr32.exe /s /i:*%sample%* scrobj.dll
XSL	wmic printjob get /format:"*%sample%*"
MOF	mofcomp *%sample%*
LNK, HTA	*%sample%*
CHM	hh.exe
XLL	excel.exe *%sample%*
ELF	*%sample%*
SH	bash *%sample%*
PY	python *%sample%*
PL	perl *%sample%*
DOCKER	docker load -i *%sample%* docker run *%image_id%*

%sample% is the name of the analyzed file on a virtual machine.

%random_name% is a randomly given name.

6.2. Uploading Files to be Analyzed

To upload a file for analysis

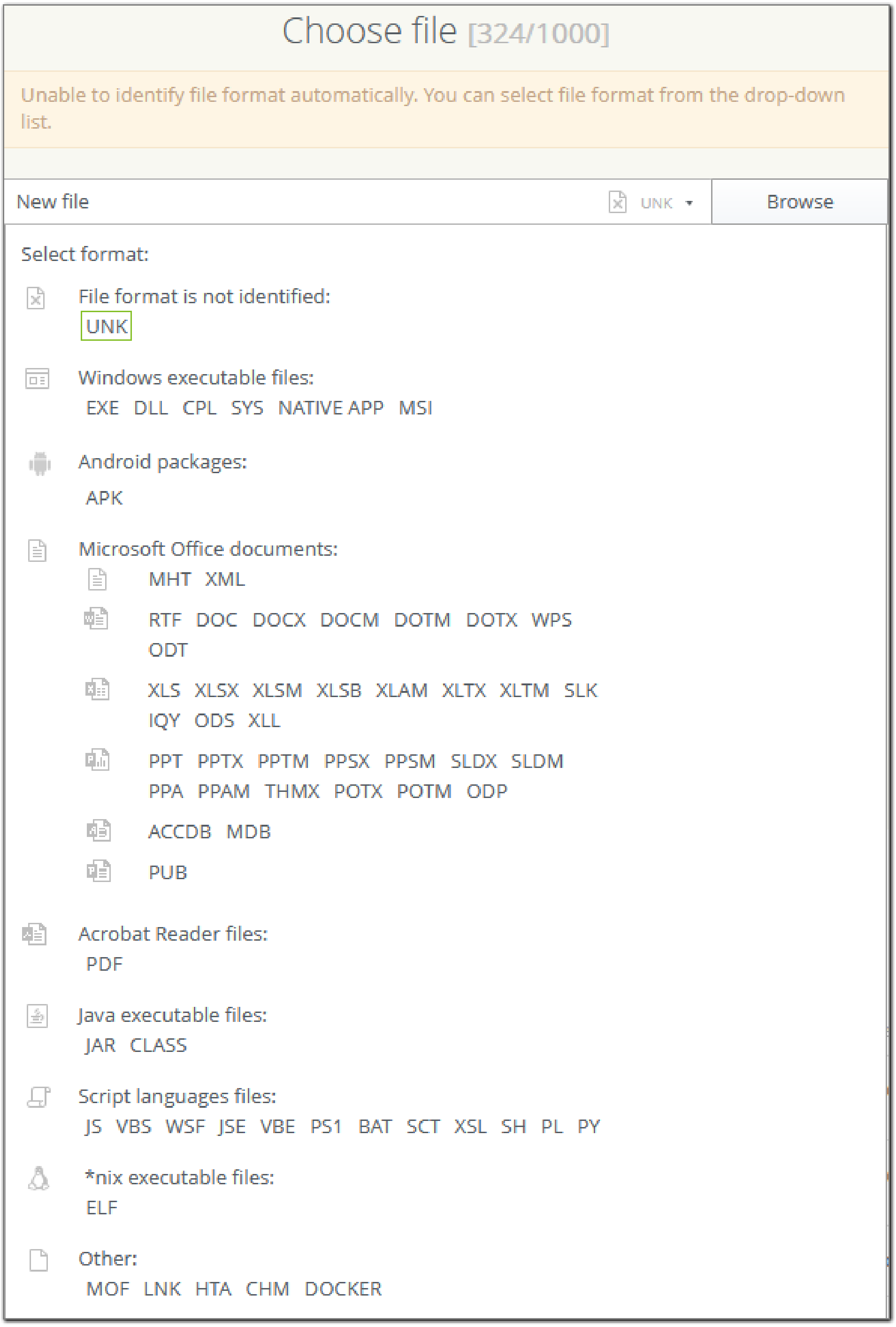
1. On the Dr.Web vxCube main page, click the **Browse** button or the file-select field. Select a file you want to analyze.

   You can also drag a file into the file-select field.

   The uploaded file format is detected automatically by its content.

   If the format is not identified (UNK), you will see the **Unable to identify file format** message. In this case, you can select the file format manually.

   ⚠ The MOF, JS, VBS, WSF, JSE, VBE, PS1, and BAT file formats may be identified incorrectly. For these files, it’s recommended that you select format manually.



Choose file [324/1000]

Unable to identify file format automatically. You can select file format from the drop-down list.

New file UNK ▾ Browse

Select format:

File format is not identified:
UNK

Windows executable files:
EXE DLL CPL SYS NATIVE APP MSI

Android packages:
APK

Microsoft Office documents:
MHT XML
RTF DOC DOCX DOCM DOTM DOTX WPS ODT
XLS XLSX XLSM XLSB XLAM XLTX XLTM SLK IQY ODS XLL
PPT PPTX PPTM PPSX PPSM SLDX SLDM PPA PPAM THMX POTX POTM ODP
ACCDB MDB
PUB

Acrobat Reader files:
PDF

Java executable files:
JAR CLASS

Script languages files:
JS VBS WSF JSE VBE PS1 BAT SCT XSL SH PL PY

*nix executable files:
ELF

Other:
MOF LNK HTA CHM DOCKER

Figure 10. Selecting file format manually

To select a file format manually, click drop-down arrow and select the corresponding format.

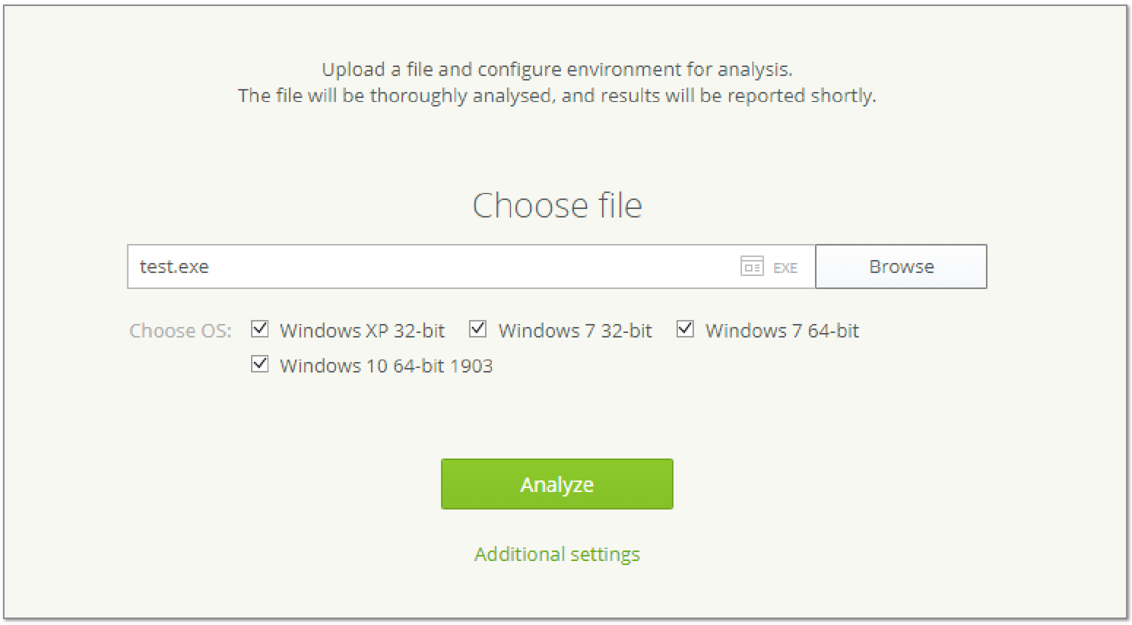
Make sure you have selected a correct file format. Otherwise, analysis results may be inaccurate.

2. Choose an operating system or an application version for running the file and specify additional settings if necessary.

You can select multiple OS versions or application versions: then multiple virtual machines will be launched. For example, if you select two Windows OS versions to analyze an executable file (.exe), Dr.Web vxCube will run two VMs.

3. Click **Analyze** to start checking the file.

You can run analysis of multiple files one by one. Click **Back** at the top of the page and then choose another file. The ⁛ icon displays progress of each analysis.



Upload a file and configure environment for analysis.
The file will be thoroughly analysed, and results will be reported shortly.

Choose file

test.exe EXE Browse

Choose OS: ☑ Windows XP 32-bit ☑ Windows 7 32-bit ☑ Windows 7 64-bit ☑ Windows 10 64-bit 1903

Figure 11. Uploading a file for analysis

6.3. Additional Settings

- **Sample name**

  Use this option if you want to submit the file for analysis under a different name. The original file won’t be overridden.

- **Use VNC**

  The use of VNC client is convenient if you choose more than one operating system and you want to influence the process on each of them.

  To activate the function, select the **Use VNC** check box. When you start the analysis, new browser tabs open automatically. Tabs are connected to the corresponding virtual machines via the VNC client. At the top of each tab, a progress bar is displayed. The bar shows the completion percentage and the current state of the analysis.

  Although new tabs open immediately, it can take some time to connect to virtual machines.

> ⚠ If you have not selected this option in **Additional settings** and have already started the analysis, click **Use VNC** on the analysis page. VNC client will open in a new tab.

- **Monitor all processes if VNC is used**

  By default, this setting is disabled and the report only includes the processes engaged in malicious activity.

- **Show MITM traffic**

  Select this check box if you want Dr.Web vxCube to parse encrypted traffic. This option is limited to Windows platforms. Once the analysis is done, you can view the decrypted traffic. To do this:

  1. Open the report page generated as a result of the analysis.
  2. Click zip **Download archive**.
  3. Unzip the archive. If prompted, input the password specified in the **Password for report archive** field in Settings. The default password is `vxcube`.
  4. Locate the `network.pcapng` file in the unpacked archive and upload it into a network packet analyzer like Wireshark.

- **Sample run time**

  The default sample run time in Dr.Web vxCube is 1 minute. You can adjust this value for the particular file if required. For example, you can increase the value if a file needs more time to show suspicious behavior. To do this, move the slider to the right.

- **Total size limit for drops**

  By default, the total size for files created during the analysis is limited to 64 MB. You can increase it to 512 MB.

- **Specify a command to run the file**

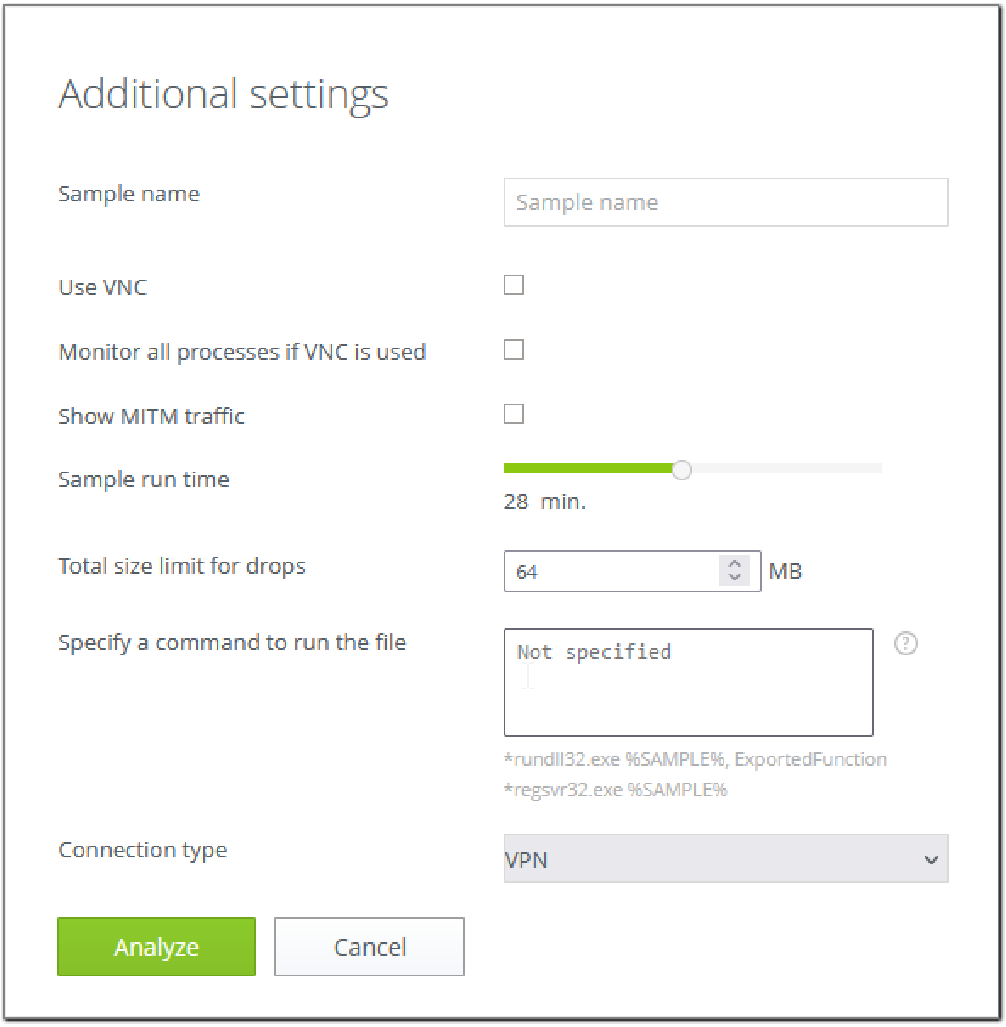
  This option allows you to set a specific command to run the file analysis. You can use any application from the standard Windows pack as a command, for example, `rundll32.exe`, `regsvr32.exe`, `notepad.exe`, etc. To use the command, specify it in the **Specify a command to run the file** field.

  You can specify a full path to the file using the special `%SAMPLE%` parameter.

  You can use this option if you need to run an executable file by calling an exported function. For example, `rundll32 %SAMPLE%, ExportedFunction`.

- **Connection type**

  VPN is used by default. For some connection types, you can specify a proxy server address and authorization parameters. Only TCP connections are proxied. Traffic of the other protocols is transferred through the default VPN server. To redirect UDP traffic, select the **Redirect UDP** check box.



Additional settings

Sample name: Sample name

Use VNC □

Monitor all processes if VNC is used □

Show MITM traffic □

Sample run time: 28 min.

Total size limit for drops: 64 MB

Specify a command to run the file: Not specified

*rundll32.exe %SAMPLE%, ExportedFunction

*regsvr32.exe %SAMPLE%

Connection type: VPN

Figure 12. Additional settings

After specifying additional settings

- Click **Analyze** to start analyzing the file.
- Click **Cancel** to reset settings and close the window.

> ⚠ Additional settings are only applied to the current file. If you close the **Additional settings** window or select another file, you will have to configure the settings again.

7. Reports

Information obtained during the analysis is recorded in a report. Then you can open or download the report.

7.1. Opening a Report

To open a report

- If you keep the analysis page open, the report opens automatically once the analysis is done.
- If you had left the page before the analysis was completed, select the file you were analyzing in the **History** section on the Dr.Web vxCube main page.

7.2. Downloading a Report

On the report page, you can use the download buttons to:

Download an original file.

Download a ZIP archive with the report. The default password for the archive is **vxcube**.

Download a report in either HTML or PDF format.

Download a PCAP file.

To download a report

1. At the top of the report page, select a platform.
2. Click **Download report** to open the **Report parameters** window.
3. Select a report format: HTML or PDF.
4. Select the sections you need to include in the report. The **API log** and **Intents** sections may contain thousands of records; you can filter the records by the degree of danger.
5. Click **Download report**.

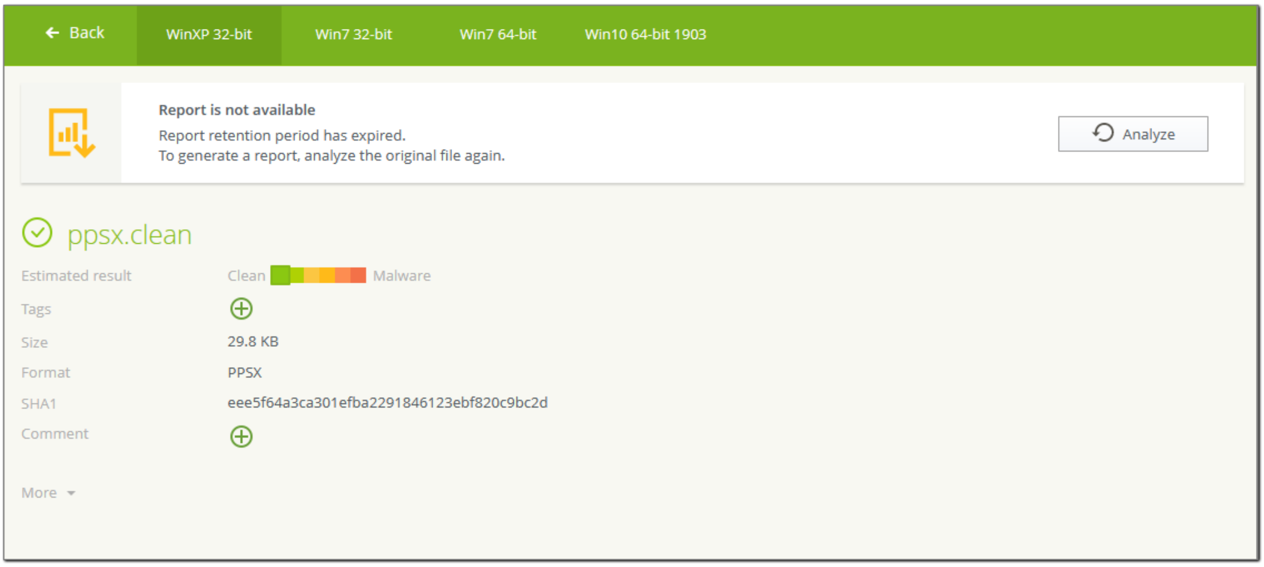
⚠ The **Intents** table is only present in reports for Android packages.

7.3. The Retention Period of Reports

The guaranteed period for keeping reports is set by an administrator. Once this period is over, a report could be removed from the server.

Once the report is removed, only the general information will remain on the report page, with a notification about the report's expiration at the top.

To generate the report again, restart the analysis. To do so, click **Analyze** on the report page.



Report is not available

Report retention period has expired.
To generate a report, analyze the original file again.

