paloalto TECHDOCS

# Custom Application IDs and Signatures

docs.paloaltonetworks.com

## Contact Information

Corporate Headquarters:
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
www.paloaltonetworks.com/company/contact-support

## About the Documentation

- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal docs.paloaltonetworks.com.
- To search for a specific topic, go to our search page docs.paloaltonetworks.com/search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

## Copyright



## Last Revised

May 31, 2024

# Table of Contents



# Custom Application and Threat Signatures

Our next-generation firewalls allow you to develop custom application and threat signatures for network traffic you want to detect, monitor, and control. You can build these pattern-based signatures using information from packet captures and our available contexts. The firewall stores the custom signatures in a database separate from our predefined App-ID™ or threat signatures, which are updated on a regular basis.

Application signatures identify web-based and client-server applications such as Gmail. You can create custom application signatures for proprietary applications, commercial applications without an App-ID, or traffic you want to identify by a custom name. Threat signatures detect malicious activity and prevent network-based attacks. You can create custom threat signatures to incorporate third-party security advisories and signatures or to identify threat activity such as brute force login attempts. The resulting application and threat visibility allows you to exercise a greater level of control over network traffic and reduces the attack surface of your enterprise.

*Weekly content releases periodically include new decoders and contexts from which you can develop signatures.*

# About Custom Application Signatures

Custom application signatures reduce unknown traffic, provide application visibility, and give you more granular control over applications on your network. For example, you may believe office productivity has decreased since the FIFA Women's World Cup began. You can create custom signatures for the FIFA landing and live streaming pages and view FIFA activity in the ACC and Traffic logs (as long as current security policies allow the traffic). From there, you can create a report, configure a QoS policy, or block the application by adding it to security policy.

An application signature identifies a pattern located within packets from an application or application function. This pattern uniquely identifies the application or function of interest. The App-ID™ traffic classification system relies on application signatures to accurately identify applications in your network. Palo Alto Networks has developed App-ID signatures for many well-known applications. (See Applipedia for a complete list). However, the volume of commercial applications and the nature of internal applications means that some applications do not have a signature. Such traffic receives "unknown" classification in the ACC and Traffic logs alongside potential threats. To properly classify this traffic and enforce security policy rules, you can create a custom application signature.

Custom application signatures enable you to:

- Minimize "unknown" traffic on your network
  - Identify internal applications or special interest applications, such as a custom payroll application or sports live streaming
- Monitor application usage in the ACC and Traffic logs
- Explicitly define allowed applications and application functions (for example, allowing Slack for instant messaging, but blocking file transfer)
- Perform QoS for a specific application
- Identify nested applications, such as Words with Friends in Facebook

*Custom applications take precedence over predefined applications when traffic matches both a custom-defined signature and a Palo Alto Networks signature. Accordingly, Traffic logs reflect the custom application name once the new application has been configured.*

# About Custom Threat Signatures

Our next-generation firewalls allow you to create custom threat signatures to monitor malicious activity or integrate third-party signatures. As with Palo Alto Networks threat signatures, you can detect, monitor, and prevent network-based attacks with custom threat signatures. Build your signature by examining packet captures for regular expression patterns that uniquely identify spyware activity and vulnerability exploits. The firewall will scan network traffic for these patterns and act based on the action specified during configuration upon threat detection. Be sure to use custom threat signatures as part of anti-spyware and vulnerability protection profiles to detect and handle command-and-control (C2) activity and system flaws that an attacker might attempt to exploit.

You can also define a combination signature for brute force attacks—a custom threat signature that triggers when traffic matches a specified pattern a certain number of times in a given time interval.

- Create a Custom Threat Signature
- Create a Combination Signature

## Combination Signatures for Brute Force Attacks

Combination signatures detect and prevent brute force attacks. A combination signature assigns a time attribute to an existing threat signature—the child signature—to form a distinct parent signature. The time attribute specifies the number of pattern matches or “hits” to the child signature and the time frame (in seconds) the hits must occur within for the parent signature to trigger. If a pattern matches the child signature alone, the default action for that signature occurs.

You can narrow the trigger conditions by including aggregation criteria, which define what the parent signature counts as a hit. You can select from “source,” “destination,” and “source-and destination.” If you wanted to count the number of hits to a particular destination IP address, you would set the aggregation criteria to “destination.” To count all hits from a particular source, select “source.” “Source-and-destination” instantiates multiple time-windows that count the n-number of instances when a single source goes to a specific destination.

- About Custom Threat Signatures
- Create a Combination Signature

# Create a Custom Application Signature

To create a custom application signature, you must do the following:

- Research the application using packet capture and analyzer tools
- Identify patterns in the packet captures
- Build your signature
- Validate your signature

Custom application signatures require you to specify the **Scope**—how your signature is applied to the traffic, **Context**—the portion of the file or protocol where you expect to find your pattern, the **Pattern**, and the **Operator** (**Pattern Match** for string contexts and **Greater Than**, **Less Than**, or **Equal To** for integer-based contexts).

Refer to the Custom Signature Contexts, Defining Applications and Syntax for Regular Expression Data Patterns while building your signature.

Tutorial: How to Configure a Custom App-ID

**STEP 1 |** Research the application using packet capture and/or analyzer tools.

- You should understand how you'd like to control the application before all else. Do you want to limit application functionality? Create a usage report? You'll want to examine the contents of packet captures to gather context and identify unique characteristics of the application.

  *Consider using a tool such as Wireshark or perform a packet capture on the firewall itself* Take a Packet Capture for Unknown Applications.

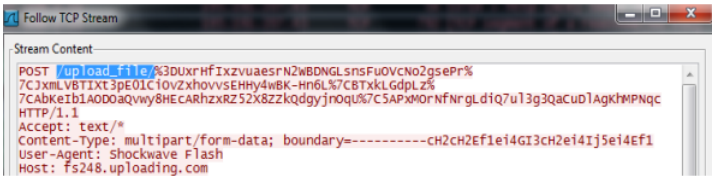
  1. Perform multiple packet captures between the client system and web server.

