RSA® NETWITNESS®

Intel Feeds Implementation Guide

Kaspersky CyberTrace

Alexey Dolgikh, Kaspersky Lab System analyst

Last Modified: March 13th, 2019

Solution Summary

Kaspersky Lab offers continuously updated Threat Intelligence Data Feeds to inform customers about risks and implications associated with cyber-threats. The real-time data helps to mitigate threats more effectively.

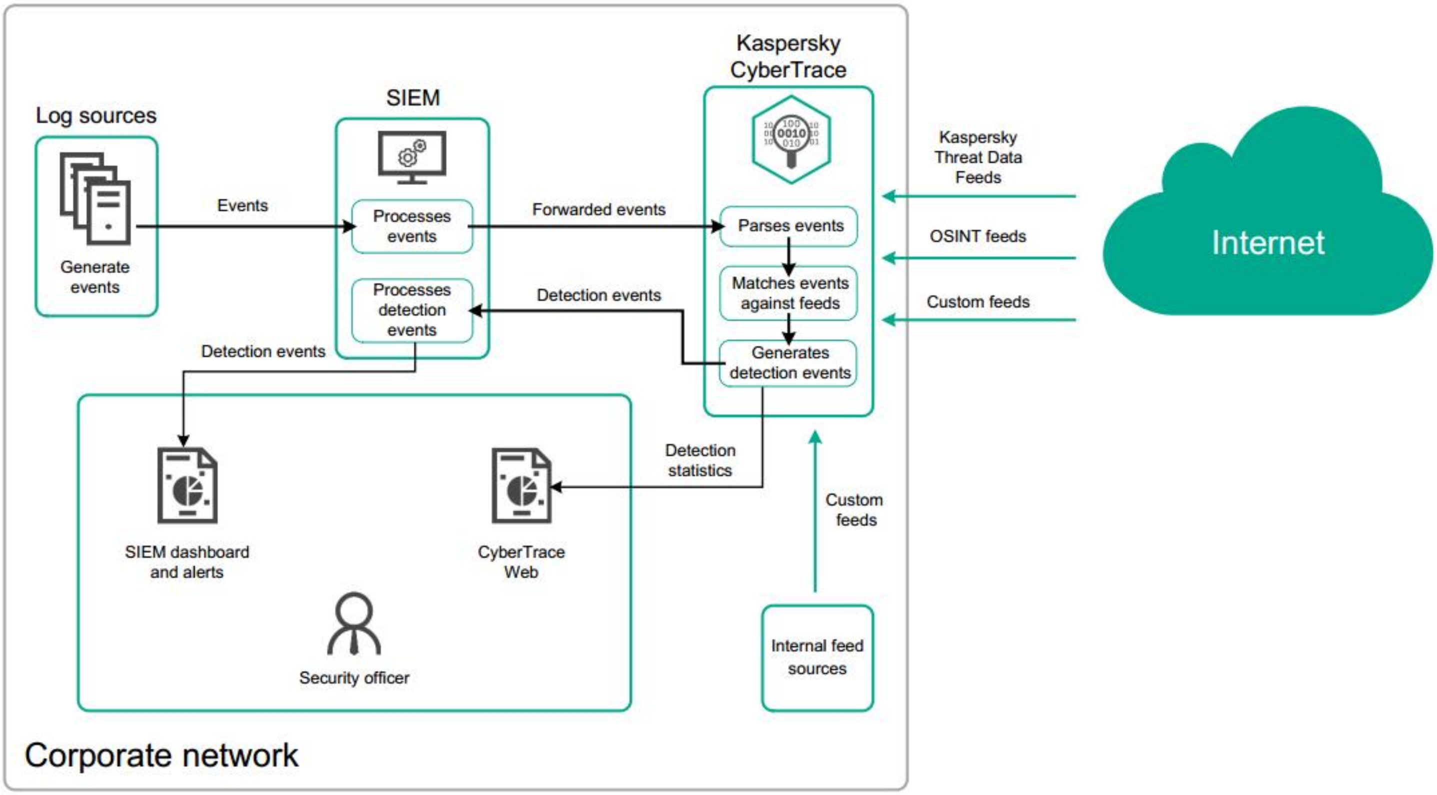
The following feeds are available:

- **Botnet CnC URL Data Feed**—A set of URLs and hashes with context that cover desktop botnet C&C servers and related malicious objects. Masked and non-masked records are available.
- **IP Reputation Data Feed**—A set of IP addresses with context that cover different categories of suspicious and malicious hosts.
- **Malicious Hash Data Feed**—A set of file hashes with context that cover the most dangerous, prevalent, or emerging malware.
- **Malicious URL Data Feed**—A set of URLs with context that cover malicious websites and web pages. Masked and non-masked records are available.
- **Mobile Botnet Data Feed**—A set of URLs with context that cover mobile botnet C&C servers.
- **Mobile Malicious Hash Data Feed**—A set of file hashes with context for detecting malicious objects that infect mobile Google Android and Apple iPhone devices.
- **P-SMS Trojan Data Feed**—A set of Trojan hashes with context for detecting SMS Trojans that send premium-rate SMS messages to mobile users as well as enable attackers to steal, delete, and respond to SMS messages.
- **Phishing URL Data Feed**—A set of URLs with context that cover phishing websites and web pages. Masked and non-masked records are available.
- **Ransomware URL Data Feed**—A set of URLs, domains, and hosts with context that cover ransomware links and websites.
- **APT Hash Data Feed**—A set of hashes that cover malicious artifacts used by advanced persistent threat (APT) actors to conduct APT campaigns.
- **APT IP Data Feed**—A set of IP addresses that belong to the infrastructure used in APT campaigns.
- **APT URL Data Feed**—A set of domains that belong to the infrastructure used in APT campaigns.