Estimated result	Clean Malware
Tags	⊕
Size	29.8 KB
Format	PPSX
SHA1	eee5f64a3ca301efba2291846123ebf820c9bc2d
Comment	⊕

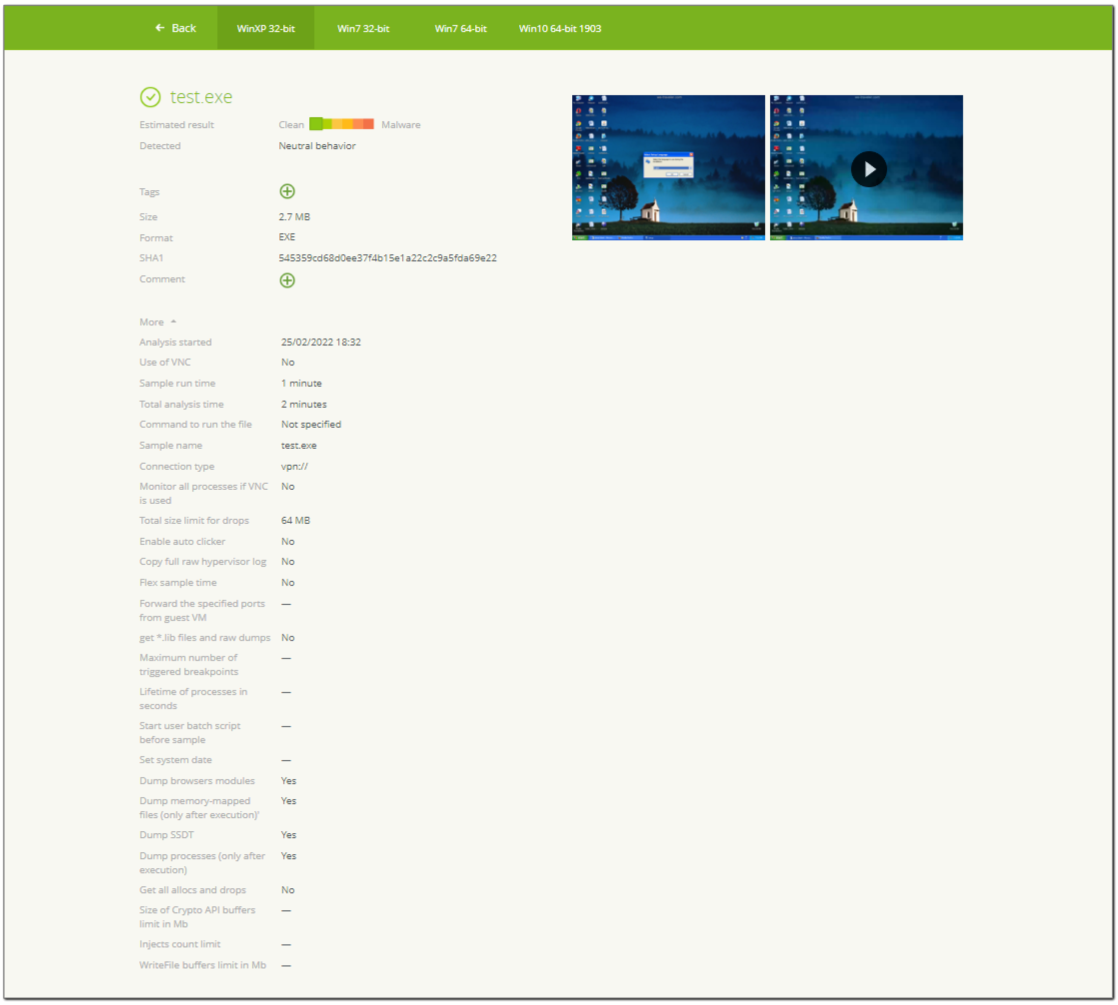
Figure 13. Notification about the report's expiration

7.4. The Report Structure

A report is divided into two parts: general information and main part.

General information consists of two sections: *basic details* and *more details*. Basic details provide a sample size, a sample format, an estimated analysis result, and other basic information. In the More details section you can find information such as a sample name, analysis start date and time, and total analysis time. Here you can also view the additional options, which are set for the analysis. You can explore these options, change them if needed and re-analyze the sample.

Main part can include the following sections: *Manifest*, *Behavior and YARA rules*, *Process graph*, *Description*, *Files and dumps*, *Phone calls and SMS*, *API Log and intents*, and *Network activity map*. The sections that are included in the list can differ based on a sample file format. For example, some of these sections are specific for reports of Android package analyses.



test.exe

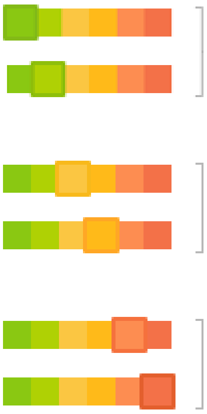
Estimated result	Clean 🟩🟨🟧🟥 Malware
Detected	Neutral behavior
Tags	⊕
Size	2.7 MB
Format	EXE
SHA1	545359cd68d0ee37f4b15e1a22c2c9a5fda69e22
Comment	⊕

More ▴

Analysis started	25/02/2022 18:32
Use of VNC	No
Sample run time	1 minute
Total analysis time	2 minutes
Command to run the file	Not specified
Sample name	test.exe
Connection type	vpn://
Monitor all processes if VNC is used	No
Total size limit for drops	64 MB
Enable auto clicker	No
Copy full raw hypervisor log	No
Flex sample time	No
Forward the specified ports from guest VM	—
get *.lib files and raw dumps	No
Maximum number of triggered breakpoints	—
Lifetime of processes in seconds	—
Start user batch script before sample	—
Set system date	—
Dump browsers modules	Yes
Dump memory-mapped files (only after execution)'	Yes
Dump SSDT	Yes
Dump processes (only after execution)	Yes
Get all allocs and drops	No
Size of Crypto API buffers limit in Mb	—
Injects count limit	—
WriteFile buffers limit in Mb	—

Figure 14. Report structure

7.4.1. General Information

Item	Description
Estimated result	Overall assessment of possible maliciousness.  Clean file Suspicious file Malware
Detected	Brief information on the file behavior and detected threats.

Item	Description
Tags	Tags added by a user or by a triggered YARA rule.
Size	File size.
Format	File format.
SHA1	File hash.
Comment	In this field, you can put any additional information you may need. There is a limit of 200 characters for a comment.
More	
Analysis started	Date and time the analysis started. It is counted from the moment the file was launched on a virtual machine.
Use of VNC	Use of the VNC client during the analysis (yes/no).
Sample run time	Sample run time that was specified in the additional settings of analysis.
Total analysis time	Total duration of file analysis.
Command to run the file	The command specified in the additional settings to run the file you are analyzing.
Sample name	The name of the file that was sent for analysis. More...
Connection type	The type of the connection. More...
Monitor all processes if VNC is used	Monitor all processes if VNC is used (yes/no). More...
Total size limit for drops	The limit on the total size of files generated during analysis. More...
Enable auto clicker	Enable auto clicker (yes/no).
Copy full raw hypervisor log	Copy full raw hypervisor log (yes/no).
Flex sample time	Use flex sample time (yes/no).
Forward the specified ports from guest VM	Forward the specified ports from guest VM. Example: `2343, 4353:tcp`.
Get `*.lib` files and raw dumps	Get `*.lib` files and raw dumps (yes/no).

Item	Description
Maximum number of triggered breakpoints	Set the maximum number of triggered breakpoints.
Lifetime of processes in seconds	Set the lifetime of processes. Example: `notepad.exe,35,winword.exe,20`.
Start user batch script before sample	Start a user batch script before running the sample.
Set system date	Set a system date on VM on which the analysis is performed. Example: `17.03.2022`.
Dump browser modules	Dump browser modules (yes/no).
Dump memory-mapped files (only after execution)	Dump memory-mapped files (only after execution) (yes/no).
Dump SSDT	Dump SSDT (yes/no).
Dump processes (only after execution)	Dump processes (yes/no).
Get all allocs and drops	Get all allocs and drops (yes/no).
Size of Crypto API buffers limit in MB	Set size of Crypto API buffers limit in MB. Example: `512`.
Injects count limit	Set a limit for injects. Example: `100`.
WriteFile buffers limit in MB	Set WriteFile buffers limit in MB. Example: `256`.

To the right from the general information part, there is a screenshot and a video report about the file’s behavior when it was run in a guest operating system.

7.4.2. Main Part

The main part may contain the following sections, depending on the sample format.

Section	Android packages (optional)	Other formats
Manifest	+	–
Behavior and YARA rules	+	+

Section	Android packages (optional)	Other formats
Process graph	–	+
Description	+	+
Files and dumps	+	+
Phone calls and SMS	+	–
API log and intents	+	**API log** only
Network activity map	+	+

7.4.2.1. Manifest (optional)

⚠ The section appears in reports for Android packages only.

The section contains the following information from the `AndroidManifest.xml` file:

Component	Comment
Package	Application package name.
Application name	Application name that appears to the user.
Version code	Internal version number.
Version name	Name and/or number of the version that appears to the user.
Permissions	Permissions that are requested by the application for its operation.

The section also contains the following components that are declared in the manifest: activities, broadcast receivers, and services.

7.4.2.2. Behavior and YARA rules

The section contains two tables: **Behavior** and **YARA rules**. To open a table, click its name.

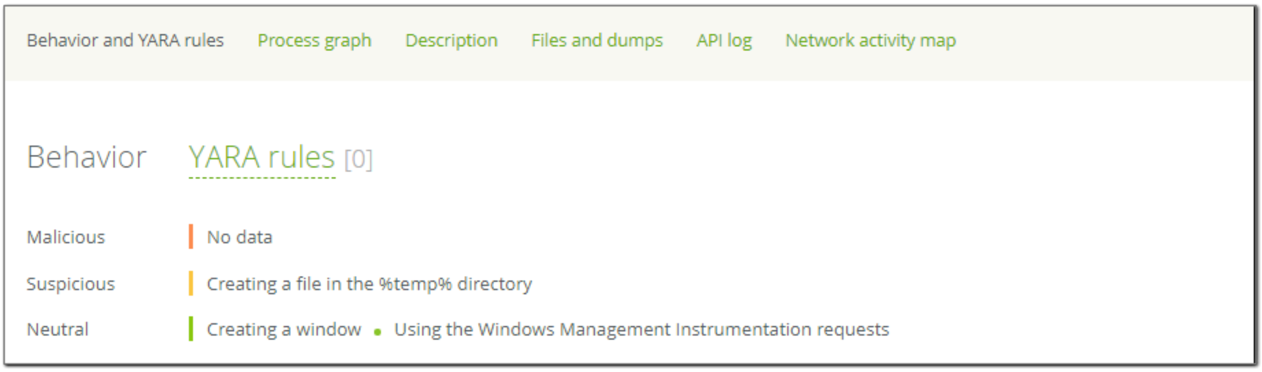
Behavior

The section contains a brief description on file behavior.

Dr.Web vxCube records all actions registered on a virtual machine throughout the analysis and categorizes them depending on how harmful they may be.

Dr.Web vxCube defines 3 categories of file behavior:

- Malicious
- Suspicious
- Neutral



Behavior and YARA rules Process graph Description Files and dumps API log Network activity map

Behavior YARA rules [0]

Malicious	No data
Suspicious	Creating a file in the %temp% directory
Neutral	Creating a window • Using the Windows Management Instrumentation requests

Figure 15. Reports on file behavior and YARA rules triggers

YARA Rules

The section contains information on YARA rule matches. The number of rules triggered during the analysis is displayed to the right from the table name.

The table displays information about the analysis results, tags, and triggered rule names.

To learn more about a rule, click its name.

To sort table columns in ascending or descending order, click the column titles.

7.4.2.3. Process Graph

The section is absent in reports for Android packages.

The section contains information about suspicious processes registered on a virtual machine. The data is represented as an interactive graph with an explanatory unit for each process.

To open the graph in a new tab, click the **Process graph** title. To zoom in or zoom out, click + or −. You can also zoom in by double-clicking the graph.

Conventions

Convention	Comment
⬜⬜🟨🟧🟥	Process or resource maliciousness. Measured on a scale from 0 to 100: ⬜ Less than 20. ⬜ Less than 40. 🟨 Less than 60. 🟧 Less than 80. 🟥 Less than 100.
▭	Process. The unit color corresponds to the process maliciousness.
☁ ☁☁ ☁☁☁	Network resource with remote access. The cloud color corresponds to the resource maliciousness. The protocol level and the IP address of the remote resource are displayed inside the cloud. 2 clouds are displayed if a process connects to the resource 2–5 times. 3 clouds are displayed if a process connects to the resource 6 times or more. In these cases, a number of connections is also displayed inside the cloud.
⌽	Sample. The sign is used to mark the first running process.
⚡	Known threat that is contained in the Dr.Web virus databases. The sign is used to mark a process if a threat is detected in its dump.
⚙	Known threat that is contained in the Dr.Web virus databases and that is detected in a dump of a loaded module. The sign is used to mark a process that a malicious module is loaded into. If threats are detected in both process and module dumps, the process is marked only with the ⚡ sign.
→	Process creation.
⇢	Injection into another process.
⤏	Web query.
⤑	RPC request.

Description

Click a process unit to show the information about a process in the description part.

Process parameters

Parameter	Description
PID	Process unique ID.
Full path	The path in which the process is run.
Run parameters	Special parameters for the process running. Optional field.
Behavior	The rules corresponding to tags about suspicious behavior of a process.
View the process activity	A link to the API log. Data in the log is filtered by process. To learn more about this feature, refer to API Log.
Download the dump file	Link for downloading the dump of the process.

Network resource parameters

Parameter	Description
Address	IP address of the network resource.
Port	Port number.

Protocol level	Protocol level of the OSI network model used for data transferring:
	• **Transport**
	• **Application**
	⚠ If the analyzer fails to determine the application level protocol, the following information will be displayed in this field:
	`Application: UNK`
	`Unknown data: {16,03,01,00,41,45…06,00,13,00,00,63,01,00}`

Query	This field is displayed if **Protocol level** is determined as **Application: DNS**.
URL	This field is displayed if **Protocol level** is determined as **Application: HTTP**.

7.4.2.4. Description

This section contains information about suspicious activity of a file, including objects, connections, etc. The data is grouped into categories and subcategories, depending on file behavior. The list of categories and subcategories is given below.

Enabling autorun and distribution

- Modifies the listed registry keys.
- Creates or modifies the listed files.
- Sets a service to autorun.
- Creates the listed services.
- Changes the listed executable system files.
- Replaces the listed executable system files.
- Replaces system binary files.
- Replaces system binary files using a symbolic link.
- Infects the listed executable files.
- Creates the following files on removable media.
- Modifies master boot record (MBR).
- Creates or modifies files to ensure autorun:
    - in /init.d;
    - in /router;
    - in /cron;
    - on desktop;
    - in other folders.
- Creates or modifies files to ensure autorun using symbolic links:
    - in /cron.
- Creates or modifies symbolic links to ensure autorun:
    - in /init.d;
    - on desktop;
    - in other folders.

Malicious functions

- Bypasses firewall, removes, or modifies the listed registry keys.
- To complicate detection of its presence in the operating system:
    - Forces the system to hide from view:
        - hidden files;
        - file extensions.
    - Blocks execution of the listed system utilities:

    - Command Prompt (CMD);
    - Windows Task Manager (Taskmgr);
    - Registry Editor (RegEdit).
    - Windows Firewall.
    - System Updates (Windows Update).
    - Windows Security Center.
    - System Anti-virus (Windows Defender).
  - Blocks the following features:
    - System Restore (SR);
    - Windows File Protection (WFP);
    - User Account Control (UAC);
    - System File Checker (SFC);
    - Windows Security Center.
    - Windows Support Center (Action Center).
  - Changes the listed system preferences:
    - changes the DNS server;
    - disables taskbar notifications.
  - removes shadow copies of volumes;
  - adds anti-virus exceptions using the listed registry keys.
- Creates and executes the listed processes:
  - creates and executes files (an exploit);
  - creates and loads libraries (an exploit);
  - downloads and executes files.
- Executes the listed processes.
- Injects code into the listed processes:
  - listed system processes;
  - listed user processes;
  - a large number of user processes.
- Installs hooks to intercept notifications:
  - About keystrokes:
    - Handler for all processes;
    - Handler for the listed processes.
- Terminates or attempts to terminate:
  - processes;
  - listed system processes;

    - listed user processes;
    - a large number of user processes.
    - processes of traffic analysis and program running applications;
    - processes by name.
- Searches for registry branches where third-party applications store passwords.
- Executes WMI operations.
- Registers a file system filter driver.
- Searches for the listed windows to:
    - bypass different anti-viruses;
    - bypass the Windows File Protection system;
    - detect analytics tools;
    - detect applications and games;
    - detect virtual machines.
- Creates an onion service.
- Loads the listed drivers.
- Hooks the following functions in the System Service Descriptor Table (SSDT):
    - a handler.
- Restores hooked functions in the System Service Descriptor Table (SSDT).
- Brute forces passwords of OS accounts.
- Performs a bruteforce attack in the network.
- Disables AMSI.
- Changes firewall settings.
- Changes router settings.
- Stops critical services.
- Manages services.
- Blocks through firewall:
    - SSH;
    - telnet;
    - standard web service ports.
- Modifies the listed settings of Windows Explorer.
- Modifies the listed settings of Windows Internet Explorer.
- Affects processes:
    - hides the listed processes;
    - traces processes;
    - injects itself in processes.

- Forces autorun for removable media.
- Sets a new unauthorized home page for Internet Explorer.
- Attempts to shut down Windows OS.
- Sends SMS.
- Executes the code of detectable threats.
- Downloads detectable threats from the internet.
- Sends contacts saved on the device to a remote server.
- Sends data on incoming SMS to a remote server.
- Overlays the interface preventing access to it.
- Sets a lock screen password.
- Prompts to install a third-party application.
- Hides its icon from screen.
- Ends incoming phone calls.
- Muffles incoming phone calls.
- Intercepts incoming SMS and terminates the process of their transmission to handlers of other apps.
- Deactivates a device administrator.
- Removes user data.
- Threat detection based on machine learning.
- Contains typical banking trojan/virus code.
- Contains typical locker code.
- Loads the listed detectable threats to be executed.
- Downloads the listed detectable threats from the internet.
- Launches a large number of processes.

File system changes

- Creates the listed files.
- Appends the “hidden” attribute to the listed files.
- Deletes the listed files.
- Sets a written file as executable.
- Sets a file as executable.
- Deletes a file.
- Deletes a system binary file.
- Creates or modifies symbolic links.
- Writes to system directory:
    - files;

  - symbolic links.
- Writes to system subdirectory:
  - files;
  - symbolic links.
- Writes to temporary directory:
  - files;
  - symbolic links.
- Creates directories:
  - in system subdirectory;
  - in system directory;
  - in temporary subdirectory;
  - in temporary directory;
  - in other directories.
- Removes directories:
  - in system directory;
  - in system subdirectory;
  - in temporary subdirectory;
  - in other directories.
- Moves the listed system files.
- Moves the listed files.
- Replaces the listed executable files.
- Modifies the HOSTS file.
- Replaces the HOSTS file.
- Moves itself.
- Deletes itself.
- Creates files.
- Changes access rights:
  - for a file;
  - for a written file.
- Changes owner:
  - for a file;
  - for a written file.
- Locks files.
- Changes the time when the file was created, accessed, or modified.
- Mounts file systems.

- Unmounts file systems.
- Creates files and demands payment for file decoding (Trojan.Encoder).
- Changes a large amount of user data (Trojan.Encoder).
- Changes file extensions in user data (Trojan.Encoder).
- Sets permissions to execute files.
- Adds an exclusion to Microsoft Defender.

Network activity

- Connects to a network resource.
- Opens a port.
- Sends data to a server.
- Receives data from a server.
- Accesses SSH.
- Connects to server through:
  - HTTP;
  - IRC.
- TCP:
  - HTTP GET requests;
  - HTTP POST requests;
  - HTTP HEAD requests;
  - HTTP PATCH requests;
  - HTTP PUT requests;
  - HTTP DELETE requests;
  - HTTP OPTIONS requests;
  - HTTP TRACE requests;
  - unknown HTTP requests.
- UDP:
  - DNS requests.

Miscellaneous

- Adds a root certificate.
- Disables certificate.
- Collects information:
  - on the OS;
  - on the CPU;
  - on the RAM;
  - on the network activity.

- Changes value of the AutoConfigURL parameter as follows.
- Substitutes an application name.
- Searches for the listed windows.
- Creates and executes files.
- File protected with the Themida packer by Oreans Technologies.
- Uses NTFS alternate data streams.
- Loads the listed drivers.
- Unloads the kernel module.
- Sets kernel module to autorun.
- Executes shell scripts.
- Runs as daemon.
- Compiles source code.
- Reads information from /proc/kallsyms.
- Loads dynamic libraries.
- Makes phone calls.
- Uses data encryption algorithms.
- Uses data decryption algorithms.
- Uses elevated privileges.
- Uses administrator rights.
- Gains root access.
- Accesses the ITelephony private interface.
- Uses libraries to hide executable bytecode.
- Can send SMS automatically.
- Accesses audio/video recording interfaces.
- Records audio/video.
- Accesses camera interface.
- Changes volume and vibration settings.
- Accesses location of the device.
- Accesses network information.
- Gets information about the device (phone number, IMEI, etc.).
- Gets information about APN settings.
- Gets information about active device administrators.
- Gets information about installed apps.
- Gets information about running apps.
- Gets information about accounts linked with the device.

- Adds tasks to the System Scheduler.
- Displays its windows over windows of other apps.
- Processes information from SMS.
- Gets information about incoming/outgoing phone calls.
- Gets information about sent/received SMS.
- Gets information about phone contacts.
- Enables/disables all cameras.
- Manages Wi-Fi connectivity.
- Checks for anti-virus applications.
- Intercepts notifications.
- Requests permission to display system alert windows.
- Sample from Google Play Store.
- Restarts the analyzed sample.