     Generate traffic for various application scenarios once you have launched the capture tool. For example, if you wanted to create a signature for 'uploading' on uploading.com, you would upload a file on that site.

     *Multiple sessions might be created for the different actions performed in the application. You will need to locate and inspect each type of session in the resulting packet captures.*

  2. Inspect packet captures for values or patterns that uniquely identify the application or application function.

     For example, after you uploaded a file to uploading.com, you would look for HTTP POST request packets in the sessions captured by your packet analyzer tool. Then, you would examine the packet contents for patterns.

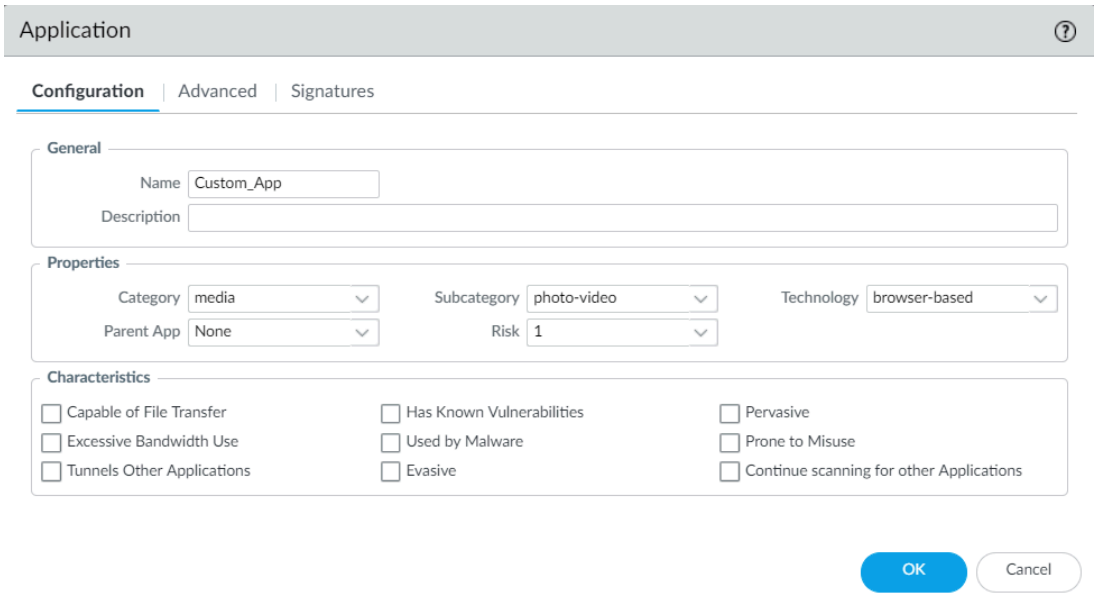


**STEP 2 |** Create the custom application.

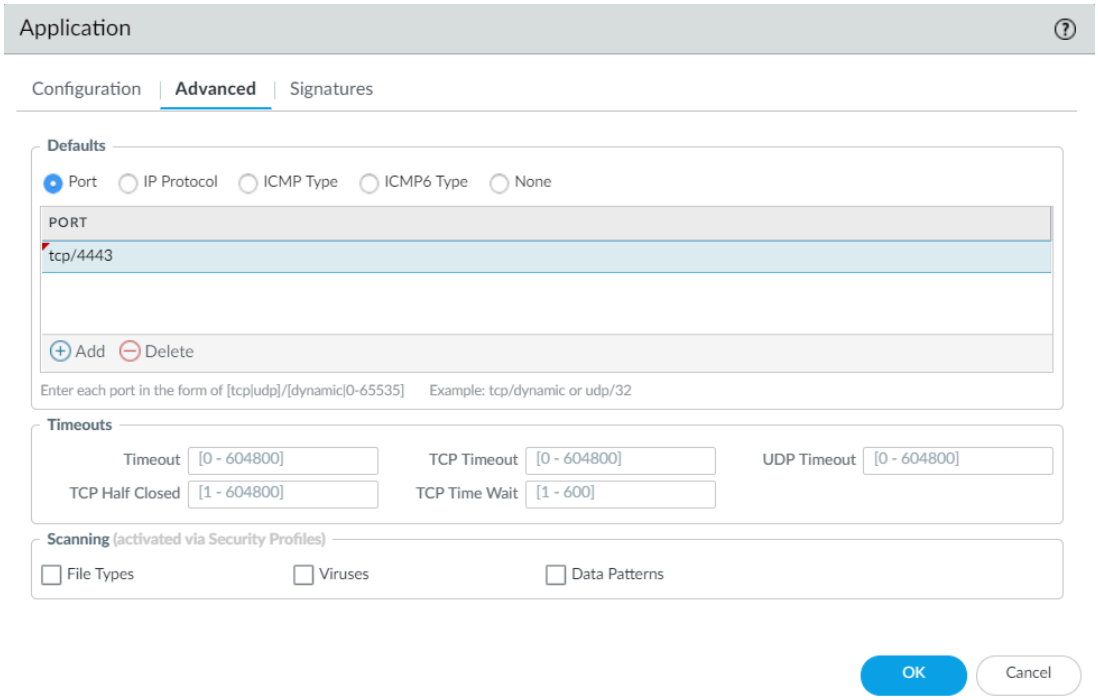
1. Select **Objects** > **Applications** and click **Add**.
2. Under **Configuration**, enter a name and optional description for the application. Specify the application's Properties and Characteristics.

   - *If your custom application has no Parent App that can be identified by regular App-ID or is used in an application override, the application cannot be scanned for threats.*
   - *If the custom application has scanning options unchecked, the threat engine will stop inspecting the traffic as soon as the custom application is identified.*




3. Under **Advanced**, define settings that will allow the firewall to identify the application protocol:
   - Specify the default ports or protocol that the application uses. To specify signatures independent of protocol, select None.
   - Specify the session timeout values. If you don't specify timeout values, the default timeout values will be used.
   - Indicate any type of additional scanning you plan to perform on the application traffic.