Every record in a Data Feed is enriched with actionable context (threat names, time stamps, geolocation, resolved IP addresses of infected web resources, hashes, popularity, and so forth).

Kaspersky CyberTrace is a threat intelligence fusion and analysis tool that integrates threat data feeds with SIEM solutions. Users can immediately leverage threat intelligence for security monitoring and incident response (IR) activities in the workflow of their existing security operations.

Kaspersky CyberTrace uses continuously updated Kaspersky Threat Data Feeds to identify existing breaches or newly launched attacks, and to inform your business or clients about risks and implications associated with the threat.



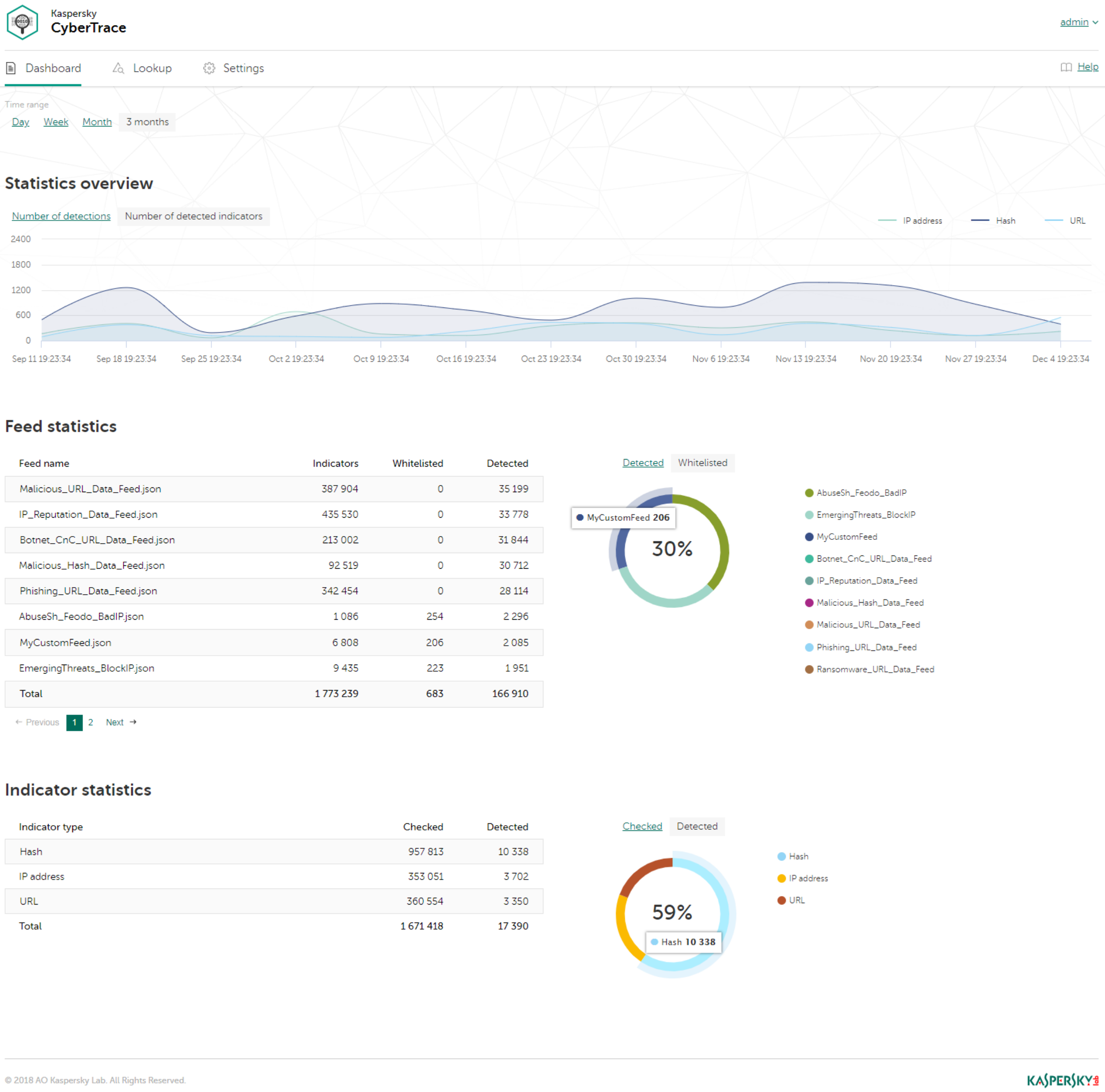
Indicators of compromise (IoCs) from Kaspersky Threat Data Feeds are not loaded into your SIEM instance, but instead are processed by Kaspersky CyberTrace in a separate offline process running on your infrastructure.
Because the task of matching events against a large number of IoCs is offloaded, your SIEM instance incurs a minimal reduction in performance. In case of a match, rich contextual information about the incident is passed to the SIEM instance and displayed on the dashboard.