7.4.2.5. Files and Dumps

The section contains two tables: **Created files** and **Dumps**. The number of objects detected during the analysis is displayed to the right from the table name.

To open a table, click its name.

To sort table columns in ascending or descending order, click the column titles.

To download a file from the table, click **Download the file** . If Dr.Web vxCube has not collected the file due to resource constraints, you are not able to download the file. In this case the icon is displayed.

Created files

The table contains information about files created during the analysis. The table displays a path, hash, and name of a detected threat.

Dumps

The table contains information about the following objects:

- Dumps.
- Injections.
- Memory blocks that are allocated by the running sample. Memory allocations may contain traces of malicious activity.

The table displays a file name, hash, unique number of a process (PID), and name of the detected threat.

> ⚠ The name of the detected threat is displayed only if it is in the Dr.Web database.

7.4.2.6. Phone Calls and SMS (optional)

> ⚠ The section appears in reports for Android packages only.

The section contains information about outgoing phone calls and SMS messages, that have been made by the analyzed application. The table contains receivers' phone numbers and message texts.

7.4.2.7. API Log and Intents

The section contains two tables: **API log** and **Intents**.

> ⚠ The **Intents** table appears in reports for Android packages only.

The number of objects detected during the analysis is displayed to the right from the table name.

To open a table, click its name.

To sort table columns in ascending or descending order, click the column titles.

To filter the data by maliciousness, click one of the colors in the ■ ■ ■ ■ ■ scale. The filter includes the upper level of maliciousness into the previous one.

API Log

The **API log** table contains information about all events that occurred on the virtual machine while the file was running. The **API log** table contains structured data from the Process graph section.

Click **Open API Log in a New Tab** to open this section in a new browser tab.

Parameter	Comment
Time	Time of the process. Counted from the moment the file analysis started.
Process	The full path to the process in the host operating system.
Event	An event which occurs while the file is running. It corresponds to the commonly used API functions.
Arguments	Arguments of the events. They indicate special conditions for executing events.

Intents

The **Intents** table contains the intents that were sent by the analyzed application to start components of other applications.

Parameter	Comment
Time	Time of the action. Counted from the moment the file analysis started.
Data	Data to perform the action upon.
Action	Name of action to perform.
Transaction	Transaction defining a type of component to start: • START_ACTIVITY—starting an activity. • START_SERVICE—starting a service. • BROADCAST_INTENT—delivering a broadcast.
Component name	Component that receives the intent.

7.4.2.8. Network Activity Map

The section contains information about data that was sent by a file and where it was transferred. The connections are marked on an interactive map. For more information on each connection, refer to the table below the map.

Parameter	Description
Protocol	Protocol that is used for the connection.
Address	Address for connection.
Application-level data	DNS request, URL request, or unknown data.

You can sort the **Protocol** and **Address** columns in ascending or descending order. To do this, click the header of the column you want to sort by. At the left of the header ▲ or ▼ will appear. To change the sorting direction, click the header again.

> By default, the map will only show connections that are initiated by the sample itself. To include connections initiated by you through the VNC client, select the **Monitor all processes if VNC is used** check box in the additional settings before starting the analysis.

7.5. History

History contains information about file analyses that have been performed before. The history section is located on the Dr.Web vxCube main page below the file uploading section.

History allows you to:

- Search for a string, filter and sort entries.
- Check the progress of the ongoing analysis.
- View, delete, and download reports of analyzed files.

History management

To set a number of entries displayed on one page

- Click the drop-down menu below the table.

To sort entries

- Click the corresponding column title.

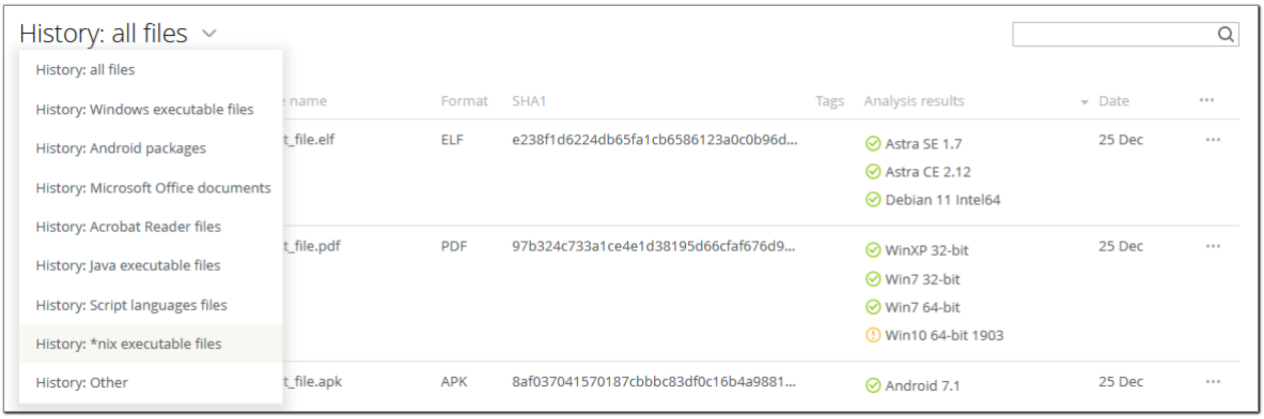
  You can sort entries by user login (if you have administrator privileges), file name, or date.

To filter entries

- Type a string into the search box. You can search across all table columns.
- Click **History** to filter by file type.

To select which columns to display

- Click ••• in the right corner of the table.
- Select the columns you want to display.


History: all files
History: all files
History: Windows executable files
History: Android packages
History: Microsoft Office documents
History: Acrobat Reader files
History: Java executable files
History: Script languages files
History: *nix executable files
History: Other

name	Format	SHA1	Tags	Analysis results	Date	···
t_file.elf	ELF	e238f1d6224db65fa1cb6586123a0c0b96d...		✓ Astra SE 1.7 ✓ Astra CE 2.12 ✓ Debian 11 Intel64	25 Dec	···
t_file.pdf	PDF	97b324c733a1ce4e1d38195d66cfaf676d9...		✓ WinXP 32-bit ✓ Win7 32-bit ✓ Win7 64-bit ! Win10 64-bit 1903	25 Dec	···
t_file.apk	APK	8af037041570187cbbbc83df0c16b4a9881...		✓ Android 7.1	25 Dec	···

Figure 16. Selecting file type

To open a page with analysis report

- Click the corresponding file name.

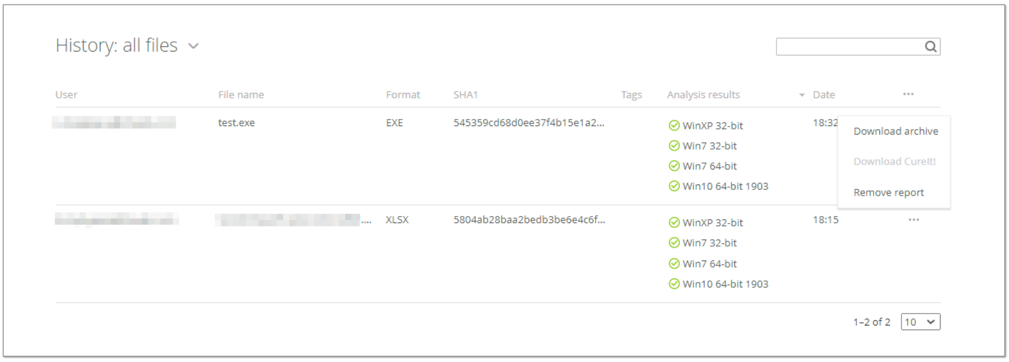
To download analysis report

- Hover over the ··· icon corresponding to the required file and select **Download archive**.

  The detailed report will be downloaded as a ZIP archive.

To remove analysis report

- Hover over the ··· icon corresponding to the required file and select **Remove report**.



User	File name	Format	SHA1	Tags	Analysis results	Date	···
[redacted]	test.exe	EXE	545359cd68d0ee37f4b15e1a2...		✓ WinXP 32-bit ✓ Win7 32-bit ✓ Win7 64-bit ✓ Win10 64-bit 1903	18:32	
[redacted]	[redacted]....	XLSX	5804ab28baa2bedb3be6e4c6f...		✓ WinXP 32-bit ✓ Win7 32-bit ✓ Win7 64-bit ✓ Win10 64-bit 1903	18:15	···

Figure 17. Actions available in the History section

7.6. Tags

To make it easier to work with reports, use special classification labels, *tags*. You can add tags in two ways:

- When adding a YARA rule. Then, if this particular rule is triggered during the analysis, the report will automatically have the specified tags added to it.
- Manually added to the generated report. To do this:

  1. Click ⊕ in the **Tag** section of the report.

2. Enter a tag name using letters, digits, or underscore.

3. Click +.

8. API

Dr.Web vxCube API allows you to:

- Automatically analyze files
- Analyze more files in less time
- Automatically sort results

We recommend you to use our Dr.Web vxCube API Client as it simplifies the interaction with vxCube. With this API client, you won’t need to generate queries manually for actions like sending samples for analysis, receiving the analysis results, or downloading reports.

Currently, Dr.Web vxCube API v2.0 is used. This version only supports the JSON format. Use the following base URL address for all your API requests:

```
https://<IP address/domain name of the server>/api-2.0/
```

8.1. Authentication

Every API request to Dr.Web vxCube service should be authenticated using an API key. The key serves as a user ID or access key to the service, much like a login and password on a web interface. To authenticate, add an `Authorization` header with an API key to your API request.

Example request

```
curl -X GET https://<IP address/domain name of the server>/api-2.0/analyses/60e21c98-
7c2a-4112-81b5-a577f6cdf4db \
-H "Content-Type: application/json" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
```

You can create an API key on the service’s web interface or by sending an API request.

8.2. Managing API Keys

You can create new API keys, view, or delete existing ones.

At most, you can have 10 API keys. If you have reached the limit, but want to add another key, start by deleting one or more existing keys.

To create an API key on the service’s web interface

1. In the top right-hand corner of the main page, click 👤 **Profile > Settings**.
2. On the left, select the **API sessions** tab.

3. In the **New key** field, enter the key name, then click +. The key appears in the **Existing keys** list.
4. If you want to copy the newly created key, click ⧉ on the right of it.

To create an API key by sending an API request

- Send the POST login request.

To view your existing API keys

1. In the top right-hand corner of the main page, click 👤 **Profile** > **Settings**.
2. On the left, select the **API sessions** tab.
3. You can find the API keys you have in the **Existing keys** list.

> ⚠ If you've already had an API key, you can retrieve this by sending the API request POST login.

To delete an API key

1. In the top right-hand corner of the main page, click 👤 **Profile** > **Settings**.
2. On the left, select the **API sessions** tab.
3. In the **Existing keys** list, click 🗑 on the right of the key.

> ⚠ You can undo the deletion of the API key. To do this, click **Restore** on the right of API key deletion info. But if you close the **Settings** window, the **Restore** button will disappear and the API key will be permanently deleted.

8.3. Endpoints

8.3.1. analyses

Use the endpoint to manage analyses.

DELETE analyses/<analysis_id:uuid>

Description	Parameters	Result
Delete analysis.	—	Analysis is deleted, code 204.

GET analyses

Description	Result
Get data about analyses.	List of Analysis objects.

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`format_group_name`	string	Filter by file type.	No

GET analyses/<analysis_id:uuid>

Description	Parameters	Result
Get detailed information about analysis.	—	Analysis object.

Usage example

GET analyses/<analysis_id:uuid>/archive

Description	Parameters	Result
Download the archive with analysis results.	—	Archive that contains analysis results on all tasks.

Usage example

GET analyses/<analysis_id:uuid>/sample

Description	Parameters	Result
Download sample.	—	Sample.

POST analyses

Description	Result
Start the file analysis.	Analysis object.

Parameters

Parameter	Type	Description	Required
`analysis_time`	integer	Sample run time in seconds, from 30 to 300. By default, `analysis_time=60`.	No
`convert_video`	boolean	Convert video while the analysis is ongoing.	No
`copylog`	boolean	Copy full raw hypervisor log.	No
`crypto_api_limit`	integer	Crypto API buffers limit in MB.	No
`custom_cmd`	string/null	Command to run the sample.	No
`drop_size_limit`	integer	Total size limit for created files.	No
`dump_browsers`	string	Dump browser modules.	No
`dump_mapped`	boolean	Dump memory-mapped files (only after execution).	No
`dump_processes`	boolean	Dump processes (only after execution).	No
`dump_size_limit`	integer	Maximum size of collectable drops.	No
`dump_ssdt`	boolean	Dump SSDT.	No
`flex_time`	boolean	Sample flex time.	No
`format_name`	string	File format.	Yes if the format is not identified automatically
`forwards`	array [string]/null	Forward the specified ports from guest VM.	No
`generate_cureit`	boolean	Generate the Dr.Web CureIt! utility for neutralizing threats in the original file and in all files created during the analysis.	No

Parameter	Type	Description	Required
`get_lib`	boolean	Get `*.lib` files and raw dumps.	No
`injects_limit`	integer	Injects count limit.	No
`monkey_clicker`	boolean	Enable auto clicker.	No
`net`	string	Command to redirect virtual machine network traffic according to specified settings. • `VPN = vpn://` (used by default if the net parameter is not specified) • `TOR = tor://` • `Socks4 = socks4://host:port` • `Socks5 = socks5://[login:password@]host:port?parameters` • `Shadowsocks = shadowsocks://[login:password@]host:port?parameters` Possible values for `parameters`: `udp`—UDP protocol behavior (`udp=on` redirects all UDP traffic, `udp=off` does not redirect traffic); `login:password`—proxy server authorization parameters (optional for Socks5, required for Shadowsocks).	No
`no_clean`	boolean	Get all allocs and drops.	No
`optional_count`	integer/null	Maximum number of triggered breakpoints.	No
`platforms`	array [string]/null	Platforms to run the sample.	No
`proc_lifetime`	string/null	Lifetime of processes in seconds. Example: `'notepad.exe,35,winword.exe,20`	No
`sample_id`	integer	Sample ID.	Yes
`set_date`	string	Set system date (format: 17.03.2022).	No
`write_file_limit`	integer	WriteFile buffers limit in MB.	No

Usage example

POST analyses/<analysis_id:uuid>/restart

Description	Parameters	Result
Restart all deleted or failed tasks of the specified analysis.	—	Restart of deleted or failed tasks.

8.3.2. formats

Use the endpoint to get data about supported formats.

GET formats

Description	Parameters	Result
Get a list of formats supported by Dr.Web vxCube.	—	List of Format objects.

8.3.3. login

Use the endpoint to get one of the existing API keys or to create a new one. You can have a maximum of 10 API keys.

POST login

Description	Result

Get API key.	`{`
	`  "new_key": <true or false>`
	`  "api_key": "<API key>"`
	`  "start_date": "<date>"`
	`  "name": <key name>`
	`}`

Parameters

Parameter	Type	Description	Required
`login`	string	User login.	Yes
`password`	string	User password.	Yes

Parameter	Type	Description	Required
`new_key`	boolean	Determines whether to create a new API key or to get one of the created earlier. By default `new_key=false`. If you have not got any API keys created, you do not have to specify the parameter—API key will be created anyway.	No
`name`	string	The name that will be used to describe this API key.	No

Usage example

8.3.4. platforms

Use the endpoint to get data about supported platforms.

GET platforms

Description	Parameters	Result
Get a list of supported platforms.	—	List of Platform objects.

Supported platforms

The following table lists the supported platforms for the various file formats.

File format	Supported platforms
EXE, DLL, CPL, SYS, Native App, MSI, JS, VBS, WSF, JSE, VBE, PS1, BAT, SCT, XSL, MOF, LNK, HTA, CHM, CDF, EML, ZIP, ARJ, XZ, TAR, BZ2, CAB, GZ, RAR, 7Z	winxpx86, win7x86, win7x64, win10x64_1511, win10x64_1903
ACCDB, DOC, DOCM, DOCX, DOTM, DOTX, IQY, MDB, MHT, ODP, ODS, ODT, POTM, POTX, PPA, PPAM, PPSM, PPSX, PPT, PPTM, PPTX, PUB, RTF, SLDM, SLDX, SLK, THMX, XLAM, XLL, XLS, XLSB, XLSM, XLSX, XLTM, XLTX, XML, WPS	office_xp, office_7_32, office_7_64, office_10_64_1511, office_10_64_1903
PDF	acrobat_xp_10, acrobat_7_32_11, acrobat_7_64_15, acrobat_10_64_1511_15, acrobat_10_64_1903_15
JAR, CLASS	java_xp, java_7_32, java_7_64, java_10_64_1511, java_10_64_1903

File format	Supported platforms
SH, PY, PL, EL, DOCKER	arm64_debian_bullseye, armel32_debian_jessie, armhf32_debian_bullseye, intel32_debian_bullseye, intel64_astra_ce_2.12, intel64_astra_se_1.7.2, intel64_debian_bullseye, mips32_debian_buster, mipsel32_debian_bullseye, mipsel64_debian_bullseye, ppc32_debian_jessie, ppcel64_debian_bullseye
APK	android4.3, android7.1

8.3.5. samples

Use the endpoint to manage samples.

GET samples

Description	Result
Get a list of samples that were uploaded earlier.	List of Sample objects.

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`md5`	string	Filter by MD5.	No
`sha1`	string	Filter by SHA1.	No
`sha256`	string	Filter by SHA256.	No
`format_name`	string	Filter by file format.	No
`format_group_name`	string	Filter by file type.	No

GET samples/<sample_id:number>

Description	Parameters	Result
Get data about the file that was uploaded earlier.	—	Sample object.

GET samples/<sample_id:number>/analyses

Description	Parameters	Result
Get data about the file analyses.	—	Analysis object.

POST samples

Description	Result
Upload a sample to the Dr.Web vxCube server.	Sample object.

Parameters

Parameter	Type	Description	Required
`file`	string	The sample that needs to be uploaded to the server. Specify a full filepath preceded by the `@` symbol.	Yes
`password`	string	The password for the uploaded archive. The password can be 1 to 25 characters long.	No

Usage example

8.3.6. sessions

Use the endpoint to manage sessions.

DELETE sessions/<api_key:string>

Description	Parameters	Result
Delete the session with the specified API key.	—	Session is deleted, code 204.

GET sessions

Description	Parameters	Result
Get a list of all open sessions.	—	List of Session objects.

8.3.7. tasks

Use the endpoint to manage analysis tasks and report data.

GET tasks/<task_id:number>

Description	Parameters	Result
Get data about the task.	—	Task object.

GET tasks/<task_id:number>/archive

Description	Parameters	Result
Download archive with analysis results.	—	Archive with analysis results.

GET tasks/<task_id:number>/sample

Description	Parameters	Result
Download sample.	—	Sample.

GET tasks/<task_id:number>/report

Description	Parameters	Result
Download one-page HTML report.	—	One-page HTML report.

GET tasks/<task_id:number>/graph

Description	Parameters	Result
Download SVG graph.	—	SVG graph.

GET tasks/<task_id:number>/dumps

Description	Result

Get data from the Dumps table.	{
	"total_count": *<number>*,
	"items": *<list of Dump objects>*
	}

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`search`	string	Pattern for string searching.	No

GET tasks/<task_id:number>/drops

Description	Result

Get data from the Created files table.	{
	"total_count": *<number>*,
	"items": *<list of Drop objects>*
	}

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`search`	string	Pattern for string searching.	No

GET tasks/<task_id:number>/networks

Description	Result

Get data from the Network activity map table.	`{`
	`"total_count": <number>,`
	`"items": <list of Connection objects>`
	`}`

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No

GET tasks/<task_id:number>/api_log

Description	Result

Get data from the API log table.	`{`
	`"total_count": <number>,`
	`"items": <list of APIEvent objects>`
	`}`

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`search`	string	Pattern for string searching.	No

GET tasks/<task_id:number>/intents (optional)

Description	Result

Get data from the Intents table. The endpoint is used for tasks started on Android.	`{`
	`"total_count": <number>,`
	`"items": <list of Intent objects>`
	`}`

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`search`	string	Pattern for string searching.	No

GET tasks/<task_id:number>/phone_actions (optional)

Description	Result

Get data from the Phone calls and SMS table. The endpoint is used for tasks started on Android.	`{`
	`"total_count": <number>,`
	`"items": <list of Call and Message objects>`
	`}`

Parameters

Parameter	Type	Description	Required
`count`	integer	Number of returning objects, 1...100. To get more objects, use several requests and the `offset` parameter. By default, `count=10`.	No
`offset`	integer	Offset, 0...+∞. By default, `offset=0`.	No
`search`	string	Pattern for string searching.	No

GET tasks/<task_id:number>/archive_storage

Description	Parameters	Result

Get a list of files and directories in the archive, or download a file or a directory from the archive.	`path` (string)—path, optional	If `path` is not specified:
		`{`
		`"folders":` *<list of folders in the archive>*`,`
		`"files":` *<list of files in the archive>*
		`}`
		If `path` is specified, file or archive of the folder

Usage example

POST tasks/<task_id:number>/restart

Description	Parameters	Result
Restart the deleted or failed task.	—	Restart of the deleted or failed task.