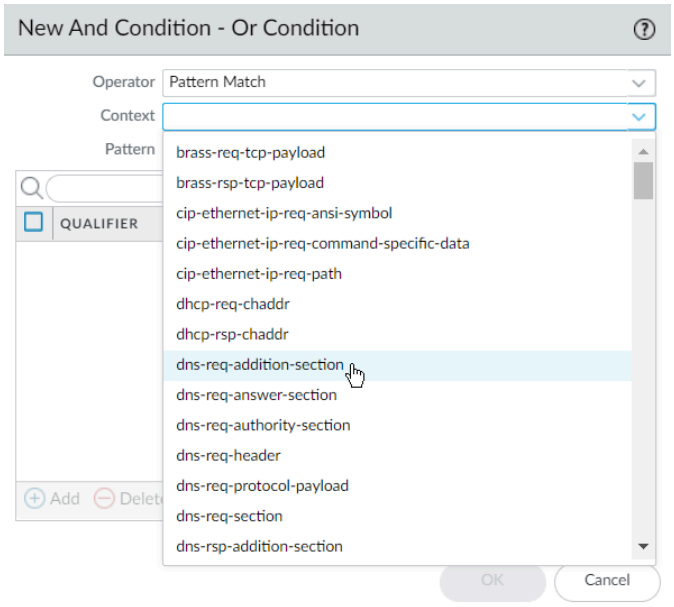


**STEP 3 |** Define your signature.

Multiple signatures may be necessary to account for all traffic specific to the application.

1. Under **Signatures**, click **Add** and enter a **Signature Name** and optional description.
2. Specify the **Scope**—Select between **Transaction** (e.g. HTTP request and response) or **Session** (e.g. a single POST request).
3. Specify the matching conditions by clicking **Add And Condition** or **Add Or Condition**.
4. Select an **Operator** to define the conditions that must be true for a signature to match traffic.
   - If you select **Pattern Match**, select a **Context** and then use a regular expression to specify the **Pattern**. Optionally, **Add** a qualifier/value pair.
   - *Qualifiers are context-dependent and limit the match condition for the given context. For example, you might use the http-method qualifier to specify that a http-req-uri-path only matters if it is found inside an HTTP GET method.*




   - If you select **Equal To**, **Less Than**, or **Greater Than**, select an integer **Context**, and enter a **Value**.
5. Repeat sub-steps 3 and 4 for each matching condition.

   If you leave **Ordered Condition Match** selected, make sure the condition or group of conditions is in the desired order. The most specific conditions should come first. To order the conditions: Select a condition or a group and click **Move Up** or **Move Down**.

   *You cannot move conditions from one group to another.*

**STEP 4 |** Save the custom signature.

1. Click **OK** to save your signature definition.
2. **Commit** your signature.

**STEP 5 |** Test your custom signature.

# Create a Custom Threat Signature

To create a custom threat signature, you must do the following:

- Research the application using packet capture and analyzer tools
- Identify patterns in the packet captures
- Build your signature
- Validate your signature

Be sure to Set Up Antivirus, Anti-Spyware, and Vulnerability Protection to specify how the firewall responds when it detects a threat.

Refer to the list of Custom Signature Contexts, Threat Details and Syntax for Regular Expression Data Patterns while building your signature.

Tutorial: Custom Vulnerability

To create a threat signature with time attributes, see create a combination signature.

**STEP 1 |** Add a custom threat.

1. Click **Objects** > **Custom Objects** > **Spyware/Vulnerability** and then click **Add**.
2. Under **Configuration**, fill out the following required fields in the General and Properties sections.
   - **Threat ID**
     - For a vulnerability signature, enter a numeric ID between 41000 and 45000. If the firewall runs PAN-OS 10.0 or later, the ID can also be between 6800001 and 6900000.
     - For a spyware signature, the ID should be between 15000 and 18000. If the firewall runs PAN-OS 10.0 or later, the ID can also be between 6900001 and 7000000.
   - **Name**—Specify the threat name.
   - **Severity**—Select the severity of the threat.

**STEP 2 |** Define your signature.

1. Under **Signatures**, leave **Standard** selected unless you wish to Create a Combination Signature. **Add** a new signature.
2. Specify the following information:
   - **Standard**—Enter a name to identify the signature.
   - **Comment**—Enter an optional description.
   - **Ordered Condition Match**—If the order in which the firewall attempts to match the signature definitions is important, make sure the check box is selected.
   - **Scope**—Indicate whether this signature applies to a full **Session** or a single **Transaction**.
3. Specify the matching conditions by clicking **Add And Condition** or **Add Or Condition**.
4. Select an **Operator** to define the conditions that must be true for a signature to match traffic.
   - If you select **Pattern Match**, specify the following:
     - **Context**—Select from available custom signature contexts.
     - **Pattern**—Use a regular expression to define this attribute.
     - Optionally, **Add** a qualifier/value pair.

       *Qualifiers are context-dependent and limit the match condition for the given context.*
   - Select **Negate** to signal a condition under which the custom signature does not trigger. The custom signature matches to traffic only when this condition is false.

     - *A custom signature cannot be created with only Negate conditions. You must include at least one positive condition in your definition.*
     - *If the signature's scope is set to Session, a negative condition cannot be configured as the last condition to match to traffic.*

   You can define exceptions for custom vulnerability or spyware signatures using the new option to negate signature generation when traffic matches both a signature and the exception to the signature. Use this option to allow certain traffic in your network that might otherwise be classified as spyware or a vulnerability exploit. In this case, the signature is generated for traffic that matches the pattern; traffic that matches the pattern but also matches the exception to the pattern is excluded from signature generation and any associated policy action (such as being blocked or dropped). For example, you can define a signature to be generated for redirected URLs; however, you

can now also create an exception where the signature is not generated for URLs that redirect to a trusted domain.

- If you select an **Equal To**, **Less Than**, or **Greater Than** operator, specify a **Context** and a **Value**.

5. Repeat sub-steps 3 and 4 for each matching condition.

   If you leave **Ordered Condition Match** selected, make sure the condition or group of conditions is in the desired order. The most specific conditions should come first. To order the conditions: Select a condition or a group and click **Move Up** or **Move Down**.