Key features of Kaspersky CyberTrace

Kaspersky CyberTrace key features include the following:

- Kaspersky CyberTrace is flexible and can be easily integrated into the existing infrastructure, which allows you to avoid the challenges of integrating threat intelligence feeds with RSA NetWitness. Kaspersky CyberTrace integrates with any threat intelligence feed you might want to use (threat intelligence feeds from Kaspersky Lab or other vendors, OSINT feeds, or your custom feeds) and uses all these feeds together.
- Kaspersky CyberTrace does not hinder the performance of existing security controls and does not miss detections. The process of parsing and matching incoming data happens inside Kaspersky CyberTrace. This reduces the load on the existing SIEM solution. Kaspersky CyberTrace parses the incoming logs and events, matches the resulting data against feeds, and generates its own alerts when threats are detected.

- Kaspersky CyberTrace helps choose superior intelligence sources. Kaspersky CyberTrace helps evaluate the effectiveness of the integrated threat intelligence data feeds by providing detailed statistics on detections and allowing analysts to compare different threat intelligence sources in terms of their relevance to the company.
- Kaspersky CyberTrace helps to reduce the frequency of false positives. By using Kaspersky CyberTrace, analysts can fight false positives by whitelisting certain IoCs and filtering threat intelligence feeds according to configurable filtering rules.



Statistics overview

Feed statistics

Feed name	Indicators	Whitelisted	Detected
Malicious_URL_Data_Feed.json	387 904	0	35 199
IP_Reputation_Data_Feed.json	435 530	0	33 778
Botnet_CnC_URL_Data_Feed.json	213 002	0	31 844
Malicious_Hash_Data_Feed.json	92 519	0	30 712
Phishing_URL_Data_Feed.json	342 454	0	28 114
AbuseSh_Feodo_BadIP.json	1 086	254	2 296
MyCustomFeed.json	6 808	206	2 085
EmergingThreats_BlockIP.json	9 435	223	1 951
Total	1 773 239	683	166 910

Indicator statistics

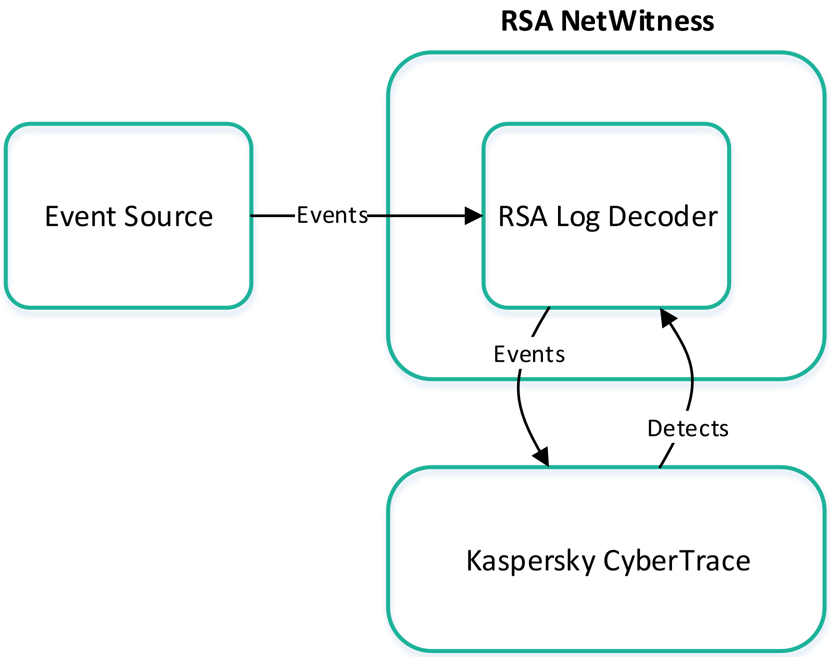
Indicator type	Checked	Detected
Hash	957 813	10 338
IP address	353 051	3 702
URL	360 554	3 350
Total	1 671 418	17 390

The following video explains how Kaspersky CyberTrace works: https://youtu.be/Ug0q0EDsTGU

RSA NetWitness Configuration

Setting up communication between Kaspersky CyberTrace and RSA NetWitness involves the following stages:

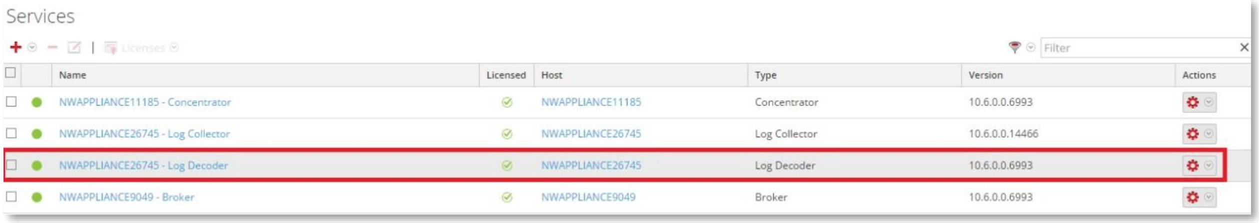
- Configuring RSA NetWitness to forward events to Kaspersky CyberTrace
- Configuring RSA NetWitness to receive events from Kaspersky CyberTrace
- Configuring and starting Kaspersky CyberTrace



Configuring RSA NetWitness to forward events to Kaspersky CyberTrace

To configure event forwarding from RSA NetWitness to Kaspersky CyberTrace:

1. In the RSA NetWitness main window, select **Administration** > **Services**.
2. In the **Services** table, in the **Name** column, select the relevant Log Decoder (the Log Decoder that receives events containing URLs, hashes, or IP addresses).



Services

☐		Name	Licensed	Host	Type	Version	Actions
☐	●	NWAPPLIANCE11185 - Concentrator	✓	NWAPPLIANCE11185	Concentrator	10.6.0.0.6993	
☐	●	NWAPPLIANCE26745 - Log Collector	✓	NWAPPLIANCE26745	Log Collector	10.6.0.0.14466	
☐	●	NWAPPLIANCE26745 - Log Decoder	✓	NWAPPLIANCE26745	Log Decoder	10.6.0.0.6993	
☐	●	NWAPPLIANCE9049 - Broker	✓	NWAPPLIANCE9049	Broker	10.6.0.0.6993	

Note: If more than one Log Decoder is used for receiving events, repeat the following steps for each Log Decoder.

3. For the selected Log Decoder, in the **Actions** column, click the **Settings** (⚙ ⌄) split button and in the drop-down list select **View** > **Config**.

4. Select the **App Rules** tab and click the **Add** button (+).

   The **Rule Editor** dialog box opens.

5. Specify the following data:

   - Rule Name: `cybertrace`
   - Condition:

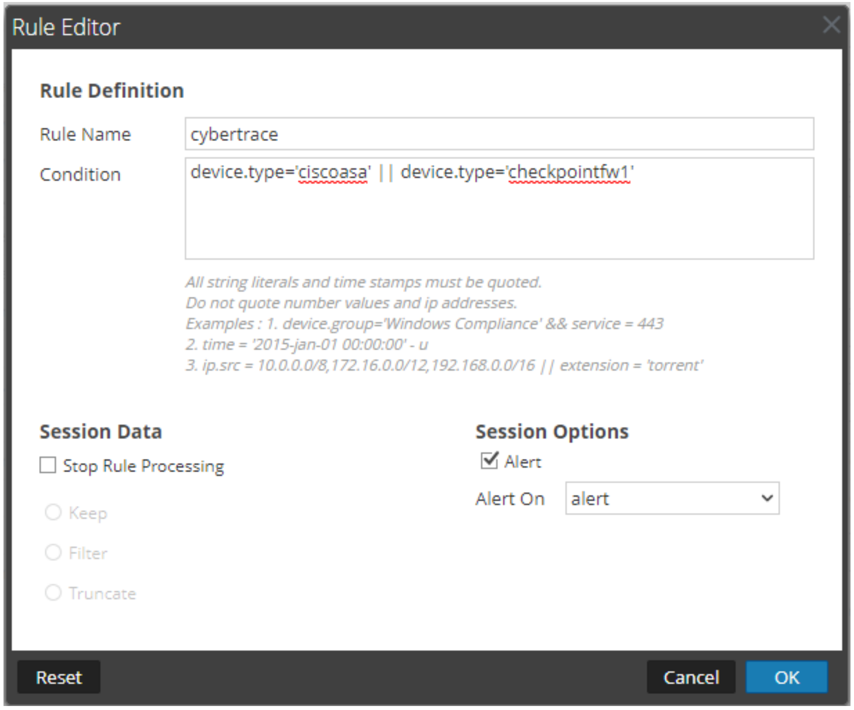
     ```
     device.type='%DEVICE_NAME_1%'
     ```

     This is an example of a condition in which the `%DEVICE_NAME_1%` string represents the name of the device whose events must be sent to Kaspersky CyberTrace. Following is another example of a condition, according to which events from Cisco ASA and Check Point Firewall must be sent to Kaspersky CyberTrace:

     ```
     device.type='ciscoasa' || device.type='checkpointfw1'
     ```

     For more information on how to create RSA rules, refer to **Rule and Query Guidelines** in the RSA NetWitness product documentation.
   - Under **Session Options**, select the **Alert** check box.

     Please note that in RSA NetWitness version 10 you must select the **Forward** check box, too.