8.3.8. ws/progress

To connect over the WebSocket protocol and get data about the analysis progress in real time, in the request, specify the following JSON object as a string:

```
{"analysis_id": <ID>}
```

In response, you receive the JSON object:

```
{'message': '<message>', 'progress': <progress>, 'task_id': <ID>}
```

8.4. Objects

8.4.1. Analysis

The **Analysis** object contains general analysis information and a list of Task objects.

Structure

Key	Type	Description
id	UUID	Task UUID.
sha1	string	SHA1 hash.
sample_id	integer	Sample ID.
size	integer	File size in bytes.
format_name	string/null	File format. Matches the Sample.format_name format, if the format was not specified explicitly when starting the file analysis.
start_date	string (datetime.iso8601)	Date and time the analysis started.
user_name	string	User login.
tasks	array [Task]	List of tasks. Corresponds to the selected platforms.

Examples

If you request a certain analysis by its ID, in response, you receive the **Analysis** object where the tasks key is a list of TaskFinished or TaskProcessing objects:

```
{
  "id": 1629b17b-fd44-46e6-97a2-1310c1f050a4,
  "sample_id": 6248,
  "size": 3242863,
  "sha1": "8c81eb1b6a87e30656d479968eca969bc59bdeb3",
  "start_date": "2018-12-12T11:29:44.645968+00:00",
  "user_name": "name_example",
  "format_name": "rtf",
  "tasks": [
    {
      "end_date": "2018-12-12T11:33:37.490050+00:00",
      "platform_code": "winxpx86",
      "maliciousness": 100,
      "id": 16916,
      "status": "successful",
      "start_date": "2018-12-12T11:29:44.645968+00:00",
      "rules": {
        "neutral": [
          "Searching for the window",
          "Creating a window",
          "DNS request",
          "Sending an HTTP GET request"
        "suspicious": [
          "Connection attempt by exploiting the app vulnerability"
        ]
      },
      "detects": [
        "behavior",
        "files_dumps"
        ],
      "verdict": "malware2"
    },
    {
      "end_date": "2018-12-12T11:33:47.716867+00:00",
      "platform_code": "win7x86",
      "maliciousness": 100,
      "id": 16917,
      "status": "successful",
      "start_date": "2018-12-12T11:29:44.645968+00:00",
      "rules": {
        "neutral": [
          "Creating a window",
          "DNS request",
          "Sending an HTTP GET request",
          "Creating a process from a recently created file",
          "Launching a process"
        ],
        "suspicious": [
          "Connection attempt by exploiting the app vulnerability"
        ]
      },
      "detects": [
        "behavior",
        "files_dumps"
        ],
      "verdict": "malware2"
    },
    {
      "end_date": "2018-12-12T11:34:08.229276+00:00",
      "platform_code": "win7x64",
      "maliciousness": 100,
      "id": 16918,
      "status": "successful",
      "start_date": "2018-12-12T11:29:44.645968+00:00",
      "rules": {
        "neutral": [
          "Creating a window",
          "DNS request",
          "Sending an HTTP GET request",
          "Creating a file in the %temp% directory",
          "Launching a process",
          "Launching the default Windows debugger (dwwin.exe)"
        ],
        "suspicious": [
          "Connection attempt by exploiting the app vulnerability"
        ]
      },
      "detects": [
        "behavior",
        "files_dumps"
        ],
      "verdict": "malware2"
    },
    {
      "end_date": "2018-12-12T11:35:11.553665+00:00",
      "platform_code": "win10x64_1903",
      "maliciousness": 100,
      "id": 16919,
      "status": "successful",
      "start_date": "2018-12-12T11:29:44.645968+00:00",
      "rules": {
        "neutral": [
          "Creating a window",
          "Sending an HTTP GET request"
        ],
        "suspicious": [
          "Connection attempt by exploiting the app vulnerability"
        ]
      },
      "detects": [
        "behavior",
        "files_dumps"
        ],
      "verdict": "malware2"
    }
  ]
}
```

If you request a list of analyses using the GET analyses method, in response, you receive a list of **Analysis** objects, each contains the `tasks` key—a list of TaskBasic objects:

```
{
    "id": 1629b17b-fd44-46e6-97a2-1310c1f050a4,
    "sample_id": 6248,
    "size": 3242863,
    "sha1": "8c81eb1b6a87e32152d439965eca944bc59bdeb3",
    "start_date": "2018-12-12T11:29:44.645968+00:00",
    "user_name": "name_example",
    "format_name": "rtf",
    "tasks": [
      {
        "end_date": "2018-12-12T11:33:37.490050+00:00",
        "platform_code": "winxpx86",
        "maliciousness": 100,
        "id": 16916,
        "status": "successful",
        "start_date": "2018-12-12T11:29:44.645968+00:00"
      },
      {
        "end_date": "2018-12-12T11:33:47.716867+00:00",
        "platform_code": "win7x86",
        "maliciousness": 100,
        "id": 16917,
        "status": "successful",
        "start_date": "2018-12-12T11:29:44.645968+00:00"
      },
      {
        "end_date": "2018-12-12T11:34:08.229276+00:00",
        "platform_code": "win7x64",
        "maliciousness": 100,
        "id": 16918,
        "status": "successful",
        "start_date": "2018-12-12T11:29:44.645968+00:00"
      },
      {
        "end_date": "2018-12-12T11:35:11.553665+00:00",
        "platform_code": "win10x64_1903",
        "maliciousness": 100,
        "id": 16919,
        "status": "successful",
        "start_date": "2018-12-12T11:29:44.645968+00:00"
      }
    ]
  }
```

8.4.2. APIEvent

The **APIEvent** object contains data about an event that occurred while the sample was running.

Structure

Key	Type	Description
`process`	string	The full path to the process in the host operating system.
`rules`	object	List of triggered rules.
`arguments`	string	Arguments of the event. They indicate special conditions for executing events.
`maliciousness`	integer	Maliciousness, from 0 to 100.
`event`	string	An event which occurs while the file is running. It corresponds to the commonly used API functions.
`timestamp`	integer	Event timestamp. Counted from the moment the file analysis started.

Example

```
{
  "process": "<CURRENT_DIR>\\example.exe:1432:2432",
  "rules": {
    "neutral": [
      "Connection attempt"
    ]
  },
  "arguments": "To '125.251.199.120':540",
  "maliciousness": 0,
  "event": "ConnectNet",
  "timestamp": 9
}
```

8.4.3. Call (optional)

The **Call** object contains data about an outgoing phone call. The object is used only in results of Android app analysis.

Structure

Key	Type	Description
`type`	string	Always `call`.
`number`	string	Phone number the call was made to.

Example

```
{
  "type": "call",
  "number": "667206"
}
```

8.4.4. Connection

The **Connection** object contains data about a network connection.

Structure

Key	Type	Description
`port`	integer	Port number.
`host`	string	Host name or IP address.
`country`	object	Country.
`app`	string	Application-level data.
`protocol`	string	Protocol that is used for the connection.
`ip`	string	Host IP address.

Example

```
{
  "port": 31,
  "host": "<IP address>",
  "country": {
    "name": "China",
    "code3": "CHN"
  },
  "app": "{70,69,6e,67}",
  "protocol": "TCP/IP",
  "ip": "<IP address>"
}
```

8.4.5. Dump

The **Dump** object contains data about a potentially malicious dump of a process.

Structure

Key	Type	Description
archive_path	string	Path to the file in the report archive.
name	string	File name.
sha1	string	SHA1 hash.
detect	string	Threat name.
pid	integer	Process identifier.

Example

```
{
  "archive_path": "dumps/4_89432000_a71a8d8316cb3bc.4.38.6.ndmp",
  "name": "a71a8d8316cb3bc",
  "sha1": "8f11bc1fb9ac4444472213e0ae91bc166493f0ab",
  "detect": "Trojan.Necurs.5",
  "pid": 4
}
```

8.4.6. Drop

The **Drop** object contains data about a file created during the analysis.

Structure

Key	Type	Description
archive_path	string	Path to the file in the report archive.
sha1	string	SHA1 hash.
detect	string	Threat name.
path	string	Path to the created file.

Example

```
{
  "archive_path": "drops/d##vault.hta(0)",
  "sha1": "392b84af9ede8fc70a29f02131e9ae91ef88c809",
  "detect": "JS.DownLoader.994",
  "path": "D:\\vault.hta"
}
```

8.4.7. Format

The **Format** object contains data about a file format.

Structure

Key	Type	Description
`name`	string	The name of the file format.
`group_name`	string	The name of the file type. Possible values: • `apk`: Android packages. • `arf`: Acrobat Reader files. • `ja`: Java executable files. • `js`: script files. • `moc`: Microsoft Office documents. • `other`: other types. • `uef`: *nix executable files. • `wef`: Windows executable files.
`platforms`	array [Platform.code]	The list of platforms.

Example

```
{
   "name": "exe",
   "group_name": "wef",
   "platforms": [
     "winxpx86",
     "win7x86",
     "win7x64",
     "win10x64_1903"
   ]
}
```

8.4.8. Intent (optional)

The **Intent** object contains data about an intent. The object is used only in results of Android app analysis.

Structure

Key	Type	Description
`cn`	string	Component that receives the intent.
`action`	string	Name of action to perform.
`data`	string	Data to perform the action upon.
`transaction`	string	Transaction defining a type of component to start: • START_ACTIVITY—starting an activity. • START_SERVICE—starting a service. • BROADCAST_INTENT—delivering a broadcast.
`maliciousness`	integer	Maliciousness, from 0 to 100.
`rules`	object	List of triggered rules.
`timestamp`	integer	Timestamp. Counted from the moment the file analysis started.

Example

```
{
  "cn": null,
  "action": "android.app.action.ADD_DEVICE_ADMIN",
  "data": null,
  "transaction": "START_ACTIVITY",
  "maliciousness": 69,
  "rules": {
    "suspicious": [
      "Using device administration features"
    ]
  },
  "timestamp": 0
}
```

8.4.9. Message (optional)

The **Message** object contains data about an outgoing SMS message. The object is only used in the results of Android app analysis.

Structure

Key	Type	Description
`type`	string	Always `message`
`number`	string	Phone number the message was sent to.
`text`	string	Text of the message.

Example

```
{
  "type": "message",
  "number": "000",
  "text": "Balance"
}
```

8.4.10. Platform

The **Platform** object contains data about an OS platform and, in some cases, about an application for running the sample.

Structure

Key	Type	Description
`code`	string	Short name of the platform.
`name`	string	App name or OS platform.
`os_code`	string	OS platform.

Example

```
{
    "code": "office_7_32",
    "name": "Microsoft Office 2010",
    "os_code": "Windows 7 32-bit"
}
```

8.4.11. Sample

The **Sample** object contains data about an original file uploaded for analysis.

Structure

Key	Type	Description
`id`	integer	Sample ID.
`name`	string	File name.
`format_name`	string	File format. It is identified by Dr.Web vxCube automatically. File format determines the command for file running if the command is not specified explicitly when starting the file analysis.
`is_x64`	boolean	Determines the bitness of the platform for file running. It is `null` if the file is not executable.
`md5`	string	MD5 hash.
`sha1`	string	SHA1 hash.
`sha256`	string	SHA256 hash.
`size`	integer	File size in bytes.
`upload_date`	string	Date and time the file was uploaded.
`platforms`	array [Platform.code]	List of supported platforms for file running.

Example

```
{
  "id": 42,
  "name": "sample.exe",
  "format_name": "sys",
  "is_x64": null,
  "md5": "a0b0f87193b79ac1db32f251f2f5e5b2",
  "sha1": "e54639e9d81680d0acc154d42ae7350ed481b848",
  "sha256": "51133e7e4d52b94e3360ac1866b76bf2b2bca056492bcf93de3c37d6b0c07104",
  "size": 1897856,
  "upload_date": "2018-07-31T11:42:36.873274+00:00"
  "platforms": [
    "winxpx86",
    "win7x86",
    "win7x64",
    "win10x64_1903"
  ]
}
```

8.4.12. Session

The **Session** object contains data about a session.

Structure

Key	Type	Description
api_key	string	API key.
start_date	string	Date and time the session was started.

Example

```
{
    "api_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "start_date": "2018-12-20T08:55:35.158344+00:00"
}
```

8.4.13. Task

The **Task** object contains data about a task. **Task** is a file analysis on a single platform. Task can contain a various set of keys: **TaskBasic**, **TaskFinished**, or **TaskProcessing**.

TaskBasic

The **TaskBasic** object contains general information about a task. Such an object with the basic set of keys is used in a list of Analysis objects.

Structure

Key	Type	Description
id	integer	Task ID.
status	string	Task status. Available values: in queue, failed, processing, deleted, successful.
platform_code	string	Platform.code.
start_date	string (datetime.iso8601)	Date and time the task was started.
end_date	string/null (datetime.iso8601)	Date and time the task was completed.
maliciousness	integer/null	Maliciousness, from 0 to 100.

Example

```
{
    "id": 20,
    "status": "failed",
    "platform_code": "winxpx86",
    "start_date": "2018-07-30T16:54:07.156371",
    "end_date": "2018-07-30T16:55:07.156371",
    "maliciousness": null
}
```

TaskFinished

The **TaskFinished** object contains the keys of the **TaskBasic** object and analysis results for the specified platform.

Structure

Key		Type	Description
`detects`		string[]	A list of detected threats. The list can include the following strings: `yara`: a YARA rule has triggered; `behavior`: malicious or suspicious behavior for a file has been detected; `files_dumps`: the threats have been detected in created files or/and memory dumps.
`end_date`		string/null (datetime.iso8601)	Date and time the task was completed.
`id`		integer	Task ID.
`maliciousness`		integer/null	Maliciousness, from 0 to 100.
`platform_code`		string	Platform.code.
`rules`		object/null	List of triggered rules.
	`malicious`	string[]	List of rules that have been triggered due to malicious activity of the sample.
	`neutral`	string[]	List of rules that have been triggered due to neutral activity of the sample.
	`suspicious`	string[]	List of rules that have been triggered due to suspicious activity of the sample.
`sample_detect`		string/null	Name of the threat detected using signature databases.

Key		Type	Description
`start_date`		string (datetime.iso8601)	Date and time the task was started.
`status`		string	The current status of the task. Available values: `in queue`, `failed`, `processing`, `deleted`, `successful`.
`tags`		string[]	The list of tags retrieved from the triggered YARA rules.
`verdict`		string	Overall result of the file maliciousness corresponding to one of three categories. The higher number corresponds to the higher level of the maliciousness probability. Available values: `none`, `clean1`, `clean2`, `suspicious1`, `suspicious2`, `malware1`, `malware2`.
`yara_rules`		object[]	List of triggered YARA rules.
	`name`	string	The name of the YARA rule
	`rule_type`	string	The type of the YARA rule. Available values: `user` (a user-defined rule) and `system` (a system-defined rule).
	`severity`	string	The file behavior category. When adding a YARA rule, you should specify the behavior category that will be assigned to the sample if the YARA rule is triggered. The specified category appears in the `severity` field. Available values: `neutral`, `suspicious`, `malware`. More about adding a YARA rule...

Example

```
{
  "id": 16916,
  "status": "successful",
  "maliciousness": 100,
  "platform_code": "winxpx86",
  "start_date": "2018-12-12T11:29:44.645968+00:00",
  "end_date": "2018-12-12T11:33:37.490050+00:00",
  "verdict": "malware2",
  "rules": null,
  "detects": [
     "files_dumps"
   ],
  "platform_code": "win7x64"
}
```

TaskProcessing

TaskFinished contains the keys of the **TaskBasic** object and data about the analysis process.

Structure

Key	Type	Description
`end_date`	string	Date and time the task was completed.
`id`	integer	Task ID.
`maliciousness`	integer/null	Maliciousness, from 0 to 100.
`message`	string/null	Message about the task progress.
`platform_code`	string	Platform.code.
`progress`	integer	Task progress, in percent.
`start_date`	string (datetime.iso8601)	Date and time the task was started.
`status`	string	The current status of the task. Available values: `in queue`, `failed`, `processing`, `deleted`, `successful`.

Example

```
{
  "id": 18656,
  "status": "processing",
  "maliciousness": null,
  "platform_code": "win7x86",
  "start_date": "2019-02-07T09:39:11.517117+00:00",
  "end_date": null,
  "message": "Waiting while the file is running (60 sec)…",
  "progress": 19
}
```

8.5. Examples

This section provides examples of how to work with Dr.Web vxCube using an API.

You will learn how to:

- Get an API key
- Upload a sample to the Dr.Web vxCube server
- Start the analysis
- Get information about the analysis

- Download the report

8.5.1. Get an API Key

To get an API key, send the POST login request with the login and password:

Get API key created earlier

To get one of the created API keys, specify the parameter value `new_key: false` or just do not specify the parameter:

```
curl -X POST https://<IP address/domain name of the server>/api-2.0/login \
-H "Content-Type: application/json" \
-d "{\"login\":\"example@drweb.com\", \"password\":\"secret_password\"}"
```

You receive a response with the API key (the API key is required to be specified in the header of each subsequent request):

```
{
    "new_key": false,
    "api_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
    "start_date": "2019-02-08T04:08:15.162342+00:00"
}
```

Create API key

To create a new API key, specify the parameter value `new_key: true` (if you have not got any API keys created, you do not have to specify the parameter—API key will be created anyway):

```
curl -X POST https://<IP address/domain name of the server>/api-2.0/login \
-H "Content-Type: application/json" \
-d "{\"login\":\"example@drweb.com\", \"password\":\"secret_password\", \"new_key\":
true, \"name\":\"example_name_api\"}"
```

You receive a response with the API key (the API key is required to be specified in the header of each subsequent request):

```
{
   "new_key": true,
   "api_key": "bbbbbbbb-cccc-dddd-eeee-ffffffffffff",
   "start_date": "2019-03-08T04:08:15.162342+00:00",
   "name": "example_name_api"
}
```

8.5.2. Upload a Sample to the Dr.Web vxCube Server

To upload a sample to the server, send the POST samples request:

```
curl -X POST https://<IP address/domain name of the server>/api-2.0/samples \
-F "file=@testfile.pdf" \
-F "password="vxcube"" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
```

In response, you receive the Sample object that contains data about the uploaded file including the file format identified automatically and a list of supported platforms. Use the received data for the further file analysis.

Response:

```
{
    "id": 6784,
    "size": 10881846,
    "name": "testfile.pdf",
    "is_x64": null,
    "format_name": "pdf",
    "upload_date": "2019-02-08T04:08:15.162343+00:00",
    "md5": "34fb8ae3c01653985085ee7e2f749ea5",
    "sha1": "00a610100a3516f4d0daa33e7de317d2ddb6c2c6",
    "sha256": "11bd131be00cbe1c43b4444ec4300dc7651805ea36393b1cca1675983dc275bc",
    "platforms": [
        "acrobat_xp_10",
        "acrobat_7_32_11",
        "acrobat_7_64_15",
        "acrobat_10_64_15"
    ]
}
```

8.5.3. Start the Analysis

To start the analysis of a sample, send the POST analyses request:

```
curl -X POST https://<IP address/domain name of the server>/api-2.0/analyses \
-H "Content-Type: application/json" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee" \
-d "{\"sample_id\":\"6784\", \"platforms\":[\"acrobat_7_32_11\", \"acrobat_7_64_15\"]}"
```

In the request, the uploaded file ID and the list of platforms are specified. The values are taken from the response for the previous request.