   *You cannot move conditions from one group to another.*

**STEP 3 |** Save the custom threat.

1. Click **OK** to save the custom threat.

**STEP 4 |** Enable your custom signature.

1. Go to **Security Profiles** > **Anti-Spyware/Vulnerability Protection** and select an existing profile.
2. Under **Exceptions**, **Show All Signatures**, enter the Threat ID you created, and **Enable** it.
3. Click **OK**.

**STEP 5 |** **Commit** your changes.

**STEP 6 |** Test your custom signature.

## Create a Combination Signature

You can create a combination signature to monitor the frequency and rate of matches to a signature on your network. You'll need to know the Threat ID of an existing threat signature or create a custom threat signature that detects a particular event such as a Wordpress login attempt. When you configure your combination signature, you'll have to specify the time conditions for matches to the threat—x number of hits in y number of seconds. You can adjust the time attribute according to needs and experience.

**STEP 1 |** Add a custom threat.

1. Click **Objects** > **Custom Objects** > **Spyware/Vulnerability** and then click **Add**.
2. Under **Configuration**, fill out the following required fields in the General and Properties sections.
   - **Threat ID**
     - For a vulnerability signature, enter a numeric ID between 41000 and 45000. If the firewall runs PAN-OS 10.0 or later, the ID can also be between 6800001 and 6900000.
     - For a spyware signature, the ID should be between 15000 and 18000. If the firewall runs PAN-OS 10.0 or later, the ID can also be between 6900001 and 7000000.
   - **Name**—Specify the threat name.
   - **Severity**—Select the severity of the threat.

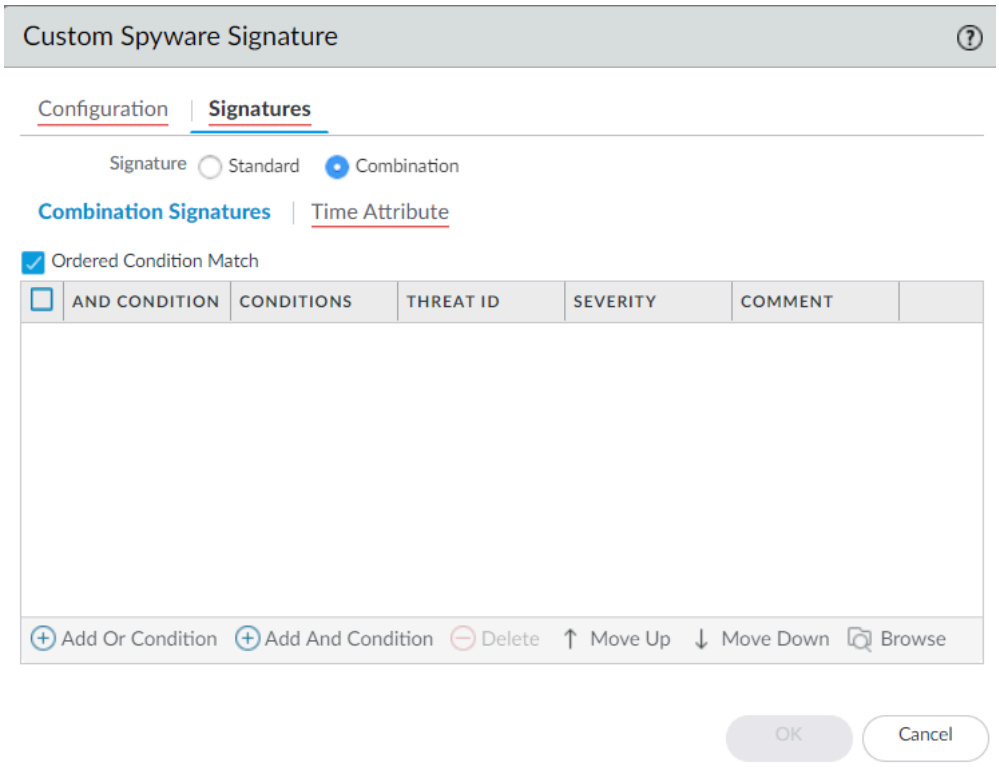
 

**STEP 2 |** Define your signature.

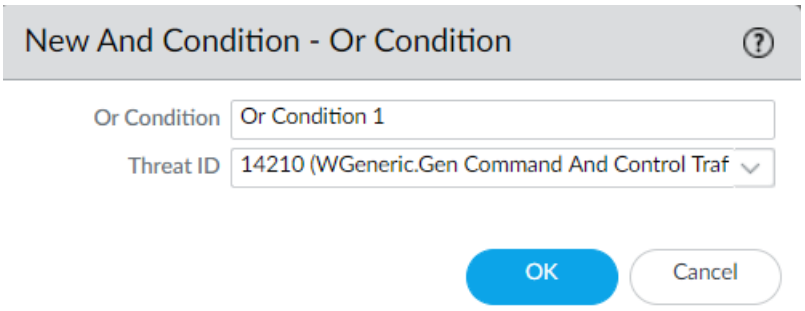
1. Click **Signatures** and select **Combination**.
2. Under **Combination Signatures**, click **Add And Condition** or **Add Or Condition**.
   - To add a condition within a group, select the group and click **Add Condition**.
   - To move a condition within a group, select the condition and click **Move Up** or **Move Down**.

     *You cannot move conditions from one group to another.*

   - To move a group, select the group and click **Move Up** or **Move Down**.