Rule Name: cybertrace

Condition: device.type='ciscoasa' || device.type='checkpointfw1'

All string literals and time stamps must be quoted.
Do not quote number values and ip addresses.
Examples : 1. device.group='Windows Compliance' && service = 443
2. time = '2015-jan-01 00:00:00' - u
3. ip.src = 10.0.0.0/8,172.16.0.0/12,192.168.0.0/16 || extension = 'torrent'

□ Stop Rule Processing

○ Keep

○ Filter

○ Truncate

☑ Alert

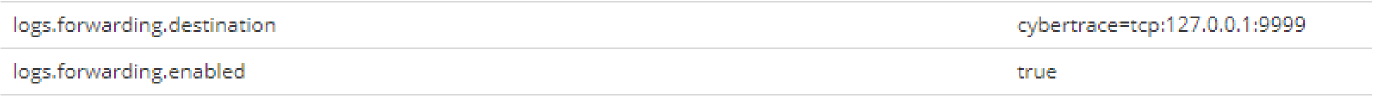
Alert On: alert

Pic. 1 Rule Editor in RSA 11

6. Click **OK**.
7. Click **Settings** (⚙ ⌄) and select **View** > **Explore**.
8. For the **/decoder/config/logs.forwarding.destination** setting, specify the destination:

   `cybertrace=tcp:%IP%:9999`

   Substitute `%IP%` with the IP address of the computer on which Kaspersky CyberTrace will be installed. By default, Kaspersky CyberTrace uses port 9999 to receive events.
9. For the **/decoder/config/logs.forwarding.enabled** setting, specify `true`.



logs.forwarding.destination	cybertrace=tcp:127.0.0.1:9999
logs.forwarding.enabled	true

After these actions are performed, RSA NetWitness will forward events that meet the `cybertrace` rule to the `%IP%:9999` address.

Configuring RSA NetWitness to receive events from Kaspersky CyberTrace

To configure the receipt of Kaspersky CyberTrace events (that match some records in feeds) by RSA NetWitness:

1. Download and deploy Kaspersky CyberTrace for RSA NetWitness. Kaspersky CyberTrace is available as an RPM package, DEP package, or TAR archive, depending on your preferences. You can download Kaspersky CyberTrace using this **link**.

   > **Note:** By default, Kaspersky CyberTrace contains a certificate for the demo version of Kaspersky Threat Data Feeds. These feeds do not require a commercial certificate. Demo feeds provide lower detection rates in comparison with their corresponding commercial versions. To obtain a certificate for the commercial version of Kaspersky Threat Data Feeds, contact the Kaspersky Cybersecurity Service team **(**intelligence@kaspersky.com**)**.

2. In the `/etc/netwitness/ng/envision/etc/devices` directory (of your SIEM instance), create a `cybertrace` subdirectory.
3. In the installation directory, go to the `/integration/cybertrace` subdirectory that contains the parser files and the export package for rules and dashboards.
4. Copy the following files from this subdirectory to the `/etc/netwitness/ng/envision/etc/devices/cybertrace` directory:

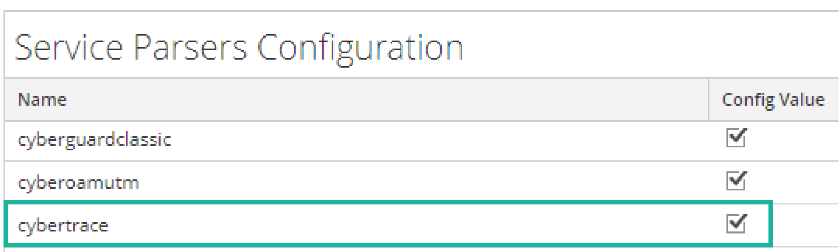
   `cybertrace.ini`—Configuration file that contains declaration of Kaspersky CyberTrace for RSA NetWitness.

   `v20_cybertracemsg.xml`—Configuration file that contains parsing rules for events that are sent from Kaspersky CyberTrace to RSA NetWitness.

5. Restart RSA NetWitness Log Decoder.

   In the **Administration/Services** table, for the selected Log Decoder, click **Settings** (⚙ ⌄) and select **Restart** from the drop-down list.

   > **Note:** Once restarted, make sure that the cybertrace parser is enabled in the Service Parsers Configuration list of RSA NetWitness Log Decoder.



Service Parsers Configuration

Name	Config Value
cyberguardclassic	☑
cyberoamutm	☑
cybertrace	☑

In the v20_cybertracemsg.xml file, the format of events from Kaspersky CyberTrace is provided in the `HEADER/content` element and in the `MESSAGE/content` element. Make sure that all fields mentioned in the `MESSAGE/content` element (except `msg`) are present in the index files of RSA NetWitness Concentrator (index-concentrator-custom.xml). Also, make sure that the value of the `flags` attribute is `None` for each of these fields in the table-map-custom.xml file.

The tables in **Appendix A** and **Appendix B** describe the fields used in the v20_cybertracemsg.xml and index-concentrator-custom.xml files.