To start the analysis of a sample using network traffic redirection, send the POST analyses request:

```
curl -X POST https://<IP address/domain name of the server>/api-2.0/analyses \
-H "Content-Type: application/json" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee" \
-d "{\"sample_id\":\"6784\", \"platforms\":[\"acrobat_7_32_11\", \"acrobat_7_64_15\",
\"net\": \"socks5://username:password@<proxyaddress>:1080?udp=on\"}"]}"
```

In response, you receive the Analysis object that contains general analysis information:

```
{
    "id": 6260,
    "sample_id": 6784,
    "size": 10881846,
    "sha1": "00a610100a3516f4d0daa33e7de317d2ddb6c2c6",
    "start_date": "2019-02-08T04:08:15.162343+00:00",
    "format_name": "pdf",
    "user_name": "example@drweb.com",
    "tasks": [{
        "message": null,
        "end_date": null,
        "platform_code": "acrobat_7_64_15",
        "maliciousness": null,
        "progress": 0,
        "id": 18676,
        "status": "in queue",
        "start_date": "2019-02-08T04:08:15.643122+00:00"
    }, {
        "message": null,
        "end_date": null,
        "platform_code": "acrobat_7_32_11",
        "maliciousness": null,
        "progress": 0,
        "id": 18675,
        "status": "in queue",
        "start_date": "2019-02-08T04:08:15.632924+00:00"
    }]
}
```

8.5.4. Get Information About the Analysis

To get detailed information about analysis, wait for the analysis to finish, and then send the GET analyses/<analysis_id:uuid> request. In the request, specify the analysis ID:

```
curl -X GET https://<IP address/domain name of the server>/api-2.0/analyses/60e21c98-
7c2a-4112-81b5-a577f6cdf4db \
-H "Content-Type: application/json" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee"
```

In response, you receive the Analysis object:

```
{
    "id": "111ba12c5-d330-40eb-b988-fa16402ee111",
    "sha1": "9e92e9408afdf75fc3dea5e457cb0c70728f74ce",
    "sample_id": 77236,
    "size": 156160,
    "format_name": "dll",
    "start_date": "2024-02-13T14:28:08.871359",
    "user_name": "test@test.com",
    "tasks": [
        {
            "id": 235182,
            "status": "successful",
            "platform_code": "win10x64_2004",
            "start_date": "2024-02-13T14:28:09.135345",
            "end_date": "2024-02-13T14:30:46.776797",
            "maliciousness": 94,
            "verdict": "malware2",
            "detects": [
                "yara"
            ],
            "sample_detect": null,
            "rules": {
                "neutral": [
                    "Creating synchronization primitives",
                    "Searching for synchronization primitives"
                ]
            },
            "yara_rules": [
                {
                    "name": "gozi3",
                    "severity": "malware",
                    "rule_type": "system"
                },
                {
                    "name": "gozi",
                    "severity": "malware",
                    "rule_type": "system"
                }
            ],
            "tags": [
                "GOZI3",
                "GOZI"
            ]
        }
    ]
}
```

8.5.5. Download a Report

To download an archived analysis report, send the GET analyses/<analysis_id:uuid>/archive:

```
curl -X GET https://<IP address/domain name of the server>/api-2.0/analyses/40e2fc98-
1c2a-4112-81b5-a57df2cd22db/archive \
-H "Content-Type: application/json" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee" \
-o <output_archive>
```

To download one of the report files, send the GET tasks/<task_id:number>/archive_storage request. A request example for downloading the PCAP file:

```
curl -X GET https://<IP address/domain name of the server>/api-
2.0/tasks/18681/archive_storage \
-H "Content-Type: application/json" \
-H "Authorization: api-key aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee" \
-d "{\"path\": \"network.pcap\"}" \
-o some_file
```

9. Technical Support

If you encounter any issues using Dr.Web vxCube, you can contact the Doctor Web technical support in the following ways:

- Fill in the web form: https://support.drweb.com/support_wizard/vxcube.
- Call by phone in Moscow: +7 (495) 789-45-86. Free phone call (within Russia): 8-800-333-7932.

Refer to the official website at https://company.drweb.com/contacts/offices/ for regional and international office information of Doctor Web company.

10. Appendix A. List of Software on Virtual Machines

Windows XP x86

- Microsoft Office Enterprise 2007 x86 (optional)
- Adobe Acrobat Reader 10.1.0
- Adobe Flash 12.0.0.77
- JAVA 6u45
- Adobe Flash Standalone 10.3.181.23 (%windir%\flash_sa.exe)
- Mozilla Firefox 52.0.2
- Opera 35.0
- Google Chrome 44.0.2403.155
- ICQ 8.3 build 7317
- QIP 2012 4.0.9380
- Pidgin 2.10.11
- Mozilla Thunderbird 31.7.0
- Visual C++ Redistributable 2005 x86
- Visual C++ Redistributable 2008 x86
- Visual C++ Redistributable 2010 x86
- Visual C++ Redistributable 2012 x86
- Visual C++ Redistributable 2013 x86
- Visual C++ Redistributable 2015 x86
- .NET Framework 1.1
- .NET Framework SDK (msvcp70.dll, msvcr70.dll)
- .NET Framework 2.0 Service Pack SP2
- .NET Framework 3.0 Service Pack SP2
- .NET Framework 3.5 Service Pack SP1
- .NET Framework 4.0
- Steam 2.91
- WinRAR 5.20 x86
- Telegram Desktop 1.2.17
- mIRC 7.43

Windows 7 x86

- Adobe Acrobat Reader 11.0.1
- Microsoft Office Professional Plus 2010 x86 (optional)
- Adobe Flash 12.0.0.77

- Adobe Flash ActiveX 17.0.0.188
- JAVA 7u11
- Adobe Flash Standalone 11.1.102.62 (%windir%\flash_sa.exe)
- Mozilla Firefox 68.0.2
- Opera 33.0.1990.115
- Google Chrome 43.0.2357.65
- ICQ 8.3 build 7317
- QIP 2012 4.0.9380
- Pidgin 2.10.11
- Mozilla Thunderbird 31.7.0я
- Visual C++ Redistributable 2005 x86
- Visual C++ Redistributable 2008 x86
- Visual C++ Redistributable 2010 x86
- Visual C++ Redistributable 2012 x86
- Visual C++ Redistributable 2013 x86
- Visual C++ Redistributable 2015 x86
- .NET Framework 1.1
- .NET Framework SDK (msvcp70.dll, msvcr70.dll)
- .NET Framework 4.8
- Steam 3.17
- .NET Framework 4.7.1
- Telegram Desktop 1.2.17
- WinRAR 5.20 x86

Windows 7 x64

- Adobe Acrobat Reader Document Cloud 15.10.20056
- Microsoft Office Professional Plus 2010 x64 (optional)
- Adobe Flash 18.0.0.261
- Adobe Flash ActiveX 19.0.0.207
- JAVA 8u45 x64
- Adobe Flash Standalone 19.0.0.226 (%windir%\flash_sa.exe)
- K-Lite Mega Codec Pack 11.1.0
- Mozilla Firefox 78.0.2
- Opera 29.0.1795.47
- Google Chrome 42.0.2311.135
- ICQ 8.3 build 7317

- Mail.Ru Agent 6.4 build 8614
- QIP 2012 4.0.9380
- Pidgin 2.10.11
- Total Commander 8.51a x64
- Mozilla Thunderbird 31.6.0
- Winamp 5.666
- Visual C++ Redistributable 2005 x86
- Visual C++ Redistributable 2008 x86
- Visual C++ Redistributable 2010 x86
- Visual C++ Redistributable 2012 x86
- Visual C++ Redistributable 2013 x86
- Visual C++ Redistributable 2015 x86
- Visual C++ Redistributable 2005 x64
- Visual C++ Redistributable 2008 x64
- Visual C++ Redistributable 2010 x64
- Visual C++ Redistributable 2012 x64
- Visual C++ Redistributable 2013 x64
- Visual C++ Redistributable 2015 x64
- .NET Framework 1.1
- .NET Framework SDK (msvcp70.dll, msvcr70.dll)
- .NET Framework 4.8
- Steam 3.17
- Telegram Desktop 1.4.3
- .NET Framework 4.7.1
- WinRAR 5.3 x64
- mIRC 7.41

Windows 10 x64

- Adobe Acrobat Reader Document Cloud 2015.010.20060
- Adobe Flash 21.0.0.197
- Adobe Flash ActiveX 21.0.0.197
- Microsoft Office Professional Plus 2016 x86 (optional)
- JAVA 8u77 x64
- Adobe Flash Standalone 19.0.0.226 (%windir%\flash_sa.exe)
- Mozilla Firefox 91.0.2 x64
- Opera 36.0.2130.46

- Google Chrome 47.0.2526.80
- ICQ 8.3 build 7317
- QIP 2012 4.0.9380
- Pidgin 2.10.11
- Mozilla Thunderbird 38.7.1
- Visual C++ Redistributable 2005 x86
- Visual C++ Redistributable 2008 x86
- Visual C++ Redistributable 2010 x86
- Visual C++ Redistributable 2012 x86
- Visual C++ Redistributable 2013 x86
- Visual C++ Redistributable 2017 x86
- Visual C++ Redistributable 2005 x64
- Visual C++ Redistributable 2010 x64
- Visual C++ Redistributable 2012 x64
- Visual C++ Redistributable 2013 x64
- Visual C++ Redistributable 2017 x64
- .NET Framework 4.6.2
- Steam 3.37
- Telegram Desktop 1.4.3
- mIRC 7.43
- WinRAR 5.31 x64

Android 7.1

- Android Keyboard (AOSP) 7.1.2
- Calculator 7.1.2
- Calendar 7.1.2
- Camera 2.0.002
- Clock 4.5.0
- Contacts 1.4.22
- Dev Tools 1.0
- Email 7.1.2
- Files 7.1.2
- Gallery 1.1.40030
- Google Play 31.6.13-21
- Google Play Games 2022.01.32371
- Google Play Services 22.09.20

- Launcher3 7.1.2
- Messaging 1.0.001
- Music 3.0
- NotePad 7.1.2
- Phone 3.00.00
- RSS Reader 7.1.2
- Search 7.1.2
- Settings 7.1.2
- Terminal Emulator 1.0.70
- WebView Shell 1.0

Astra SE 1.7 (Voronezh)

- Standard software

Astra CE 2.12 (Orel)

- Standard software

Debian 8 (Jessie) ARMel 32-bit

- Standard software

Debian 8 (Jessie) PowerPC 32-bit

- Standard software

Debian 10 (Buster) MIPS 32-bit

- Standard software

Debian 11 (Bullseye) ARM 64-bit

- Standard software

Debian 11 (Bullseye) ARMhf 32-bit

- Standard software

Debian 11 (Bullseye) Intel 32-bit

- Standard software

Debian 11 (Bullseye) Intel 64-bit

- Standard software

Debian 11 (Bullseye) MIPSel 32-bit

- Standard software

Debian 11 (Bullseye) MIPSel 64-bit

- Standard software

Debian 11 (Bullseye) PowerPCel 64-bit

- Standard software

11. Appendix B. Functions of the dr_sandbox Module

Functions for the Android sandbox (the 'andr' category)

- archive_files
- certificate_sha1
- dynamic
  - created_files
    - path
    - sha1
  - crypto_dumps
  - downloaders
    - detect
    - sha1
  - downloads
    - detect
    - sha1
    - url
  - droppers
    - detect
    - sha1
  - dumps
    - detect
    - path
    - sha1
  - executed_commands
  - flags

  - phone_calls
  - sms
    - message
    - number
  - urls
- manifest
  - activities
  - app_name
  - filters
  - home_activity
  - is_firmware
  - main_activity
  - meta_data
    - name
    - resource
    - value
  - package
  - permissions
  - receivers
  - services
  - strings_resources
  - version_code
  - version_name
- resources_digests
- sha1

source_host

Functions for the Windows sandbox (the 'descr_tech' category)

Enabling autorun and distribution (the 'autorun' category)

Changes executable system files (change_system_executable_files)

Creates files on removable media (create_files_on_removable_media)

Creates or modifies files (create_or_modify_files)

Creates services (create_services)

Infects executable files (infect_executables)

Modifies the master boot record (modify_mbr)

Modifies registry keys (modify_registry)

Substitutes executable system files (replace_system_executable_files)

Modifies a file system (the 'filesystem' category)

Changes file extensions in user data (change_user_data_extensions)

Creates files (create_files)

Creates files and demands payment for file decoding (create_ransom_message_files)

Modifies a HOSTS file (modify_hosts)

Modifies user data files (modify_user_data_files)

Moves files (move_files)

Moves itself (move_self)

Moves system files (move_system_files)

Deletes files (remove_files)

Deletes itself (remove_self)

Assigns the 'hidden' attribute to files (set_hidden)

Substitutes executable files (substitute_executables)

Substitutes files (substitute_files)

Replaces a HOSTS file (substitute_hosts)

Malicious functions (the 'malicious' category)

Adds an anti-virus exclusion (add_antivirus_exclusion)

Blocks Command Prompt (block_cmd)

Blocks Registry Editor (block_regedit)

Blocks System File Checker (block_system_file_checker)

Blocks System Restore (block_system_restore)

Blocks Windows Task Manager (block_taskmgr)

Blocks User Account Control (block_user_account_control)

Blocks Windows Action Center (block_windows_action_center)

Blocks Windows Defender (block_windows_defender)

Blocks Windows File Protection (block_windows_file_protection)

Blocks Windows Firewall (block_windows_firewall)

Blocks Windows Security Center (block_windows_security_center)

Blocks Windows Update (block_windows_updates)

Brute forces passwords of OS accounts (bruteforce_os_accounts)

Creates and executes processes (create_and_exec)

Creates an onion service (create_onion_service)

Removes shadow copies of volumes (delete_volume_shadow_copies)

Searches for windows to detect virtual machines (detect_virtual_machine)

Disables AMSI (disable_amsi)

Downloads and executes (downloads_and_executes)

Downloads and executes files (downloads_and_executes_files)

Downloads the files (download_file)

Downloads files (download_files)

Executes (exec)

Executes WMI operations (exec_wmi)

Creates and executes files (an exploit) (exploit_create_and_exec)

Creates and loads libraries (an exploit) (exploit_create_and_load_library)

Executes the following (an exploit) (exploit_exec)

Forces autorun for removable media (force_autorun_for_removable_media)

Forces the system to hide file extensions from view (hide_from_view_file_extensions)

Forces the system to hide hidden files from view (hide_from_view_hidden_files)

Hides processes (hide_processes)

Disables taskbar notifications (hide_taskbar_notifications)

Hooks functions in browsers (hook_in_browser)

Installs hooks to intercept notifications on keystrokes for all processes (hook_keyboard_all_processes)

Installs hooks to intercept notifications on keystrokes for specific processes (hook_keyboard_concrete_processes)

Installs hooks to intercept notifications on window messages (hook_keyboard_on_window_messages)

Injects code into numerous user processes (inject_to_a_lot_of_user_processes)

Injects code into system processes (inject_to_system_proc)

Injects code into user processes (inject_to_user_proc)

Modifies settings of Windows Explorer (modify_explorer_settings)

Modifies settings of Windows Internet Explorer (modify_ie_settings)

Removes or modifies registry (modify_registry_to_bypass_firewall)

Changes the DNS server (modify_system_dns)

Modifies system settings (modify_system_settings)

Reads files that store third party passwords (read_third_party_passwords)

Registers BHO (register_bho)

Registers a COM server (register_com_server)

Registers file system filter (register_filesystem_filter)

Restores hooked functions in SSDT (restore_ssdt_hooks)

Searches for registry branches that store third party passwords (search_password_in_registry)

Searches for windows to detect analytical utilities (search_wnd_for_analyzing_soft)

Searches for windows to detect programs and games (search_wnd_for_programs_and_games)

Searches for windows to bypass anti-viruses (search_wnd_to_bypass_av)

Searches for windows to bypass WFP system (search_wnd_to_bypass_wfp)

Hook specific functions in SSDT (set_concrete_ssdt_hooks)

Sets a home page for Windows IE (set_homepage_for_ie)

Hook specific functions in SSDT (set_ssdt_hooks)

Terminates numerous user processes (try_to_terminate_a_lot_of_user_processes)

Terminates system processes (try_to_terminate_system_processes)

Terminates user processes (try_to_terminate_user_processes)

Miscellaneous (the 'miscellaneous' category)

Adds a root certificate (add_root_certificate)

Creates and executes (create_and_exec)

Disables a certificate (disable_certificate)

Executes (exec)

Loads drivers (load_driver)

Changes the AutoConfigURL parameter (modify_auto_config_url)

Searches for windows (search_wnd)

Attempts to shut down Windows OS (shut_down_windows)

Uses NTFS alternate data streams (use_ntfs_data_streams)

Network activity (the 'network' category)

Connects to the following (connect_to)

TCP requests (tcp)

HTTP GET requests using TCP (tcp_http_get)

HTTP POST requests using TCP (tcp_http_post)

Unknown HTTP requests using TCP (tcp_http_unk)

UDP requests (udp)

Functions for the Linux sandbox (the 'descr_tech_lbcl' category)

Enabling autorun and distribution ('autorun' category)

Creates or modifies files (create_or_modify_files)

Creates or modifies symbolic links (create_or_modify_symlinks)

Modifies a file system (the 'filesystem' category)

Changes the time when the file was created, accessed, or modified (change_time_of_file)

Creates directories (create_dir)

Creates or modifies files (create_or_modify_file)

Creates symbolic links (create_symlink)

Blocks files (lock_file)

Changes file access rights (modify_file_access_rights)

Changes file owner (modify_file_owner)

Mounts file systems (mount_file_system)

Deletes directories (remove_dir)

Deletes files (remove_files)

Unmounts file systems (unmount_file_system)

Malicious functions ('malicious' category)

Tries to kill system processes (attempt_kill_system_proc)

Tries to kill analyzers (attept_kill_analyzers)

Tries to kill processes (attept_kill_proc)

Compiles source code (compile_program_from_source_codes)

Gains root access (gain_root_privileges)

Accesses SSH keys (get_access_to_ssh_keys)

Injects itself in processes (inject_to_proc)

Kills analyzers (kill_analyzers)

Kills processes (kill_proc)

Kills system processes (kill_system_proc)

Launches itself as a daemon (launch_itself_as_daemon)

Launches processes (launch_processes)

Manages services (manage_services)

Changes firewall settings (modify_firewall_settings)

Changes router settings (modify_router_settings)

Operates kernel modules (operate_kernel_modules)

Performs process tracing (perform_process_tracing)

Deletes itself (remove_self)

Deletes system files (remove_system_files)

Replaces system files (replace_system_files)

Stops system services (stops_system_services)

Substitutes an application name (substitute_application_name_for)

Network activity (the 'network' category)

Performs a bruteforce attack via the SSH protocol (attack_bruteforce_via_ssh)

Performs a bruteforce attack via the TELNET protocol (attack_bruteforce_via_telnet)

Performs a bruteforce attack via the undefined protocol (attack_bruteforce_via_unk_protocol)

Connects to servers (connect_to)

Connects to servers over the IRC protocol (connect_to)

DNS requests (dns_ask)

HTTP GET requests (http_get)

Other HTTP requests (http_other)

HTTP POST requests (http_post)

Awaits incoming connections on ports (listening_port)

Receives data from servers (receive_data_from_server)

Sends data to servers (send_data_to_server)

Other (the 'other' category)

Collects information about the CPU (collect_cpu_info)

Collects information about the network activity (collect_network_info)

Collects information about the OS (collect_os_info)

Collects information about RAM (collect_ram_info)

Reads information from /proc/kallsyms (read_info_from_proc_kallsyms)

Detects (the 'detects' category)

All detects (all_detects_here)

Detects of alloc files (detects_of_allocs)

Detects of drops (detects_of_drops)

Detects of dumps (detects_of_dumps)

Detects of injects (detects_of_injects)

Detects of src files (detects_of_src)

Checks the buffer with the offset (check_buffer)

Checks the byte with the offset (check_byte)

Checks DWORD with the offset (check_dword)

Checks WORD with the offset (check_word)

Searches for the case-insensitive ASCII or wide string (ci_any)

Searches for the case-insensitive ASCII string (ci_ascii)

Searches for the case-insensitive wide string (ci_wide)

Searches for the case-insensitive XORed string (ci_wide)

Calculates the crc32 hash of the buffer (crc32)

Searches for the case-sensitive ASCII or wide string (cs_any)

Searches for the case-sensitive ASCII string (cs_ascii)

Searches for the case-sensitive wide string (cs_wide)

Returns detects for the file (detects_of_this_file)

Searches for the file name (filename)

Searches for the file name using boost::regex (filename_boost_regex)

Searches for actions with a file system (filesystem_access)

Searches for a network activity (network_access)