3. Choose the **Threat ID** for the signature you'd like to use. You may also edit the condition name.



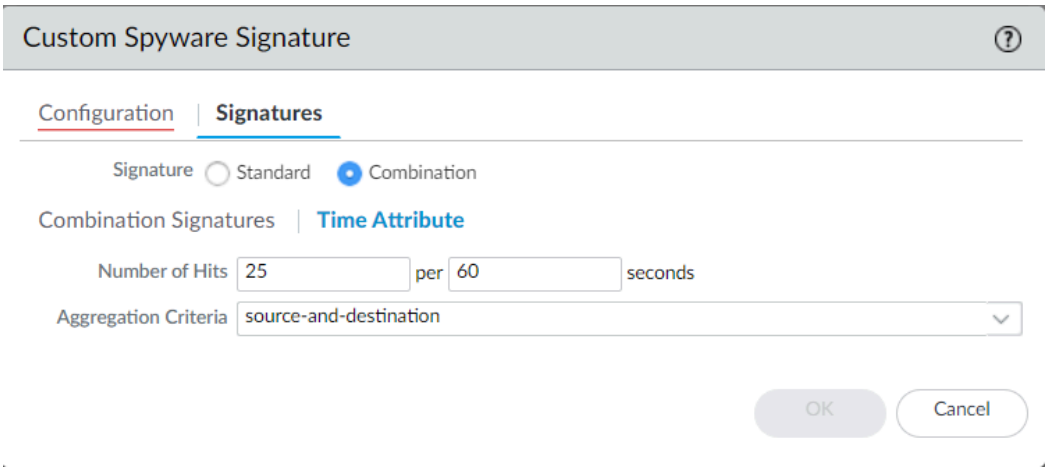

4. Under **Time Attribute** specify the following:

- **Number of Hits**—Specify the threshold that will trigger any policy-based action as a number of hits (1-1000) in a specified number of seconds (1-3600).
- **Aggregation Criteria**—Specify whether the hits are tracked by source IP address, destination IP address, or a combination of source and destination IP addresses.
- To move a condition within a group, select the condition and click **Move Up** or **Move Down**.

  *You cannot move conditions from one group to another.*

- To move a group, select the group and click **Move Up** or **Move Down**.




5. Repeat sub-steps 2, 3, and 4 for each matching condition.

   If you leave **Ordered Condition Match** selected, make sure the condition or group of conditions is in the desired order. The most specific conditions should come first. To order the conditions: Select a condition or a group and click **Move Up** or **Move Down**.

   *You cannot move conditions from one group to another.*

**STEP 3 |** Save the custom threat.

1. Click **OK** to save the custom threat.
2. **Commit** your signature(s).

**STEP 4 |** Test your custom signature.

## Create a Custom Threat Signature from a Snort Signature

The following steps illustrate the process for converting a Snort signature into a custom spyware signature compatible with Palo Alto Networks firewalls. The use case below uses a Snort rule for a North Korean Trojan malware variant as identified by the Department of Homeland Security, the Federal Bureau of Investigation, and other US government partners.

With Panorama version 10.0 or later, you can use the IPS Signature Converter plugin to automatically convert Snort and Suricata rules into custom Palo Networks threat signatures instead of manually performing the following procedure.

Snort rule:

```
alert tcp any any -> any any (msg:"Malformed_UA"; content:"User-
Agent: Mozillar/"; depth:500; sid:99999999;)
```

Reference: https://www.us-cert.gov/ncas/alerts/TA17-318B

IOC List: https://www.us-cert.gov/sites/default/files/publications/TA-17-318B-IOCs.csv

In this example you can:

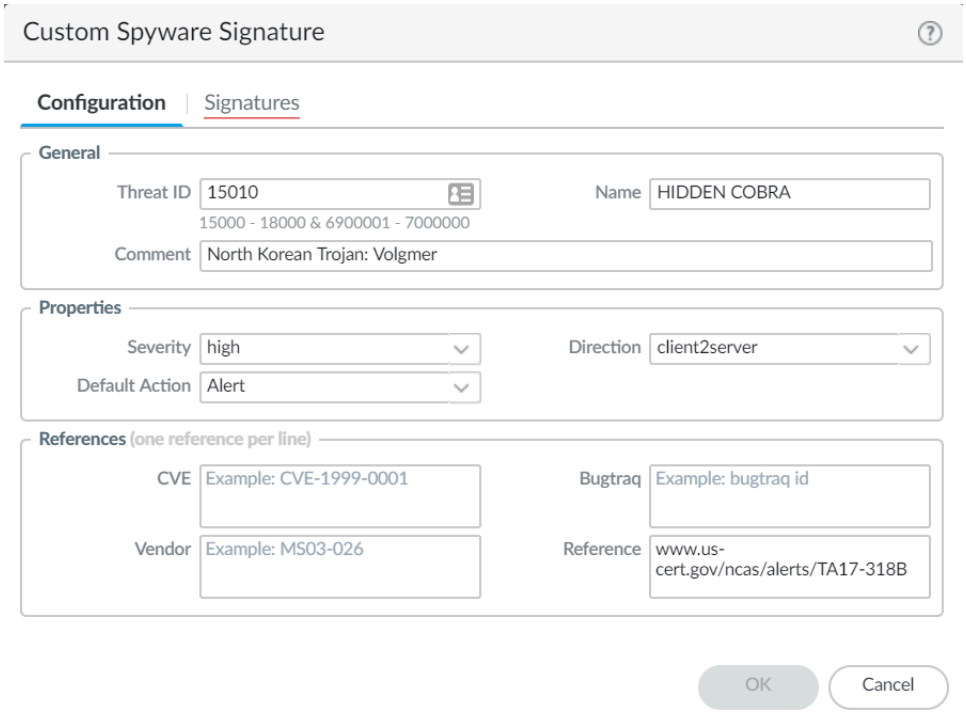
- Use the IP addresses provided as part of the IOC List to detect if a possible infection already exists by searching the firewall logs.
- The IP addresses provided can be part of an EDL or Address group and added to a Policy to block traffic to and from the suspicious list.
- Use the provided Snort signature and convert it to a custom spyware signature. This signature will become part of the spyware profile added to the appropriate policy.

For other use cases, see our companion article.

**STEP 1 |** Create a Custom Spyware Object.

1. Navigate to **Objects** > **Custom Objects** > **Spyware/Vulnerability**.
2. Click **Add** and provide a **Threat ID**, an optional comment, and fill out the Properties section.