Configuring and starting Kaspersky CyberTrace

Kaspersky CyberTrace for RSA NetWitness sends two types of events:

- Alert events (for example, KL_ALERT_ServiceStarted)
- Detection events (in case there is a match with feeds)

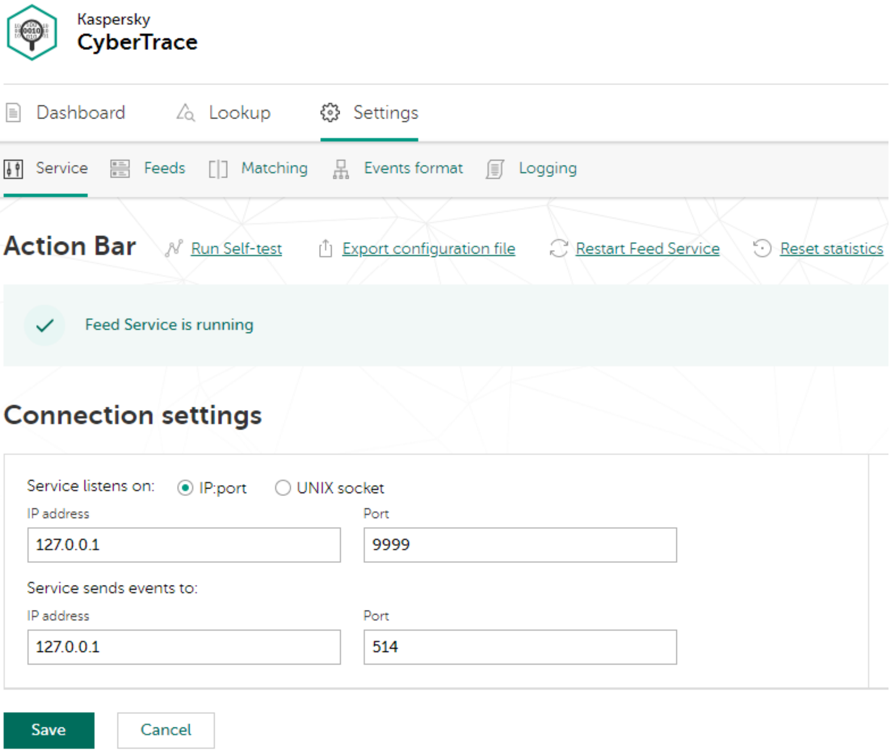
It is recommended that you use Kaspersky CyberTrace Web to configure Kaspersky CyberTrace. You can enable Kaspersky CyberTrace Web during **installation** of Kaspersky CyberTrace. Or, you can configure Kaspersky CyberTrace manually (see information about manual configuration in the Kaspersky CyberTrace **documentation**).

To configure Kaspersky CyberTrace for sending events to RSA NetWitness:

1. In Kaspersky CyberTrace Web, open the **Settings** > **Service** page and specify the following value in the **Service sends events to** box:

   ```
   IP address: %IP% Port: 514
   ```

   Substitute `%IP%` with the IP address of the computer on which RSA NetWitness Log Decoder is installed.
2. Save the changes.
3. Restart Kaspersky CyberTrace.



Feed Service is running

Connection settings

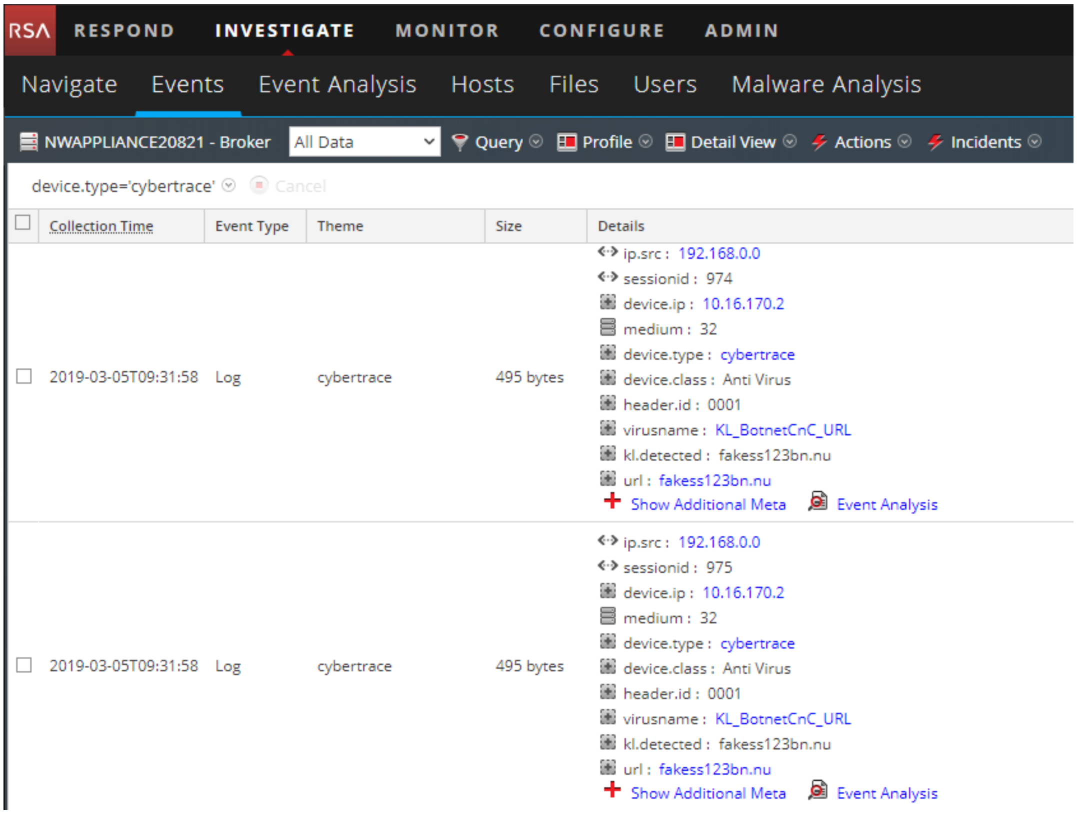
Service listens on: ● IP:port ○ UNIX socket