Searches for actions with a registry (registry_access)

Returns a file type (sb_filetype)

Searches for the substring in the buffer (search_substring_in_range)

The descriptions of the functions for the Android sandbox (the 'andr' category)

Function	Result	Examples
`archive_file(regex)`	The list of files that are included in APK and match the pattern: ARCHIVE_FILES_PATTERN = ['.dll', '.js', '.html', '.so'].	`dr_sandbox.andr.archive_files(/pattern/)`
`archive_file_num`	The list of files that are included in APK and match the pattern: ARCHIVE_FILES_PATTERN = ['.dll', '.js', '.html', '.so'].	`dr_sandbox.andr.archive_files_num`
`certificate_sha1(regex)`	The SHA1 hash of the certificate that an app is signed with.	`dr_sandbox.andr.certificate_sha1(/pattern/)`
`certificate_sha1_num`	The SHA1 hash of the certificate that an app is signed with.	`dr_sandbox.andr.certificate_sha1_num`
The **`dynamic`** `subcategory`		
`created_files.path(regex)`	Created files: a path.	`dr_sandbox.andr.`**`dynamic`**`.created_files.path(/pattern/)`
`created_files.path_num`	Created files: a path.	`dr_sandbox.andr.`**`dynamic`**`.created_files.path_num`
`created_files.sha1(regex)`	Created files: SHA1.	`dr_sandbox.andr.`**`dynamic`**`.created_files.sha1(/pattern/)`
`created_files.sha1_num`	Created files: SHA1.	`dr_sandbox.andr.`**`dynamic`**`.created_files.sha1_num`
`crypto_dumps(regex)`	Encrypted dumps.	`dr_sandbox.andr.`**`dynamic`**`.crypto_dumps(/pattern/)`
`crypto_dumps_num`	Encrypted dumps.	`dr_sandbox.andr.`**`dynamic`**`.crypto_dumps_num`
`downloaders.detect(regex)`	The list of samples that download an analyzed sample.	`dr_sandbox.andr.`**`dynamic`**`.downloaders.detect(/pattern/)`

Function	Result	Examples
downloaders.detect_num	The list of samples that download an analyzed sample.	dr_sandbox.andr.**dynamic**.downloaders.detect_num
downloaders.sha1(regex)	The list of samples that download an analyzed sample.	dr_sandbox.andr.**dynamic**.downloaders.sha1(/pattern/)
downloaders.sha1_num	The list of samples that download an analyzed sample.	dr_sandbox.andr.**dynamic**.downloaders.sha1_num
downloads.detect(regex)	The downloaded payload (apk/dex).	dr_sandbox.andr.**dynamic**.downloads.detect(/pattern/)
downloads.detect_num	The downloaded payload (apk/dex).	dr_sandbox.andr.**dynamic**.downloads.detect_num
downloads.sha1(regex)	The downloaded payload (apk/dex).	dr_sandbox.andr.**dynamic**.downloads.sha1(/pattern/)
downloads.sha1_num	The downloaded payload (apk/dex).	dr_sandbox.andr.**dynamic**.downloads.sha1_num
downloads.url(regex)	The downloaded payload (apk/dex).	dr_sandbox.andr.**dynamic**.downloads.url(/pattern/)
downloads.url_num	The downloaded payload (apk/dex).	dr_sandbox.andr.**dynamic**.downloads.url_num
droppers.detect(regex)	The list of samples that upload an analyzed sample.	dr_sandbox.andr.**dynamic**.droppers.detect(/pattern/)
droppers.detect_num	The list of samples that upload an analyzed sample.	dr_sandbox.andr.**dynamic**.droppers.detect_num
droppers.sha1(regex)	The list of samples that upload an analyzed sample.	dr_sandbox.andr.**dynamic**.droppers.sha1(/pattern/)
droppers.sha1_num	The list of samples that upload an analyzed sample.	dr_sandbox.andr.**dynamic**.droppers.sha1_num
dumps.detect(regex)	The payload dump: a detect.	dr_sandbox.andr.**dynamic**.dumps.detect(/pattern/)

Function	Result	Examples
`dumps.detect_num`	The payload dump: a detect.	`dr_sandbox.andr.`**`dynamic`**`.dumps.detect_num`
`dumps.path(regex)`	The payload dump: a path.	`dr_sandbox.andr.`**`dynamic`**`.dumps.path(/pattern/)`
`dumps.path_num`	The payload dump: a path.	`dr_sandbox.andr.`**`dynamic`**`.dumps.path_num`
`dumps.sha1(regex)`	The payload dump: a SHA1 hash.	`dr_sandbox.andr.`**`dynamic`**`.dumps.sha1(/pattern/)`
`dumps.sha1_num`	The payload dump: a SHA1 hash.	`dr_sandbox.andr.`**`dynamic`**`.dumps.sha1_num`
`executed_commands(regex)`	Executed shell commands.	`dr_sandbox.andr.`**`dynamic`**`.executed_commands(/pattern/)`
`executed_commands_num`	Executed shell commands.	`dr_sandbox.andr.`**`dynamic`**`.executed_commands_num`
`flags(regex)`	Behavior flags.	`dr_sandbox.andr.`**`dynamic`**`.flags(/pattern/)`
`flags_num`	Behavior flags.	`dr_sandbox.andr.`**`dynamic`**`.flags_num`
`phone_calls(regex)`	Phone calls.	`dr_sandbox.andr.`**`dynamic`**`.phone_calls(/pattern/)`
`phone_calls_num`	Phone calls.	`dr_sandbox.andr.`**`dynamic`**`.phone_calls_num`
`sms.message(regex)`	Sent SMS: a message content.	`dr_sandbox.andr.`**`dynamic`**`.sms.message(/pattern/)`
`sms.message_num`	Sent SMS: a message content.	`dr_sandbox.andr.`**`dynamic`**`.sms.message_num`
`sms.number(regex)`	Sent SMS: a phone number.	`dr_sandbox.andr.`**`dynamic`**`.sms.number(/pattern/)`
`sms.number_num`	Sent SMS: a phone number.	`dr_sandbox.andr.`**`dynamic`**`.sms.number_num`
`urls(regex)`	Found URLs. Only the URLs that match the regular expression are counted.	`dr_sandbox.andr.`**`dynamic`**`.urls(/pattern/)`

Function	Result	Examples
`urls_num`	Found URLs.	`dr_sandbox.andr.`**`dynamic`**`.urls_num`
The **`manifest`** `subcategory`		
`activities(regex)`	The list of app activities (screens).	`dr_sandbox.andr.`**`manifest`**`.activities(/pattern/)`
`activities_num`	The list of all app activities (screens).	`dr_sandbox.andr.`**`manifest`**`.activities_num`
`app_name(regex)`	The app name on the device.	`dr_sandbox.andr.`**`manifest`**`.app_name(/pattern/)`
`app_name_num`	The app name on the device.	`dr_sandbox.andr.`**`manifest`**`.app_name_num`
`filters(regex)`	The list of actions from the manifest.	`dr_sandbox.andr.`**`manifest`**`.filters(/pattern/)`
`filters_num`	The list of actions from the manifest.	`dr_sandbox.andr.`**`manifest`**`.filters_num`
`home_activity(regex)`	Activity, the app entry point.	`dr_sandbox.andr.`**`manifest`**`.home_activity(/pattern/)`
`home_activity_num`	Activity, the app entry point.	`dr_sandbox.andr.`**`manifest`**`.home_activity_num`
`is_firmware(regex)`	Is app from firmware or not.	`dr_sandbox.andr.`**`manifest`**`.is_firmware(/pattern/)`
`is_firmware_num`	Is app from firmware or not.	`dr_sandbox.andr.`**`manifest`**`.is_firmware_num`
`main_activity(regex)`	Main activity, the app entry point.	`dr_sandbox.andr.`**`manifest`**`.main_activity(/pattern/)`
`main_activity_num`	Main activity, the app entry point.	`dr_sandbox.andr.`**`manifest`**`.main_activity_num`
`meta_data.name(regex)`	Metadata: the name.	`dr_sandbox.andr.`**`manifest`**`.meta_data.name(/pattern/)`
`meta_data.name_num`	Metadata: the name.	`dr_sandbox.andr.`**`manifest`**`.meta_data.name_num`
`meta_data.resour`	Metadata: the resource.	`dr_sandbox.andr.`**`manifest`**`.meta_data.resource(/pattern/)`

Function	Result	Examples
ce(regex)		
meta_data.resource_num	Metadata: the resource.	dr_sandbox.andr.**manifest**.meta_data.resource_num
meta_data.value(regex)	Metadata: the value.	dr_sandbox.andr.**manifest**.meta_data.value(/pattern/)
meta_data.value_num	Metadata: the value.	dr_sandbox.andr.**manifest**.meta_data.value_num
package(regex)	The app package name.	dr_sandbox.andr.**manifest**.package(/pattern/)
package_num	The app package name.	dr_sandbox.andr.**manifest**.package_num
permissions(regex)	The list of permissions that the app needs.	dr_sandbox.andr.**manifest**.permissions(/pattern/)
permissions_num	The list of permissions that the app needs.	dr_sandbox.andr.**manifest**.permissions_num
receivers(regex)	The list of broadcast receivers.	dr_sandbox.andr.**manifest**.receivers(/pattern/)
receivers_num	The list of broadcast receivers.	dr_sandbox.andr.**manifest**.receivers_num
services(regex)	The list of app services.	dr_sandbox.andr.**manifest**.services(/pattern/)
services_num	The list of app services.	dr_sandbox.andr.**manifest**.services_num
strings_resources(regex)	The list of all string resources.	dr_sandbox.andr.**manifest**.strings_resources(/pattern/)
strings_resources_num	The list of all string resources.	dr_sandbox.andr.**manifest**.strings_resources_num
version_code(regex)	The version code.	dr_sandbox.andr.**manifest**.version_code(/pattern/)
version_code_num	The version code.	dr_sandbox.andr.**manifest**.version_code_num
version_name(regex)	The version name.	dr_sandbox.andr.**manifest**.version_name(/pattern/)

Function	Result	Examples
`version_name_num`	The version name.	`dr_sandbox.andr.`**`manifest`**`.version_name_num`
`resources_digests(regex)`	The list of SHA1-Digest for APK resource files.	`dr_sandbox.andr.resources_digests(/pattern/)`
`resources_digests_num`	The list of SHA1-Digest for APK resource files.	`dr_sandbox.andr.resources_digests_num`
`sha1(regex)`	SHA1 of the sample.	`dr_sandbox.andr.sha1(/pattern/)`
`sha1_num`	SHA1 of the sample.	`dr_sandbox.andr.sha1_num`
`source_host(regex)`	The sample source.	`dr_sandbox.andr.source_host(/pattern/)`
`source_host_num`	The sample source.	`dr_sandbox.andr.source_host_num`

The descriptions of the functions for the Windows sandbox (the 'descr_tech' category)

Enabling autorun and distribution (the 'autorun' category)

Function	Result	Event type	Examples
`change_system_executable_files(regex)`	Returns the number of events of a specific type.	Changes executable system files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.change_system_executable_files(/beep.sys/)`
`change_system_executable_files_num`	Returns the amount of events of a certain type.	Changes executable system files.	`dr_sandbox.descr_tech.autorun.change_system_executable_files_num > 0`
`create_files_on_removable_media(regex)`	Returns the number of events of a specific type.	Creates files on removable media. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.create_files_on_removable_media(/10thingscondoms.pdf/)`
`create_files_on_removable_media_num`	Returns the number of events of a specific type.	Creates files on removable media.	`dr_sandbox.descr_tech.autorun.create_files_on_removable_media_num > 0`

Function	Result	Event type	Examples
`create_or_modify_files(regex)`	Returns the number of events of a specific type.	Creates or changes files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.create_or_modify_files(/YogaGuide.job/)`
`create_or_modify_files_num`	Returns the number of events of a specific type.	Creates or modifies files.	`dr_sandbox.descr_tech.autorun.create_or_modify_files_num == 1`
`create_services(regex)`	Returns the number of events of a specific type.	Creates services. Only the services that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.create_services(/rsdsys/)`
`create_services_num`	Returns the amount of events of a certain type.	Creates services.	`dr_sandbox.descr_tech.autorun.create_services_num > 0`
`infect_executables(regex)`	Returns the amount of events of a certain type.	Infects executable files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.infect_executables(/eirmayxm/)`
`infect_executables_num`	Returns the number of events of a specific type.	Infects executable files.	`dr_sandbox.descr_tech.autorun.infect_executables_num > 0`
`modify_mbr`	Returns 1 if a master boot record (MBR) is modified, 0 otherwise.	Modifies the master boot record (MBR).	`dr_sandbox.descr_tech.autorun.modify_mbr`
`modify_registry(regex)`	Returns the number of events of a specific type.	Modifies registry keys. Only the keys that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.modify_registry(/C:\Users\user\AppData\Roaming\Sample.lnk/)`
`modify_registry_num`	Returns the number of events of a specific type.	Modifies registry keys.	`dr_sandbox.descr_tech.autorun.modify_registry_num >= 2`
`replace_system_executable_files(regex)`	Returns the number of events of a specific type.	Replaces executable system files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.autorun.replace_system_executable_files(/ir50_qc.dll/)`

Function	Result	Event type	Examples
`replace_system_executable_files_num`	Returns the number of events of a specific type.	Replaces executable system files.	`dr_sandbox.descr_tech.autorun.replace_system_executable_files_num > 0`

Modifies a file system (the 'filesystem' category)

Function	Result	Event type	Examples
`change_user_data_extensions`	Returns the number of events of a specific type.	Changes file extensions in user data (Trojan.Encoder).	`dr_sandbox.descr_tech.filesystem.change_user_data_extensions`
`create_files(regex)`	Returns the number of events of a specific type.	Creates files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.create_files(/nsArray.dll/)`
`create_files_num`	Returns the number of events of a specific type.	Creates files.	`dr_sandbox.descr_tech.filesystem.create_files_num >= 2`
`create_ransom_message_files`	Returns the number of events of a specific type.	Creates files and demands payment for file decoding (Trojan.Encoder).	`dr_sandbox.descr_tech.filesystem.create_ransom_message_files`
`modify_hosts`	Returns 1 if the HOSTS file is modified, 0 otherwise.	Modifies the HOSTS file.	`dr_sandbox.descr_tech.filesystem.modify_hosts`
`modify_user_data_files`	Returns the number of events of a specific type.	Changes a large amount of user data (Trojan.Encoder).	`dr_sandbox.descr_tech.filesystem.modify_user_data_files`
`move_files(regex)`	Returns the number of events of a specific type.	Moves files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.move_files(/%WINDIR%.*CONFIG\security.config.cch/)`
`move_files_num`	Returns the number of events of a specific type.	Moves files.	`dr_sandbox.descr_tech.filesystem.move_files_num >= 2`
`move_self(regex)`	Returns the number of events of a	Moves itself.	`dr_sandbox.descr_tech.filesystem.move_self(/CreativeAudi`

Function	Result	Event type	Examples
	specific type.		`o/)`
`move_self_num`	Returns the number of events of a specific type.	Moves itself.	`dr_sandbox.descr_tech.filesystem.move_self_num >= 2`
`move_system_files(regex)`	Returns the number of events of a specific type.	Moves system files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.move_system_files(/ir50_qc.dll/)`
`move_system_files_num`	Returns the number of events of a specific type.	Moves system files.	`dr_sandbox.descr_tech.filesystem.move_system_files_num >= 2`
`remove_files(regex)`	Returns the number of events of a specific type.	Deletes files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.remove_files(/^%TEMP%\7zS1.tmp\GOMPLAYERENSETUP.EXE$/)`
`remove_files_num`	Returns the number of events of a specific type.	Deletes files.	`dr_sandbox.descr_tech.filesystem.remove_files_num >= 2`
`remove_self`	Returns the number of events of a specific type.	Deletes itself.	`dr_sandbox.descr_tech.filesystem.remove_self`
`set_hidden(regex)`	Returns the number of events of a specific type.	Assigns the 'hidden' attribute to files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.set_hidden(/^%TEMP%\~2.cmd$/)`
`set_hidden_num`	Returns the number of events of a specific type.	Assigns the 'hidden' attribute to files.	`dr_sandbox.descr_tech.filesystem.set_hidden_num >= 2`
`substitute_executables(regex)`	Returns the number of events of a specific type.	Substitutes executable files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.substitute_executables(/pattern/)`
`substitute_executables_num`	Returns the number of events of a specific type.	Substitutes executable files.	`dr_sandbox.descr_tech.filesystem.substitute_executables_num >= 2`

Function	Result	Event type	Examples
`substitute_files(regex)`	Returns the number of events of a specific type.	Substitutes files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.filesystem.substitute_files(/pattern/)`
`substitute_files_num`	Returns the number of events of a specific type.	Substitutes files.	`dr_sandbox.descr_tech.filesystem.substitute_files_num >= 2`
`substitute_hosts`	Returns the number of events of a specific type.	Replaces the HOSTS file.	`dr_sandbox.descr_tech.filesystem.substitute_hosts`

Malicious functions (the 'malicious' category)

Function	Result	Event type	Examples
`add_antivirus_exclusion(regex)`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, adds anti-virus exclusions using the registry keys. Only the keys that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.add_antivirus_exclusion(/pattern/)`
`add_antivirus_exclusion_num`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, adds anti-virus exclusions using the registry keys.	`dr_sandbox.descr_tech.malicious.add_antivirus_exclusion_num`
`block_cmd`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks the Command Prompt (CMD) system utility.	`dr_sandbox.descr_tech.malicious.block_cmd`
`block_regedit`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks the	`dr_sandbox.descr_tech.malicious.block_regedit`

Function	Result	Event type	Examples
		Registry Editor (RegEdit) system utility.	
block_system_file_checker	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks System File Checker (SFC).	dr_sandbox.descr_tech.malicious.block_system_file_checker
block_system_restore	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks System Restore (SR).	dr_sandbox.descr_tech.malicious.block_system_restore
block_taskmgr	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks the Windows Task Manager (Taskmgr) system utility.	dr_sandbox.descr_tech.malicious.block_taskmgr
block_user_account_control	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks User Account Control (UAC).	dr_sandbox.descr_tech.malicious.block_user_account_control
block_windows_action_center	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks Windows Action Center.	dr_sandbox.descr_tech.malicious.block_windows_action_center
block_windows_defender	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks the Windows Defender system utility.	dr_sandbox.descr_tech.malicious.block_windows_defender
block_windows_file_protection	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks	dr_sandbox.descr_tech.malicious.block_windows_file_protection

Function	Result	Event type	Examples
		Windows File Protection (WFP).	
`block_windows_firewall`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks the Windows Firewall system utility.	`dr_sandbox.descr_tech.malicious.block_windows_firewall`
`block_windows_security_center`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks Windows Security Center.	`dr_sandbox.descr_tech.malicious.block_windows_security_center`
`block_windows_updates`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, blocks the Windows Update system utility.	`dr_sandbox.descr_tech.malicious.block_windows_updates`
`bruteforce_os_accounts`	Returns 1 if the event occurred, 0 otherwise.	Brute forces passwords of OS accounts.	`dr_sandbox.descr_tech.malicious.bruteforce_os_accounts`
`create_and_exec(regex)`	Returns the number of events of a specific type.	Creates and executes. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.create_and_exec(/Total Commander/)`
`create_and_exec_num`	Returns the number of events of a specific type.	Creates and executes.	`dr_sandbox.descr_tech.malicious.create_and_exec_num > 0`
`create_onion_service`	Returns the number of events of a specific type.	Creates an onion service.	`dr_sandbox.descr_tech.malicious.create_onion_service`
`delete_volume_shadow_copies`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, deletes volume shadow copies.	`dr_sandbox.descr_tech.malicious.delete_volume_shadow_copies`