3. Under **Signatures**, press **Add**.
4. Specify the following information:
   - **Standard**—Enter a name to identify the signature in the field.
   - **Comment**—Enter an optional description.
   - If the order in which the firewall attempts to match the signature definitions is important, keep **Ordered Condition Match** selected.
   - **Scope**—Indicate whether this signature applies to a full **Session** or a single **Transaction**.
5. Add a condition by clicking **Add And Condition** or **Add Or Condition**.
6. Select an **Operator** from the drop-down menu to define the conditions that must be true for the signature to match traffic.
   - If you select **Pattern Match**, identify a **Context** in the Snort pattern that matches our available contexts, provide a regular expression **Pattern**, and optionally, **Add** a qualifier/value pair. Select **Negate** to specify conditions under which the custom signature does not trigger.
   - If you select **Equal To**, **Less Than**, or **Greater Than**, select a **Context** and enter a **Value**.
7. Click **OK** to finish creating the Spyware object.

**STEP 2 |** Verify that the custom Spyware object is part of your Anti-Spyware Profile.

1. Go to **Security Profiles** > **Anti-Spyware**. Click an existing profile, then under **Exceptions**, search for your signature's Threat ID and **Enable** it.

**STEP 3 |** Create an EDL object.

1. Navigate to **Objects** > **External Dynamic Lists**. Click **Add**.
2. Add the suspicious IP address provided from the IOC list to a previously created EDL or a new EDL as shown below.

**STEP 4 |** Add the EDL and Anti-Spyware profiles to appropriate Policy Objects.

**STEP 5 |** Test policy is working as expected by looking at Threat logs.

**STEP 6 |** Change the action for the spyware object from alert to drop/reset after verification. Also, change the severity of the object created as needed.

**STEP 7 |** **Commit** your signature(s).

**STEP 8 |** Test your custom signature.

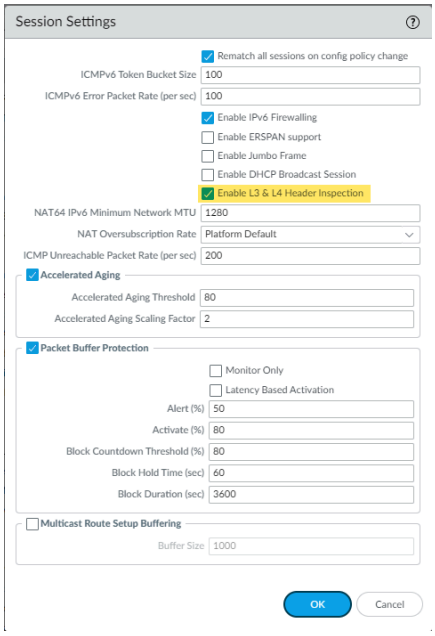
# Create a Custom L3 & L4 Vulnerability Signature

You can create custom threat signatures (vulnerability) based on Layer3 and Layer4 header fields (such as IP flags, acknowledgment numbers, etc). This enables you to provide user-created vulnerability signature coverage for old and deprecated TCP/IP stacks used in embedded / IoT devices that normally would not have any existing threat signature coverage.

Custom L3 & L4 vulnerability signatures are expressed through your Zone and Zone Protection profile configuration. You must specify how the firewall responds when it detects a threat.

**STEP 1 |** Log in to the PAN-OS web interface.

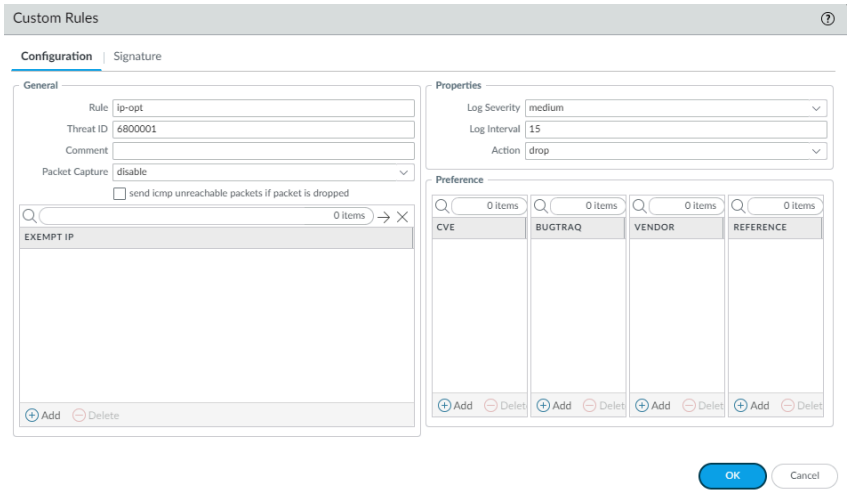
**STEP 2 |** Select **Device** > **Setup** > **Session** and enable **L3 & L4 Header Inspection** globally on the firewall.



**STEP 3 |** Create a Zone Protection profile and configure your L3 & L4 header inspection settings.

1. Select **Network** > **Network Profiles** > **Zone Protection** and either select an existing profile or **Add** a new profile.
2. If you are creating a new zone protection profile, enter a **Name** for the profile and an optional **Description**.
3. Select **L3 & L4 Header Inspection** to define your custom vulnerability signatures.
4. **Add** new custom rules by defining the configuration and signature details for each entry, which are performed in their respective tabs: **Configuration** and **Signature**.
5. Under **Configuration**, fill out the following required fields in the General, Properties, and Reference section.