IP address: 127.0.0.1 Port: 9999

Service sends events to:

IP address: 127.0.0.1 Port: 514

After Kaspersky CyberTrace is properly configured, RSA NetWitness analysts will see events—originated from Kaspersky CyberTrace (device.type = 'cybertrace')—in RSA NetWitness Investigator, as shown in the figure below.



RSA RESPOND INVESTIGATE MONITOR CONFIGURE ADMIN

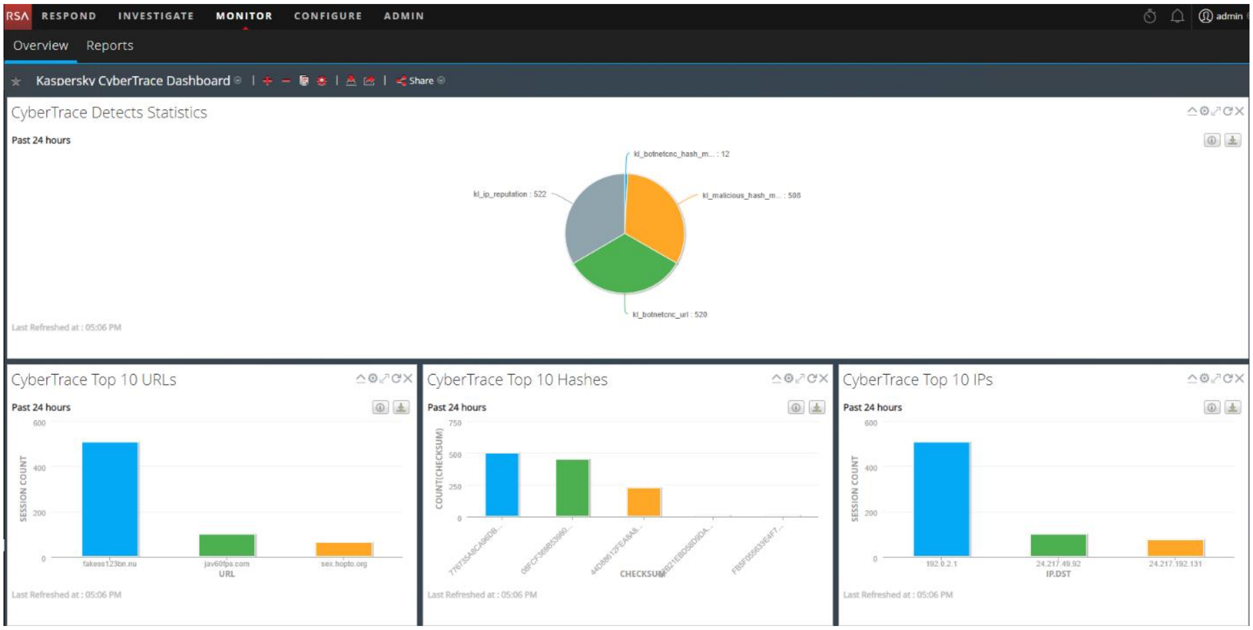
Navigate Events Event Analysis Hosts Files Users Malware Analysis

NWAPPLIANCE20821 - Broker | All Data | Query | Profile | Detail View | Actions | Incidents

device.type='cybertrace' | Cancel

□	Collection Time	Event Type	Theme	Size	Details
□	2019-03-05T09:31:58	Log	cybertrace	495 bytes	ip.src : 192.168.0.0 sessionid : 974 device.ip : 10.16.170.2 medium : 32 device.type : cybertrace device.class : Anti Virus header.id : 0001 virusname : KL_BotnetCnC_URL kl.detected : fakess123bn.nu url : fakess123bn.nu + Show Additional Meta Event Analysis
□	2019-03-05T09:31:58	Log	cybertrace	495 bytes	ip.src : 192.168.0.0 sessionid : 975 device.ip : 10.16.170.2 medium : 32 device.type : cybertrace device.class : Anti Virus header.id : 0001 virusname : KL_BotnetCnC_URL kl.detected : fakess123bn.nu url : fakess123bn.nu + Show Additional Meta Event Analysis

The RSA NetWitness interface can also be supplemented with dashboards that are relevant to Kaspersky CyberTrace:



Certification Checklist for RSA NetWitness

Date Tested: April 4, 2017

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA NetWitness	10.6.2, 11.2.0.0	Virtual appliance
Kaspersky CyberTrace	3.0.0.386 or later	SaaS

Security Analytics Test Case	Result
Investigation	
Threat Intelligence Feed is received through Decoder Meta	✓
Threat Intelligence Feed is received through Packet Decoder	✓