Function	Result	Event type	Examples
`detect_virtual_machine(regex)`	Returns the number of events of a specific type.	Searches for windows to detect virtual machines. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.detect_virtual_machine(/pattern/)`
`detect_virtual_machine_num`	Returns the number of events of a specific type.	Searches for windows to detect virtual machines.	`dr_sandbox.descr_tech.malicious.detect_virtual_machine_num`
`disable_amsi`	Returns the number of events of a specific type.	Disables AMSI.	`dr_sandbox.descr_tech.malicious.disable_amsi`
`downloads_and_executes(regex)`	Returns the number of events of a specific type.	Downloads and executes. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.downloads_and_executes(/pattern/)`
`downloads_and_executes_num`	Returns the number of events of a specific type.	Downloads and executes.	`dr_sandbox.descr_tech.malicious.downloads_and_executes_num`
`downloads_and_executes_files`	Returns the number of events of a specific type.	Downloads and executes the files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.downloads_and_executes_files`
`download_file(regex)`	Returns the number of events of a specific type.	Downloads files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.download_file(/pattern/)`
`download_file_num`	Returns the number of events of a specific type.	Downloads files.	`dr_sandbox.descr_tech.malicious.download_file_num`
`download_files`	Returns 1 if the event occurred, 0 otherwise.	Downloads files.	`dr_sandbox.descr_tech.malicious.download_files`
`exec(regex)`	Returns the number of events of a specific type.	Executes. Only the objects that match the regular	`dr_sandbox.descr_tech.malicious.exec(/netsh.exe/)`

Function	Result	Event type	Examples
		expression are counted.	
`exec_num`	Returns the number of events of a specific type.	Executes.	`dr_sandbox.descr_tech.malicious.exec_num > 0`
`exec_wmi(regex)`	Returns the number of events of a specific type.	Executes WMI operations. Only the operations that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.exec_wmi(/pattern/)`
`exec_wmi_num`	Returns the number of events of a specific type.	Executes WMI operations.	`dr_sandbox.descr_tech.malicious.exec_wmi_num`
`exploit_create_and_exec(regex)`	Returns the number of events of a specific type.	Creates and executes (an exploit). Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.exploit_create_and_exec(/pattern/)`
`exploit_create_and_exec_num`	Returns the number of events of a specific type.	Creates and executes files (an exploit).	`dr_sandbox.descr_tech.malicious.exploit_create_and_exec_num`
`exploit_create_and_load_library(regex)`	Returns the number of events of a specific type.	Creates and loads libraries (an exploit). Only the libraries that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.exploit_create_and_load_library(/pattern/)`
`exploit_create_and_load_library_num`	Returns the number of events of a specific type.	Creates and loads libraries (an exploit).	`dr_sandbox.descr_tech.malicious.exploit_create_and_load_library_num`
`exploit_exec(regex)`	Returns the number of events of a specific type.	Executes (an exploit). Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.exploit_exec(/pattern/)`
`exploit_exec_num`	Returns the number of events of a specific type.	Executes (an exploit).	`dr_sandbox.descr_tech.malicious.exploit_exec_num`

Function	Result	Event type	Examples
`force_autorun_for_removable_media`	Returns 1 if the event occurred, 0 otherwise.	Forces autorun for removable media.	`dr_sandbox.descr_tech.malicious.force_autorun_for_removable_media`
`hide_from_view_file_extensions`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, forces the system to hide file extensions from view.	`dr_sandbox.descr_tech.malicious.hide_from_view_file_extensions`
`hide_from_view_hidden_files`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, forces the system to hide hidden files from view.	`dr_sandbox.descr_tech.malicious.hide_from_view_hidden_files`
`hide_processes(regex)`	Returns the number of events of a specific type.	Hides processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.hide_processes(/cscript.exe/)`
`hide_processes_num`	Returns the number of events of a specific type.	Hides processes.	`dr_sandbox.descr_tech.malicious.hide_processes_num > 0`
`hide_taskbar_notifications`	Returns 1 if the event occurred, 0 otherwise.	In order to make it harder to detect in the operating system, disables taskbar notifications.	`dr_sandbox.descr_tech.malicious.hide_taskbar_notifications`
`hook_in_browser(regex)`	Returns the number of events of a specific type.	Hooks functions in browsers. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.hook_in_browser(/pattern/)`
`hook_in_browser_num`	Returns the number of events of a specific type.	Hooks functions in browsers.	`dr_sandbox.descr_tech.malicious.hook_in_browser_num`
`hook_keyboard_all_processes(regex)`	Returns the number of events of a specific type.	Installs hooks to intercept	`dr_sandbox.descr_tech.malicious.hook_keyboard_all_processes(/OQKWHP\BJX.01/)`

Function	Result	Event type	Examples
		notifications on keystrokes. Handler for all processes (?LibraryPath).	
`hook_keyboard_all_processes_num`	Returns the number of events of a specific type.	Installs hooks to intercept notifications on keystrokes.	`dr_sandbox.descr_tech.malicious.hook_keyboard_all_processes_num > 0`
`hook_keyboard_concrete_processes(regex)`	Returns the number of events of a specific type.	Installs hooks to intercept notifications on keystrokes. Handler for the '(?HookedProcess.Name)' process: (?LibraryPath).	`dr_sandbox.descr_tech.malicious.hook_keyboard_concrete_processes(/IMDCSC.exe/)`
`hook_keyboard_concrete_processes_num`	Returns the number of events of a specific type.	Installs hooks to intercept notifications on keystrokes.	`dr_sandbox.descr_tech.malicious.hook_keyboard_concrete_processes_num > 0`
`hook_keyboard_on_window_messages(regex)`	Returns the number of events of a specific type.	Installs hooks to intercept notifications on window messages. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.hook_keyboard_on_window_messages(/pattern/)`
`hook_keyboard_on_window_messages_num`	Returns the number of events of a specific type.	Installs hooks to intercept notifications on window messages.	`dr_sandbox.descr_tech.malicious.hook_keyboard_on_window_messages_num`
`inject_to_a_lot_of_user_processes`	Returns 1 if the event occurred, 0 otherwise.	Injects code into numerous user processes.	`dr_sandbox.descr_tech.malicious.inject_to_a_lot_of_user_processes`
`inject_to_system_proc(regex)`	Returns the number of events of a specific type.	Injects code into system processes. Only the processes that match the	`dr_sandbox.descr_tech.malicious.inject_to_system_proc(/RegAsm.exe/)`

Function	Result	Event type	Examples
		regular expression are counted.	
`inject_to_system_proc_num`	Returns the number of events of a specific type.	Injects code into system processes.	`dr_sandbox.descr_tech.malicious.inject_to_system_proc_num > 0`
`inject_to_user_proc(regex)`	Returns the number of events of a specific type.	Injects code into user processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.inject_to_user_proc(/^iexplore.exe$/)`
`inject_to_user_proc_num`	Returns the number of events of a specific type.	Injects code into user processes.	`dr_sandbox.descr_tech.malicious.inject_to_user_proc_num > 0`
`modify_explorer_settings(regex)`	Returns the number of events of a specific type.	Modifies settings of Windows Explorer. Only the settings that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.modify_explorer_settings(/'NoFolderOptions' = '00000001'/)`
`modify_explorer_settings_num`	Returns the number of events of a specific type.	Modifies settings of Windows Explorer.	`dr_sandbox.descr_tech.malicious.modify_explorer_settings_num > 0`
`modify_ie_settings(regex)`	Returns the number of events of a specific type.	Modifies settings of Windows Internet Explorer. Only the settings that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.modify_ie_settings(/Zones\1] '1206' = '00000000'/)`
`modify_ie_settings_num`	Returns the number of events of a specific type.	Modifies settings of Windows Internet Explorer.	`dr_sandbox.descr_tech.malicious.modify_ie_settings_num > 0`
`modify_registry_to_bypass_firewall(regex)`	Returns the number of events of a specific type.	To bypass firewall, removes or modifies registry keys. Only the keys that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.modify_registry_to_bypass_firewall(/Enabled:taskmg.exe/)`
`modify_registry_to_bypass_`	Returns the number of events of a	To bypass firewall, removes or modifies	`dr_sandbox.descr_tech.malicious.modify_registry_to_bypas`

Function	Result	Event type	Examples
`firewall_num`	specific type.	registry keys.	`s_firewall_num > 0`
`modify_system_dns(regex)`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, modifies DNS servers. Only the servers that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.modify_system_dns(/pattern/)`
`modify_system_dns_num`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, modifies DNS servers.	`dr_sandbox.descr_tech.malicious.modify_system_dns_num`
`modify_system_settings(regex)`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, modifies system settings. Only the settings that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.modify_system_settings(/pattern/)`
`modify_system_settings_num`	Returns the number of events of a specific type.	In order to make it harder to detect in the operating system, modifies system settings.	`dr_sandbox.descr_tech.malicious.modify_system_settings_num`
`read_third_party_passwords(regex)`	Returns the number of events of a specific type.	Reads files that store third party app passwords. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.read_third_party_passwords(/pattern/)`
`read_third_party_passwords_num`	Returns the number of events of a specific type.	Reads files that store third party app passwords.	`dr_sandbox.descr_tech.malicious.read_third_party_passwords_num`
`register_bho(regex)`	Returns the number of events of a specific type.	Registers BHO. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.register_bho(/pattern/)`

Function	Result	Event type	Examples
`register_com_server(regex)`	Returns the number of events of a specific type.	Registers a COM server. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.register_com_server(/pattern/)`
`register_com_server_num`	Returns the number of events of a specific type.	Registers a COM server.	`dr_sandbox.descr_tech.malicious.register_com_server_num`
`register_filesystem_filter(regex)`	Returns the number of events of a specific type.	Registers a file system filter. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.register_filesystem_filter(/pattern/)`
`restore_ssdt_hooks`	Returns 1 if the event occurred, 0 otherwise.	Restores hooked functions in the System Service Descriptor Table (SSDT).	`dr_sandbox.descr_tech.malicious.restore_ssdt_hooks`
`search_password_in_registry(regex)`	Returns the number of events of a specific type.	Searches for registry branches where third party apps store their passwords. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.search_password_in_registry(/MessengerService/)`
`search_password_in_registry_num`	Returns the number of events of a specific type.	Searches for registry branches where third-party apps store their passwords.	`dr_sandbox.descr_tech.malicious.search_password_in_registry_num > 0`
`search_wnd_for_analyzing_soft(regex)`	Returns the number of events of a specific type.	Searches for windows to detect analytical utilities. Only the objects that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.search_wnd_for_analyzing_soft(/PEiD/)`
`search_wnd_for_analyzing_soft_num`	Returns the number of events of a specific type.	Searches for windows to detect analytical utilities.	`dr_sandbox.descr_tech.malicious.search_wnd_for_analyzing_soft_num > 0`

Function	Result	Event type	Examples
`search_wnd_for_programs_and_games(regex)`	Returns the number of events of a specific type.	Searches for windows to detect apps and games. Only the windows that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.search_wnd_for_programs_and_games(/The Wireshark Network Analyzer/)`
`search_wnd_for_programs_and_games_num`	Returns the number of events of a specific type.	Searches for windows to detect apps and games.	`dr_sandbox.descr_tech.malicious.search_wnd_for_programs_and_games_num > 0`
`search_wnd_to_bypass_av(regex)`	Returns the number of events of a specific type.	Searches for windows to bypass anti-viruses. Only the windows that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.search_wnd_to_bypass_av(/AVP.AlertDialog/)`
`search_wnd_to_bypass_av_num`	Returns the number of events of a specific type.	Searches for windows to bypass anti-viruses.	`dr_sandbox.descr_tech.malicious.search_wnd_to_bypass_av_num > 0`
`search_wnd_to_bypass_wfp(regex)`	Returns the number of events of a specific type.	Searches for windows to bypass Windows Files Protection (WFP). Only the windows that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.search_wnd_to_bypass_wfp(/Windows File Protection/)`
`search_wnd_to_bypass_wfp_num`	Returns the number of events of a specific type.	Searches for windows to bypass Windows Files Protection (WFP).	`dr_sandbox.descr_tech.malicious.search_wnd_to_bypass_wfp_num > 0`
`set_concrete_ssdt_hooks(regex)`	Returns the number of events of a specific type.	Hooks functions in System Service Descriptor Table (SSDT). Only the functions that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.set_concrete_ssdt_hooks(/pattern/)`
`set_concrete_ssdt_hooks_num`	Returns the number of events of a specific type.	Hooks functions in System Service Descriptor Table (SSDT).	`dr_sandbox.descr_tech.malicious.set_concrete_ssdt_hooks_num`

Function	Result	Event type	Examples
`set_homepage_for_ie`	Returns 1 if the event occurred, 0 otherwise.	Sets a new unauthorized home page for Windows Internet Explorer.	`dr_sandbox.descr_tech.malicious.set_homepage_for_ie`
`set_ssdt_hooks`	Returns the number of events of a specific type.	Hooks functions in System Service Descriptor Table (SSDT).	`dr_sandbox.descr_tech.malicious.set_ssdt_hooks`
`try_to_terminate_a_lot_of_user_processes`	Returns 1 if the event occurred, 0 otherwise.	Terminates or attempts to terminate numerous user processes.	`dr_sandbox.descr_tech.malicious.try_to_terminate_a_lot_of_user_processes`
`try_to_terminate_system_processes(regex)`	Returns the number of events of a specific type.	Terminates or attempts to terminate system processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.try_to_terminate_system_processes(/ctfmon.exe/)`
`try_to_terminate_system_processes_num`	Returns the number of events of a specific type.	Terminates or attempts to terminate system processes.	`dr_sandbox.descr_tech.malicious.try_to_terminate_system_processes_num > 0`
`try_to_terminate_user_processes(regex)`	Returns the number of events of a specific type.	Terminates or attempts to terminate user processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech.malicious.try_to_terminate_user_processes(/^AVSYNMGR.EXE$/)`
`try_to_terminate_user_processes_num`	Returns the number of events of a specific type.	Terminates or attempts to terminate user processes.	`dr_sandbox.descr_tech.malicious.try_to_terminate_user_processes_num > 0`

Miscellaneous (the 'miscellaneous' category)

Function	Result	Event type	Examples
`add_root_certificate`	Returns 1 if the scanned object adds certificate, 0 otherwise.	Adds a root certificate.	`dr_sandbox.descr_tech.miscellaneous.add_root_certificate`
`create_and_exec`	Returns 1 if the event occurred, 0 otherwise.	Creates and executes (with a hidden window).	`dr_sandbox.descr_tech.miscellaneous.create_and_exec`
`disable_certificate`	Returns 1 if the event occurred, 0 otherwise.	Disables a certificate.	`dr_sandbox.descr_tech.miscellaneous.disable_certificate`
`exec(regex)`	Returns the number of events of a specific type.	Executes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech.miscellaneous.exec(/pattern/)`
`load_driver(regex)`	Returns the number of events of a specific type.	Loads the drivers. Only the drivers that match the regular expression are counted.	`dr_sandbox.descr_tech.miscellaneous.load_driver(/pattern/)`
`load_driver_num`	Returns the number of events of a specific type.	Loads drivers.	`dr_sandbox.descr_tech.miscellaneous.load_driver_num`
`modify_auto_config_url(regex)`	Returns the number of events of a specific type.	Changes the AutoConfigURL parameter. Only the values that match the regular expression are counted.	`dr_sandbox.descr_tech.miscellaneous.modify_auto_config_url(/pattern/)`
`search_wnd(regex)`	Returns the number of events of a specific type.	Searches for windows. Only the windows that match the regular expression are counted.	`dr_sandbox.descr_tech.miscellaneous.search_wnd(/MS_WebcheckMonitor/)`
`search_wnd_num`	Returns the number of events of a specific type.	Searches for windows.	`dr_sandbox.descr_tech.miscellaneous.search_wnd_num == 3`

Function	Result	Event type	Examples
`shut_down_windows`	Returns 1 if the event occurred, 0 otherwise.	Attempts to shut down Windows OS.	`dr_sandbox.descr_tech.miscellaneous.shut_down_windows`
`use_ntfs_data_streams`	Returns 1 if the event occurred, 0 otherwise.	Uses NTFS alternate data streams.	`dr_sandbox.descr_tech.miscellaneous.use_ntfs_data_streams`

Network activity (the 'network' category)

Function	Result	Event type	Examples
`connect_to(regex)`	Returns the number of events of a specific type.	Connects to the objects listed in the regular expression.	`dr_sandbox.descr_tech.network.connect_to(/www.xfo.cn/)`
`connect_to_num`	Returns the number of events of a specific type.	Connects to the objects.	`dr_sandbox.descr_tech.network.connect_to_num >= 2`
`tcp(regex)`	Returns the number of events of a specific type.	TCP requests.	`dr_sandbox.descr_tech.network.tcp(/pattern/)`
`tcp_num`	Returns the number of events of a specific type.	TCP requests.	`dr_sandbox.descr_tech.network.tcp_num`
`tcp_http_get(regex)`	Returns the number of events of a specific type.	HTTP GET requests using TCP.	`dr_sandbox.descr_tech.network.tcp_http_get(/addurl.html$/)`
`tcp_http_get_num`	Returns the number of events of a specific type.	HTTP GET requests using TCP.	`dr_sandbox.descr_tech.network.tcp_http_get_num >= 2`
`tcp_http_post(regex)`	Returns the number of events of a specific type.	HTTP POST requests using TCP.	`dr_sandbox.descr_tech.network.tcp_http_post(/addurl.html$/)`
`tcp_http_post_num`	Returns the number of events of a specific type.	HTTP POST requests using TCP.	`dr_sandbox.descr_tech.network.tcp_http_post_num >= 2`
`tcp_http_unk(regex)`	Returns the number of events of a specific type.	Unknown HTTP requests.	`dr_sandbox.descr_tech.network.tcp_http_unk(/pattern/)`

Function	Result	Event type	Examples
tcp_http_unk_num	Returns the number of events of a specific type.	Unknown HTTP requests.	dr_sandbox.descr_tech.network.tcp_http_unk_num
udp(regex)	Returns the number of events of a specific type.	UDP requests.	dr_sandbox.descr_tech.network.udp(/disk57/)
udp_num	Returns the number of events of a specific type.	UDP requests.	dr_sandbox.descr_tech.network.udp_num >= 2

Functions for the Linux sandbox (the 'descr_tech_lbcl' category)

Enabling autorun and distribution (the 'autorun' category)

Function	Result	Event type	Examples
create_or_modify_files(regex)	Returns the number of events of a specific type.	Creates or changes files. Only the files that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.autorun.create_or_modify_files(/pattern/)
create_or_modify_files_num	Returns the number of events of a specific type.	Creates or modifies files.	dr_sandbox.descr_tech_lbcl.autorun.create_or_modify_files_num
create_or_modify_symlinks(regex)	Returns the number of events of a specific type.	Creates or modifies symbolic links. Only the links that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.autorun.create_or_modify_symlinks(/pattern/)
create_or_modify_symlinks_num	Returns the number of events of a specific type.	Creates or modifies symbolic links.	dr_sandbox.descr_tech_lbcl.autorun.create_or_modify_symlinks_num

Modifies a file system (the 'filesystem' category)

Function	Result	Event type	Examples
change_time_o	Returns the number of events of a	Changes the time when the file was	dr_sandbox.descr_tech_lbcl.filesystem.change_time_of_fil

Function	Result	Event type	Examples
`f_file(regex)`	specific type.	created, accessed, or modified. Only the files that match the regular expression are counted.	`e(/pattern/)`
`change_time_of_file_num`	Returns the number of events of a specific type.	Changes the time when the file was created, accessed, or modified.	`dr_sandbox.descr_tech_lbcl.filesystem.change_time_of_file_num`
`create_dir(regex)`	Returns the number of events of a specific type.	Creates directories. Only the directories that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.filesystem.create_dir(/pattern/)`
`create_dir_num`	Returns the number of events of a specific type.	Creates directories.	`dr_sandbox.descr_tech_lbcl.filesystem.create_dir_num`
`create_or_modify_file(regex)`	Returns the number of events of a specific type.	Creates or changes files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.filesystem.create_or_modify_file(/pattern/)`
`create_or_modify_file_num`	Returns the number of events of a specific type.	Creates or modifies files.	`dr_sandbox.descr_tech_lbcl.filesystem.create_or_modify_file_num`
`create_symlink(regex)`	Returns the number of events of a specific type.	Creates symbolic links.	`dr_sandbox.descr_tech_lbcl.filesystem.create_symlink(/pattern/)`
`create_symlink_num`	Returns the number of events of a specific type.	Only the links that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.filesystem.create_symlink_num`
`lock_file(regex)`	Returns the number of events of a specific type.	Blocks files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.filesystem.lock_file(/pattern/)`
`lock_file_num`	Returns the number of events of a specific type.	Blocks files.	`dr_sandbox.descr_tech_lbcl.filesystem.lock_file_num`
`modify_file_a`	Returns the number of events of a	Changes file access rights.	`dr_sandbox.descr_tech_lbcl.filesystem.modify_file_access`