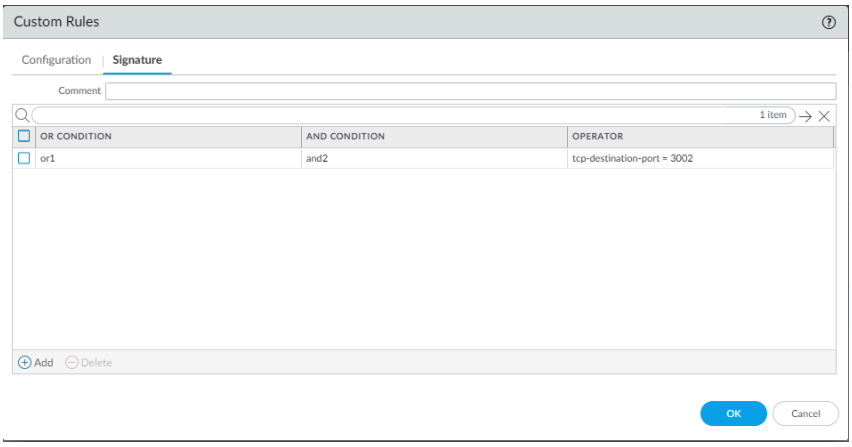

- **Rule**—Specify the custom rule name.
- **Threat ID**—Enter a numeric ID between 41000 and 45000 or 6800001 and 6900000.
- **Comment**—Optionally, add a description of the custom rule.
- **Packet Capture**—Select a packet capture setting.

  *Optionally, select **send icmp unreachable packets if packet is dropped** to send an ICMP unreachable response to the client upon packet loss.*

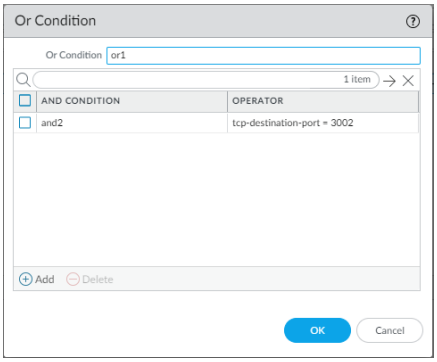
- **Exempt IP**—Enter the IP address(es) for which you do not want the custom rule to apply to.
- **Log Severity**—Select the severity of the threat.
- **Log Interval**—Indicates how frequently an event is logged.
- **Action**—Choose the action to take when there is a custom signatures match. Options include alert, drop, reset-client, reset-server, and reset-both. Refer to Security Policy Actions for more information about these action settings.
- **Reference**—Add references to provide context or related information about the custom threat signature. You can add CVEs, Bugtraq citations, 3rd party vendor IDs, or reference links to additional analysis or background information.

6. From the **Signature** tab, provide a name or description of the custom vulnerability under **Comment**. After specifying a name, select **Add** to provide the custom signature details.

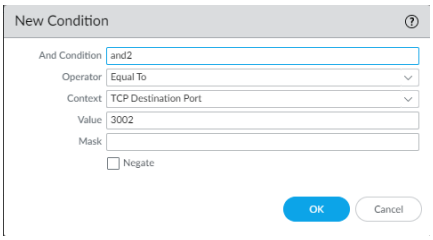
 




- Specify a matching Or Condition. When finished, select **Add** to configure an And Condition and the associated values in a new window.

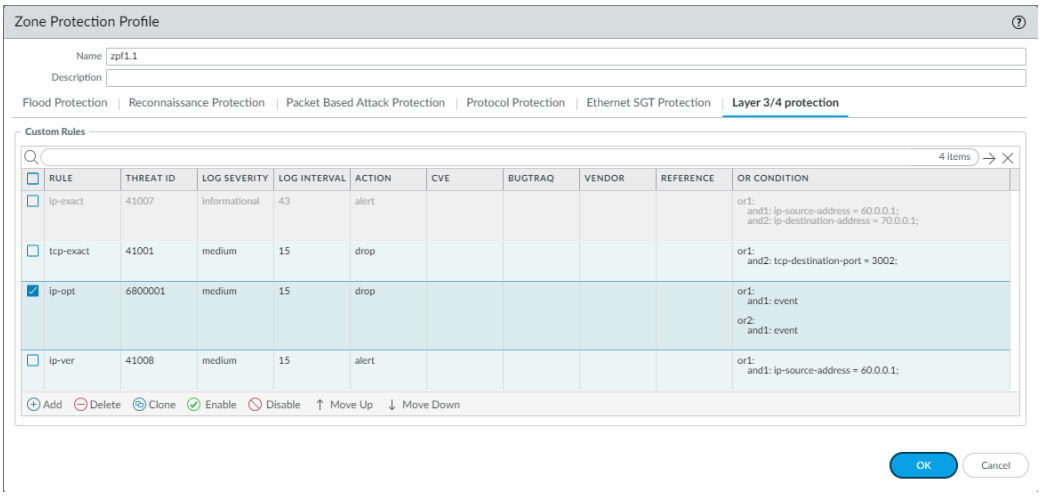



- If you select a **Less Than** or **Greater Than** operator, specify a **Context** and a **Value**. The **Equal To** operator additionally has **Mask** and **Negate** options. Click OK when you have finished configuring the new and condition.



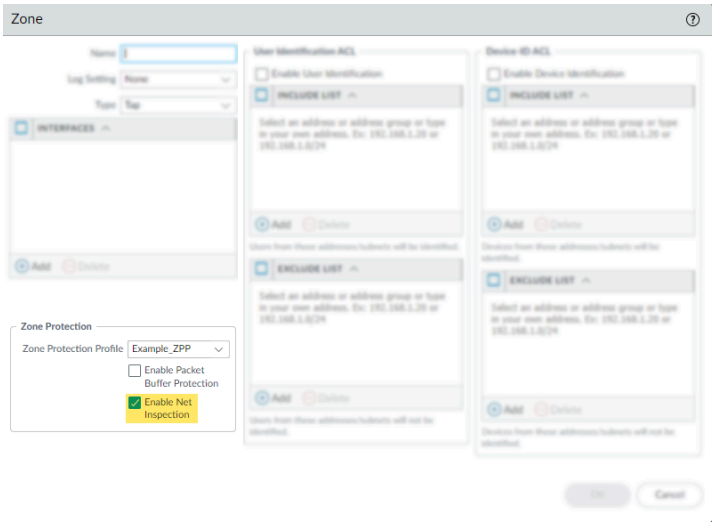

7. Repeat for each matching condition that you want to add.
8. Click **OK** and review your signatures. Click **OK** again to return to the zone protection profile.
9. From the **L3 & L4 Header Inspection** tab, you can reorder, disable, and clone the custom rule entries as necessary. Click **OK** to exit the zone protection profile.