✓ = Pass ✗ = Fail N/A = Non-available function

Appendix A

Field	Description
action	Kaspersky CyberTrace alert event (for example, KL_ALERT_ServiceStarted)
msg	Additional information about the Kaspersky CyberTrace alert event
virusname	Category of the object detected by Kaspersky CyberTrace
url	URL specified in the event forwarded by RSA NetWitness
checksum	Hash specified in the event forwarded by RSA NetWitness
daddr	Destination IP address specified in the event forwarded by RSA NetWitness
saddr	Source IP address specified in the event forwarded by RSA NetWitness
hostip	Device IP address specified in the event forwarded by RSA NetWitness
event_source	Name of the device that has sent the event (specified in the event forwarded by RSA NetWitness)
c_username	Name of the user under whose account the activity specified in the event is performed
fld1	Context of the feed record involved in the detection process

Appendix B

If you want to see the context from Kaspersky Lab Data Feeds in **separate fields**, add the following elements to the table-map-custom.xml and index-concentrator-custom.xml RSA NetWitness configuration files:

- To the table-map-custom.xml file add the following entries:

```
<mapping envisionName="kl_detected_indicator" nwName="kl.detected" flags="None"/>
<mapping envisionName="kl_mask" nwName="kl.mask" flags="None"/>
<mapping envisionName="kl_ip" nwName="kl.ip" flags="None"/>
<mapping envisionName="kl_category" nwName="kl.category" flags="None"/>
<mapping envisionName="kl_first_seen" nwName="kl.first_seen" flags="None"/>
<mapping envisionName="kl_last_seen" nwName="kl.last_seen" flags="None"/>
<mapping envisionName="kl_popularity" nwName="kl.popularity" flags="None"/>
<mapping envisionName="kl_threat" nwName="kl.threat" flags="None"/>
<mapping envisionName="kl_industry" nwName="kl.industry" flags="None"/>
<mapping envisionName="kl_threat_score" nwName="kl.threat_score" flags="None"/>
<mapping envisionName="kl_file_size" nwName="kl.file_size" flags="None"/>
<mapping envisionName="kl_file_type" nwName="kl.file_type" flags="None"/>
<mapping envisionName="kl_behaviour" nwName="kl.behaviour" flags="None"/>
<mapping envisionName="kl_verdict" nwName="kl.verdict" flags="None"/>
<mapping envisionName="kl_pub_name" nwName="kl.pub_name" flags="None"/>
<mapping envisionName="kl_detection_date" nwName="kl.detect_date" flags="None"/>
<mapping envisionName="kl_md5" nwName="kl.md5" flags="None"/>
<mapping envisionName="kl_sha1" nwName="kl.sha1" flags="None"/>
<mapping envisionName="kl_sha2" nwName="kl.sha2" flags="None"/>
```

- To the index-concentrator-custom.xml file add the following entries:

```
<key description="kl_detected_indicator" format="Text" level="IndexKeys"
name="kl.detected" defaultAction="Open"/>
<key description="kl_mask" format="Text" level="IndexKeys" name="kl.mask"
defaultAction="Open"/>
<key description="kl_ip" format="IPv4" level="IndexKeys" name="kl.ip"
defaultAction="Open"/>
<key description="kl_category" format="Text" level="IndexKeys" name="kl.category"
defaultAction="Open"/>
<key description="kl_first_seen" format="Text" level="IndexKeys" name="kl.first_seen"
defaultAction="Open"/>
<key description="kl_last_seen" format="Text" level="IndexKeys" name="kl.last_seen"
defaultAction="Open"/>
<key description="kl_popularity" format="UInt8" level="IndexKeys"
name="kl.popularity" defaultAction="Open"/>
<key description="kl_threat" format="Text" level="IndexKeys" name="kl.threat"
defaultAction="Open"/>
<key description="kl_industry" format="Text" level="IndexKeys" name="kl.industry"
defaultAction="Open"/>
<key description="kl_threat_score" format="UInt8" level="IndexKeys"
name="kl.threat_score" defaultAction="Open"/>
<key description="kl_file_size" format="UInt16" level="IndexKeys" name="kl.file_size"
defaultAction="Open"/>
```

```
<key description="kl_file_type" format="Text" level="IndexKeys" name="kl.file_type"
defaultAction="Open"/>
<key description="kl_behaviour" format="Text" level="IndexKeys" name="kl.behaviour"
defaultAction="Open"/>
<key description="kl_verdict" format="Text" level="IndexKeys" name="kl.verdict"
defaultAction="Open"/>
<key description="kl_pub_name" format="Text" level="IndexKeys" name="kl.pub_name"
defaultAction="Open"/>
<key description="kl_detection_date" format="Text" level="IndexKeys"
name="kl.detect_date" defaultAction="Open"/>
<key description="kl_md5" format="Text" level="IndexKeys" name="kl.md5"
defaultAction="Open"/>
<key description="kl_sha1" format="Text" level="IndexKeys" name="kl.sha1"
defaultAction="Open"/>
<key description="kl_sha2" format="Text" level="IndexKeys" name="kl.sha2"
defaultAction="Open"/>
```