Function	Result	Event type	Examples
ccess_rights(regex)	specific type.		_rights(/pattern/)
modify_file_access_rights_num	Returns the number of events of a specific type.	Changes file access rights.	dr_sandbox.descr_tech_lbcl.filesystem.modify_file_access_rights_num
modify_file_owner(regex)	Returns the number of events of a specific type.	Changes a file owner.	dr_sandbox.descr_tech_lbcl.filesystem.modify_file_owner(/pattern/)
modify_file_owner_num	Returns the number of events of a specific type.	Changes a file owner.	dr_sandbox.descr_tech_lbcl.filesystem.modify_file_owner_num
mount_file_system(regex)	Returns the number of events of a specific type.	Mounts file systems. Only the systems that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.filesystem.mount_file_system(/pattern/)
mount_file_system_num	Returns the number of events of a specific type.	Mounts file systems.	dr_sandbox.descr_tech_lbcl.filesystem.mount_file_system_num
remove_dir(regex)	Returns the number of events of a specific type.	Deletes directories. Only the directories that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.filesystem.remove_dir(/pattern/)
remove_dir_num	Returns the number of events of a specific type.	Deletes directories.	dr_sandbox.descr_tech_lbcl.filesystem.remove_dir_num
remove_file(regex)	Returns the number of events of a specific type.	Deletes files. Only the files that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.filesystem.remove_file(/pattern/)
remove_file_num	Returns the number of events of a specific type.	Deletes files.	dr_sandbox.descr_tech_lbcl.filesystem.remove_file_num
unmount_file_system(regex)	Returns the number of events of a specific type.	Unmounts file systems. Only the systems that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.filesystem.unmount_file_system(/pattern/)

Function	Result	Event type	Examples
unmount_file_system_num	Returns the number of events of a specific type.	Unmounts file systems.	dr_sandbox.descr_tech_lbcl.filesystem.unmount_file_system_num

Malicious functions (the 'malicious' category)

Function	Result	Event type	Examples
attempt_kill_system_proc(regex)	Returns the number of events of a specific type.	Tries to kill system processes. Only the processes that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.malicious.attempt_kill_system_proc(/pattern/)
attempt_kill_system_proc_num	Returns the number of events of a specific type.	Tries to kill system processes.	dr_sandbox.descr_tech_lbcl.malicious.attempt_kill_system_proc_num
attept_kill_analyzers(regex)	Returns the number of events of a specific type.	Tries to kill analyzers. Only the analyzers that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.malicious.attept_kill_analyzers(/pattern/)
attept_kill_analyzers_num	Returns the number of events of a specific type.	Tries to kill analyzers.	dr_sandbox.descr_tech_lbcl.malicious.attept_kill_analyzers_num
attept_kill_proc(regex)	Returns the number of events of a specific type.	Tries to kill processes. Only the processes that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.malicious.attept_kill_proc(/pattern/)
attept_kill_proc_num	Returns the number of events of a specific type.	Tries to kill processes.	dr_sandbox.descr_tech_lbcl.malicious.attept_kill_proc_num
compile_program_from_source_codes(regex)	Returns the number of events of a specific type.	Compiles source code.	dr_sandbox.descr_tech_lbcl.malicious.compile_program_from_source_codes(/pattern/)
compile_program_from_source_codes_num	Returns the number of events of a specific type.	Compiles source code.	dr_sandbox.descr_tech_lbcl.malicious.compile_program_from_source_codes_num

Function	Result	Event type	Examples
`gain_root_privileges`	Returns the number of events of a specific type.	Gains root access.	`dr_sandbox.descr_tech_lbcl.malicious.gain_root_privileges`
`get_access_to_ssh_keys`	Returns the number of events of a specific type.	Accesses SSH keys.	`dr_sandbox.descr_tech_lbcl.malicious.get_access_to_ssh_keys`
`inject_to_proc(regex)`	Returns the number of events of a specific type.	Injects itself in processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.inject_to_proc(/pattern/)`
`inject_to_proc_num`	Returns the number of events of a specific type.	Injects itself in processes.	`dr_sandbox.descr_tech_lbcl.malicious.inject_to_proc_num`
`kill_analyzers(regex)`	Returns the number of events of a specific type.	Kills analyzers. Only the analyzers that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.kill_analyzers(/pattern/)`
`kill_analyzers_num`	Returns the number of events of a specific type.	Kills analyzers.	`dr_sandbox.descr_tech_lbcl.malicious.kill_analyzers_num`
`kill_proc(regex)`	Returns the number of events of a specific type.	Kills processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.kill_proc(/pattern/)`
`kill_proc_num`	Returns the number of events of a specific type.	Kills processes.	`dr_sandbox.descr_tech_lbcl.malicious.kill_proc_num`
`kill_system_proc(regex)`	Returns the number of events of a specific type.	Kills system processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.kill_system_proc(/pattern/)`
`kill_system_proc_num`	Returns the number of events of a specific type.	Kills system processes.	`dr_sandbox.descr_tech_lbcl.malicious.kill_system_proc_num`

Function	Result	Event type	Examples
`launch_itself_as_daemon`	Returns the number of events of a specific type.	Launches itself as a daemon.	`dr_sandbox.descr_tech_lbcl.malicious.launch_itself_as_daemon`
`launch_processes(regex)`	Returns the number of events of a specific type.	Launches processes. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.launch_processes(/pattern/)`
`launch_processes_num`	Returns the number of events of a specific type.	Launches processes.	`dr_sandbox.descr_tech_lbcl.malicious.launch_processes_num`
`manage_services(regex)`	Returns the number of events of a specific type.	Manages services. Only the services that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.manage_services(/pattern/)`
`manage_services_num`	Returns the number of events of a specific type.	Manages services.	`dr_sandbox.descr_tech_lbcl.malicious.manage_services_num`
`modify_firewall_settings(regex)`	Returns the number of events of a specific type.	Changes firewall settings. Only the settings that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.modify_firewall_settings(/pattern/)`
`modify_firewall_settings_num`	Returns the number of events of a specific type.	Changes firewall settings.	`dr_sandbox.descr_tech_lbcl.malicious.modify_firewall_settings_num`
`modify_router_settings(regex)`	Returns the number of events of a specific type.	Changes router settings. Only the settings that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.modify_router_settings(/pattern/)`
`modify_router_settings_num`	Returns the number of events of a specific type.	Changes router settings.	`dr_sandbox.descr_tech_lbcl.malicious.modify_router_settings_num`
`operate_kernel_modules(regex)`	Returns the number of events of a specific type.	Operates kernel modules.	`dr_sandbox.descr_tech_lbcl.malicious.operate_kernel_modules(/pattern/)`

Function	Result	Event type	Examples
`operate_kernel_modules_num`	Returns the number of events of a specific type.	Operates kernel modules.	`dr_sandbox.descr_tech_lbcl.malicious.operate_kernel_modules_num`
`perform_process_tracing(regex)`	Returns the number of events of a specific type.	Performs process tracing. Only the processes that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.perform_process_tracing(/pattern/)`
`perform_process_tracing_num`	Returns the number of events of a specific type.	Performs process tracing.	`dr_sandbox.descr_tech_lbcl.malicious.perform_process_tracing_num`
`remove_self`	Returns the number of events of a specific type.	Deletes itself.	`dr_sandbox.descr_tech_lbcl.malicious.remove_self`
`remove_system_files(regex)`	Returns the number of events of a specific type.	Deletes system files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.remove_system_files(/pattern/)`
`remove_system_files_num`	Returns the number of events of a specific type.	Deletes system files.	`dr_sandbox.descr_tech_lbcl.malicious.remove_system_files_num`
`replace_system_files(regex)`	Returns the number of events of a specific type.	Replaces system files. Only the files that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.replace_system_files(/pattern/)`
`replace_system_files_num`	Returns the number of events of a specific type.	Replaces system files.	`dr_sandbox.descr_tech_lbcl.malicious.replace_system_files_num`
`stops_system_services(regex)`	Returns the number of events of a specific type.	Stops system services. Only the services that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.malicious.stops_system_services(/pattern/)`
`stops_system_services_num`	Returns the number of events of a specific type.	Stops system services.	`dr_sandbox.descr_tech_lbcl.malicious.stops_system_services_num`

Function	Result	Event type	Examples
substitute_application_name_for(regex)	Returns the number of events of a specific type.	Substitutes an application name.	dr_sandbox.descr_tech_lbcl.malicious.substitute_application_name_for(/pattern/)
substitute_application_name_for_num	Returns the number of events of a specific type.	Substitutes an application name.	dr_sandbox.descr_tech_lbcl.malicious.substitute_application_name_for_num

Network activity (the 'network' category)

Function	Result	Event type	Examples
attack_bruteforce_via_ssh	Returns the number of events of a specific type.	Performs a bruteforce attack via the SSH protocol.	dr_sandbox.descr_tech_lbcl.network.attack_bruteforce_via_ssh
attack_bruteforce_via_telnet	Returns the number of events of a specific type.	Performs a bruteforce attack via the TELNET protocol.	dr_sandbox.descr_tech_lbcl.network.attack_bruteforce_via_telnet
attack_bruteforce_via_unk_protocol	Returns the number of events of a specific type.	Performs a bruteforce attack via the undefined protocol.	dr_sandbox.descr_tech_lbcl.network.attack_bruteforce_via_unk_protocol
connect_to(regex)	Returns the number of events of a specific type.	Connects to servers. Only the servers that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.network.connect_to(/pattern/)
connect_to_num	Returns the number of events of a specific type.	Connects to servers.	dr_sandbox.descr_tech_lbcl.network.connect_to_num
connect_to_irc(regex)	Returns the number of events of a specific type.	Connects to servers over the IRC protocol. Only the servers that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.network.connect_to_irc(/pattern/)
dns_ask(regex)	Returns the number of events of a specific type.	DNS queries.	dr_sandbox.descr_tech_lbcl.network.dns_ask(/pattern/)

Function	Result	Event type	Examples
`dns_ask_num`	Returns the number of events of a specific type.	DNS queries.	`dr_sandbox.descr_tech_lbcl.network.dns_ask_num`
`http_get(regex)`	Returns the number of events of a specific type.	HTTP GET requests.	`dr_sandbox.descr_tech_lbcl.network.http_get(/pattern/)`
`http_get_num`	Returns the number of events of a specific type.	HTTP GET requests.	`dr_sandbox.descr_tech_lbcl.network.http_get_num`
`http_other(regex)`	Returns the number of events of a specific type.	Other HTTP requests.	`dr_sandbox.descr_tech_lbcl.network.http_other(/pattern/)`
`http_other_num`	Returns the number of events of a specific type.	Other HTTP requests.	`dr_sandbox.descr_tech_lbcl.network.http_other_num`
`http_post(regex)`	Returns the number of events of a specific type.	HTTP POST requests.	`dr_sandbox.descr_tech_lbcl.network.http_post(/pattern/)`
`http_post_num`	Returns the number of events of a specific type.	HTTP POST requests.	`dr_sandbox.descr_tech_lbcl.network.http_post_num`
`listening_port(regex)`	Returns the number of events of a specific type.	Awaits incoming connections on ports. Only the ports that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.network.listening_port(/pattern/)`
`listening_port_num`	Returns the number of events of a specific type.	Awaits incoming connections on ports.	`dr_sandbox.descr_tech_lbcl.network.listening_port_num`
`receive_data_from_server(regex)`	Returns the number of events of a specific type.	Receives data from servers. Only the servers that match the regular expression are counted.	`dr_sandbox.descr_tech_lbcl.network.receive_data_from_server(/pattern/)`
`receive_data_from_server_num`	Returns the number of events of a specific type.	Receives data from servers.	`dr_sandbox.descr_tech_lbcl.network.receive_data_from_server_num`

Function	Result	Event type	Examples
send_data_to_server(regex)	Returns the number of events of a specific type.	Sends data to servers. Only the servers that match the regular expression are counted.	dr_sandbox.descr_tech_lbcl.network.send_data_to_server(/pattern/)
send_data_to_server_num	Returns the number of events of a specific type.	Sends data to servers.	dr_sandbox.descr_tech_lbcl.network.send_data_to_server_num

Other (the 'other' category)

Function	Result	Event type	Examples
collect_cpu_info	Returns the number of events of a specific type.	Collects information about the CPU.	dr_sandbox.descr_tech_lbcl.other.collect_cpu_info
collect_network_info	Returns the number of events of a specific type.	Collects information about the network activity.	dr_sandbox.descr_tech_lbcl.other.collect_network_info
collect_os_info	Returns the number of events of a specific type.	Collects information about the OS.	dr_sandbox.descr_tech_lbcl.other.collect_os_info
collect_ram_info	Returns the number of events of a specific type.	Collects information about RAM.	dr_sandbox.descr_tech_lbcl.other.collect_ram_info
read_info_from_proc_kallsyms	Returns the number of events of a specific type.	Reads information from /proc/kallsyms.	dr_sandbox.descr_tech_lbcl.other.read_info_from_proc_kallsyms

Detects (the 'detects' category)

Function	Result	Event type	Examples
all_detects_here(regexp)	Returns the number of events of a specific type.	All detects.	dr_sandbox.detects.all_detects_here(/Virlock/)
all_detects_here_num	Returns the number of events of a	All detects.	dr_sandbox.detects.all_detects_here_num

Function	Result	Event type	Examples
	specific type.		
`detects_of_allocs(regexp)`	Returns the number of events of a specific type.	Detects of alloc files.	`dr_sandbox.detects.detects_of_allocs(/Virlock/)`
`detects_of_allocs_num`	Returns the number of events of a specific type.	Detects of alloc files.	`dr_sandbox.detects.detects_of_allocs_num`
`detects_of_drops(regexp)`	Returns the number of events of a specific type.	Detects of drops.	`dr_sandbox.detects.detects_of_drops(/Virlock/)`
`detects_of_drops_num`	Returns the number of events of a specific type.	Detects of drops.	`dr_sandbox.detects.detects_of_drops_num`
`detects_of_dumps(regexp)`	Returns the number of events of a specific type.	Detects of dumps.	`dr_sandbox.detects.detects_of_dumps(/Virlock/)`
`detects_of_dumps_num`	Returns the number of events of a specific type.	Detects of dumps.	`dr_sandbox.detects.detects_of_dumps_num`
`detects_of_injects(regexp)`	Returns the number of events of a specific type.	Detects of injects.	`dr_sandbox.detects.detects_of_injects(/Virlock/)`
`detects_of_injects_num`	Returns the number of events of a specific type.	Detects of injects.	`dr_sandbox.detects.detects_of_injects_num`
`detects_of_src(regexp)`	Returns the number of events of a specific type.	Detects of src files.	`dr_sandbox.detects.detects_of_src(/Virlock/)`
`detects_of_src_num`	Returns the number of events of a specific type.	Detects of src files.	`dr_sandbox.detects.detects_of_src_num`

Other functions

Function	Description	Examples
`check_buffer(offset, buffer_asciihex_value)`	Check an asciihex buffer at the specified offset. Length must be even. Can be used instead of 'strings' part, for example, to not slow down the scanning. Returns 1 if the string is found, 0 otherwise.	`dr_sandbox.check_buffer(0,"4d5A")`
`check_byte(offset, byte_value)`	Check bytes at the specified offset. Can be used instead of 'strings' part, for example, to not slow down the scanning. Returns 1 if a value in bytes is found, 0 otherwise.	`dr_sandbox.check_byte(0,0x4d)`
`check_dword(offset, dword_value)`	Check dwords at the specified offset. Can be used instead of 'strings' part, for example, to not slow down the scanning. Returns 1 if a DWORD value is found, 0 otherwise.	`dr_sandbox.check_dword(0,0x00905A4D)`
`check_word(offset, word_value)`	Check words at the specified offset. Can be used instead of the 'strings' part, for example, to not slow down the scanning. Returns 1 if a WORD value is found, 0 otherwise.	`dr_sandbox.check_word(0,0x5a4d)`
`ci_any(string)`	Returns 1 if the case-insensitive ASCII or wide string is found, 0 otherwise.	`dr_sandbox.ci_any("string")`
`ci_any_num(string)`	Returns the number of case-insensitive ASCII or wide strings that are found, 0 otherwise.	`dr_sandbox.ci_any_num("string")`

Function	Description	Examples
`ci_ascii(string)`	Returns 1 if the case-insensitive ASCII string is found, 0 otherwise.	`dr_sandbox.ci_ascii("string")`
`ci_ascii_num(string)`	Returns the number of case-insensitive ASCII strings that are found, 0 otherwise.	`dr_sandbox.ci_ascii_num("string")`
`ci_wide(string)`	Returns 1 if a case-insensitive wide string is found, 0 otherwise.	`dr_sandbox.ci_wide("string")`
`ci_wide_num(string)`	Returns the number of case-insensitive wide strings that are found, 0 otherwise.	`dr_sandbox.ci_wide_num("string")`
`ci_xor(string)`	Returns 1 if the case-insensitive XOR-ed 1-byte ASCII string is found, 0 otherwise.	`dr_sandbox.ci_xor("string")`
`ci_xor_num(string)`	Returns the number of case-insensitive XOR-ed 1-byte ASCII strings that are found, 0 otherwise.	`dr_sandbox.ci_xor_num("string")`
`crc32(integer, integer)`	Calculates and returns the crc32 hash of the buffer. The first parameter is the offset, and the second parameter is the length of the buffer.	`dr_sandbox.crc32(0, 0)`
`cs_any(string)`	Returns 1 if the case-sensitive ASCII or wide string is found, 0 otherwise.	`dr_sandbox.cs_any("string")`
`cs_any_num(string)`	Returns the number of case-sensitive ASCII or wide strings that are found, 0 otherwise.	`dr_sandbox.cs_any_num("string")`
`cs_ascii(string)`	Returns 1 if the case-sensitive ASCII string is found, 0 otherwise.	`dr_sandbox.cs_ascii("string")`
`cs_ascii_num(string)`	Returns the number of case-sensitive ASCII strings that are found, 0 otherwise.	`dr_sandbox.cs_ascii_num("string")`

Function	Description	Examples
`cs_wide(string)`	Returns 1 if the case-sensitive wide string is found, 0 otherwise.	`dr_sandbox.cs_wide("string")`
`cs_wide_num(string)`	Returns the number of case-sensitive wide strings that are found, 0 otherwise.	`dr_sandbox.cs_wide_num("string")`
`detects_of_this_file(regex)`	Returns the number of detects on a scanned file.	`dr_sandbox.detects_of_this_file(/Virlock/) == 0`
`detects_of_this_file_num`	Returns the number of detects on a scanned file.	`dr_sandbox.detects_of_this_file_num`
`filename(regex)`	Returns 1 if the regular expression is found in the file name, 0 otherwise.	`dr_sandbox.filename(/xtbl/)`
`filename_boost_regex(string_with_regex)`	Search for a regular expression in a file name using `boost::regex`. Flags for regex: `boost::regex::perl`. Search by `boost::regex_search`. Can be used if you need regex features like negative lookahead or backreferences, which are not supported in the YARA regex. Note that invalid regex will slow down the scanning. Moreover, `boost::regex` is slower than the YARA regex, it's recommended to use `dr_sandbox.filename(//)` if possible. Returns 1 if the regular expression is found, 0 otherwise.	`dr_sandbox.filename_boost_regex(“(?<!abc)def”)`
`filesystem_access(regex)`	The high-level function, which matches all filesystem operations to the regular expression.	`dr_sandbox.filesystem_access(/AnnaKournikova\.jpg\.vbs/)`
`network_access(re`	The high-level function, which matches all network	`dr_sandbox.network_access(/\.php\?id=[0-9]+&token=[0-9]+/)`

Function	Description	Examples
`gex)`	operations to the regular expression.	
`registry_access(regex)`	Returns the number of actions with a registry.	`dr_sandbox.registry_access(/pattern/)`
`sb_filetype`	Returns a file type. Used for comparing with the following `SB_FILETYPE_*` constants: `SB_FILETYPE_SRC`; `SB_FILETYPE_DROP`; `SB_FILETYPE_MEMDMP`; `SB_FILETYPE_ALLOC`; `SB_FILETYPE_DUMP`; `SB_FILETYPE_INJECT`.	`dr_sandbox.sb_filetype == dr_sandbox.SB_FILETYPE_SRC`
`search_substring_in_range(string, integer, integer)`	Search for the substring in the buffer using the Boyer–Moore algorithm. The first argument is the asciihex string, the second parameter is the offset, and the third parameter is the length. Use it carefully because it's not performance free.	`dr_sandbox.search_substring_in_range("string", 0, 0)`