Using YARA Rules with RiskVision

YARA Rules | TRITON RiskVision | 02-Jun-2016

YARA is a tool often used by malware researchers for identifying and classifying content based on textural or binary patterns. It supports a comprehensive set of rules using wild-cards, case-insensitive strings, regular expressions, special operators and many other complex and powerful features.

The analytic tools used by RiskVision include a database of YARA rules used during Local Analysis (before files are sent for sandboxing or other external analysis). If your organization uses YARA, you can configure RiskVision to use your YARA rules in combination with its own (see *Adding YARA rules to RiskVision*, page 2).

RiskVision applies YARA rules to both inbound and outbound HTTP and SMTP traffic to:

1. Identify malicious content based on signatures.
2. Identify protocol applications.
3. Identify certain patterns in the headers of the request and responses.

Matching rules can be used to assign labels to transactions to specify how those transactions are processed by other RiskVision plugins (see *Rules and labels*, page 1).

Rules and labels

YARA Rules | TRITON RiskVision | 02-Jun-2016

If a transaction analyzed by RiskVision contains patterns that match a YARA rule, one or more labels can be appended to the transaction. For your own YARA rules, you can specify the labels that are added.

- In most cases, if a transaction contains a YARA rule match, but no other analytic flags the transaction as an incident, the transaction is discarded, and no record is added to the Configuration and Reporting Database.
- If a transaction contains a YARA rule match, and another analytic flags the transaction as an incident, the labels appended to the transaction by the YARA Plugin are recorded as part of the incident.

  These labels are listed in the Transaction Viewer's Details pane on the Incidents page in the RiskVision Local Manager.

- If the YARA Plugin applies the OffBoxScanRequired label to a transaction, even if no other analytic flags the transaction as an incident, an incident record is added to the Configuration and Reporting Database.

Adding YARA rules to RiskVision

YARA Rules | TRITON RiskVision | 02-Jun-2016

RiskVision downloads YARA rules from Forcepoint download servers and saves them in the **/opt/websense/yara/download** directory on the appliance.

Note

Currently, the downloaded YARA rules are used only to identify macros in text files and forward them to Threat Protection Appliance for further analysis.

- RiskVision checks for updates daily.
- Downloaded YARA rules are encrypted, and cannot be edited directly by RiskVision administrators.
- Administrators can prompt RiskVision to reload all of its YARA rules by creating an empty file called **update** in the **/opt/websense/yara/** directory.
  - The update occurs within 5 minutes, and does not require restarting any services.
  - When the update is complete, the YARA plugin deletes the empty “update” file.

Add your own custom rules by copying them to the appropriate directory on the RiskVision appliance:

- For rules that match data in HTTP headers:

  ```
  /opt/websense/yara/custom/header/
  ```

  Because SMTP transactions do not have request and response headers, header rules are not applied to SMTP traffic.
- For rules that match data in files associated with HTTP and SMTP transactions:

  ```
  /opt/websense/yara/custom/content
  ```

Creating header rules

To prompt RiskVision to add labels to a header rule, add the following line to the “meta” section of the rule:

```
txnLabel = "Label_that_you_want_to_add"
```

A header rule may contain as many labels as needed. Specify each label on a separate line.

The result will look something like this:

```
rule sample_rule {
   meta:
      author = "Authoring_Entity"
      description = "What does this rule do?"
      date = "yyyy-mm"
      txnLabel = "CustomLabel"
   strings:
      $s1 = "First string to match"
      $s2 = "Second string to match"
      $s3 = "Third string to match"
   condition:
      $s1 and ($s2 or $s3)
}
```

For example:

```
rule header_malicious_rule {
   meta:
      author = "TRITON RiskVision"
      description = "Detects malware sample files"
      date = "2015-11"
      version = "0.1"
      txnLabel = "Detected by YARA"
   strings:
      $s1 = "testdatabasewebsense.com"
      $s2 = "malicioustest2.exe"
      $s3 = "maliciousRIAtest.swf"
      $s4 = "wbsn-ts-test-1_sbx_test.exe"
   condition:
      $s1 and ($s2 or $s3 or $s4)
}
```

The sample header rule is a match if the YARA plugin finds the string "testdatabasewebsense.com" followed by any one of the following strings: malicioustest2.exe, maliciousRIAtest.swf, or wbsn-ts-test-1_sbx_test.exe.

Since the URL is in the header of HTTP request, this rule is matched when a monitored user visits any of the following URLs:

```
http://testdatabasewebsense.com/realtime/maliciouswebsites/
malicioustest2.exe
http://testdatabasewebsense.com/realtime/maliciouswebsites/
maliciousRIAtest.swf
```

```
http://testdatabasewebsense.com/threatscope/wbsn-ts-test-
1_sbx_test.exe
```

Creating content rules

To prompt RiskVision to add labels to a content rule, add the following line to the "meta" section of the rule:

```
dataLabel = "Label_that_you_want_to_add"
```

A content rule may contain as many labels as needed. Specify each label on a separate line.

For example:

```
rule content_malicious_rule {
   meta:
      author = "TRITON RiskVision"
      description = "Malicious app sample"
      datalabel = "Detected by YARA"
   strings:
      $s1 = { 5668c4d3 }
      $s2 = "KERNEL32.dll" wide ascii
   condition:
      all of them
}
```

The sample rule above is matched if the YARA plugin finds **both** of the strings specified ($s1 and $s2) in the payload content.

This rule is matched when the "maliciousapp.exe" sample file is sent or received in a monitored SMTP transaction, or when a monitored user visits:

```
http://testdatabasewebsense.com/threatscope/maliciousapp.exe
```