**STEP 4 |** Apply the Zone Protection profile to a security zone that is assigned to interfaces you want to protect.

1. Select **Network** > **Zones** and select the zone where you want to assign the Zone Protection profile.
2. **Add** the **Interfaces** belonging to the zone.
3. For **Zone Protection Profile**, select the profile you just created.
4. Select **Enable Net Inspection** to enable the L3 & L4 header inspection configuration settings.




5. Click **OK**.

**STEP 5 |** **Commit** your changes.

**STEP 6 |** Test your custom signature.

# Test a Custom Signature

Custom signatures are particularly at risk for false positives and false negatives—the incorrect identification of traffic or failed detection of applications or threats. You should always test a custom signature after committing its configuration to verify that it functions as expected. Poorly written or outdated custom signatures may only be detected (and improved) through testing. If left unexamined, your signatures can reduce the efficacy of the firewall.

For custom App-ID signatures, generate traffic matching the application or application functions on a client system with a firewall between it and the application. Then, check the Traffic logs to verify that the generated sessions match the signatures you wrote. Your signature is incomplete if any traffic from your session does not match. Look at streams of sessions that do not match your signature with a packet capture tool like Wireshark. Identify unique patterns from those streams and add them to your signature to improve the accuracy of your signature.

For custom threat signatures, run penetration tests to detect system vulnerabilities. Then, view the Threat logs to see threat activity and the actions taken. Investigate any false positives or negatives. You may need to modify your signature, change its default action, or examine security profiles and policies.

Validate that traffic matches your signature as expected.

1. Run application traffic/penetration testing.
2. Navigate to **Monitor** > **Logs** > **Traffic/Threat**. Verify that you see traffic matching the custom application/threat (and that it is being handled per your policy rule).

   For example, if you wrote an application signature for uploading on example.com, you would visit example.com and upload a file. In the Traffic logs, you would verify that the session updated from “web-browsing” to “uploading-example” after the file upload.
3. Fine-tune your signature by adding additional patterns or conditions to the signature, if necessary.
4. Repeat.

# Custom Signature Pattern Requirements

The pattern requirements and available syntax for custom signatures depends on your firewall version. Firewalls running PAN-OS 10.0 (or a later version) have more flexible pattern requirements and a wider selection of regular expression (regex) syntax.

Refer to Syntax for Regular Expression Data Patterns for more details about the differences in syntax and pattern requirements between pre-PAN-OS 10.0 releases and PAN-OS 10.0 (and later) releases. You can switch between documentation releases by using the version switcher located in the left navigation bar.

If you encounter any errors using your custom signatures, verify that they conform to the following requirements.

| | | |
|---|---|---|
| **Custom Signature Pattern Requirements** | All versions | • You can enter hex-based patterns by surrounding the bytes with '**\x**'.<br>• Most signature patterns can contain a maximum of 127 characters.<br>  • If you need to use a pattern longer than 127 characters, create two separate conditions—one beginning where the other left off—and join them with '**AND**'. You can still use **Ordered Condition Match**to require the firewall to consider one condition before the other to ensure a closer match to the full string.<br>  • PA-220 and PA-800 appliances running PAN-OS 10.2 and later support a maximum pattern length of 64 characters for the following contexts: tcp-context-free and udp-context-free.<br>  *Signature compilation processes can cause other signatures to support a maximum pattern length of 64 characters, however, this is a rare occurrence.*<br>  As noted above, you can also create a pattern that is longer that the maximum size of 64 |

| | | |
|---|---|---|
| | | characters by creating two separate conditions.<br>• Some application decoders may be case-sensitive for a given field, depending on the decoder the firewall uses. For this reason, you should define variations of the pattern. For example, \.CNN\.com and \.cnn\.com will ensure your signature functions properly regardless of case. |
| | PAN-OS 9.1 and earlier versions | • Every pattern you create must contain at least one 7-byte string with fixed values.<br>  • The 7 bytes cannot include a period (`.`), an asterisk (*), a plus sign (+), or `[a-z]` (ranges).<br>  • The 7-byte string can be anywhere in your pattern.<br>• The curly braces (repetition operator) has some limitations.<br>  • Curly braces must be preceded by a '`.`' (period).<br>  • You must have 7 static bytes after the braces.<br>• If you have two strings that are both less than 7 bytes and that are separated by a regular expression wildcard element, you must increase the size of at least one of the strings to 7 or more bytes. |

# Testing Pattern Performance Impact

Firewalls running PAN-OS 10.0 or later have an enhanced pattern-matching engine that loosens pattern requirements and offers a richer selection of syntax. Used incorrectly, these features can have consequences that range from higher latency to dropped packets. To help you avoid performance degradation, the firewall enables you to check the performance impact of your signatures before you commit them.

The firewall scores the performance impact of a signature on a scale of 0 to 100%. A score of 0% means the signature severely affects firewall performance and a score of 100% means it minimally affects performance.

Use either of the following two commands to check the performance impact of a signature:

| Command | Description |
|---|---|
| test custom-signature-type pattern <pattern> | Calculates the performance impact of a signature without a context and determines whether the pattern is not valid, is valid but in only the new engine (lscan), or is valid in both the old and new engine (pscan/AHO).<br>Example:<br>**admin@VM-FW-75-252> test custom-signature-type pattern aaaa.**<br>*The pattern is lscan pattern<br>Performance score: 68% |
| test custom-signature-perf context <context> pattern <pattern> | Calculates the performance impact of a signature with a context and displays a warning if the performance score is below 55%.<br>Example:<br>**admin@VM-FW-75-252> test custom-signature-perf context http-rsp-headers pattern aaaa.***<br>Performance score: 42%<br>This signature will have performance impact |

When you test a custom signature without a context, the score is a function of the literal parts of the pattern. The literal parts are the characters in the string with fixed values, such as “pan” and “net” in **pan.{4}net**. The greater the number and length of the literal parts, the higher the score of the pattern.

When you test a pattern with a context, the firewall performs the above calculation and adjusts it based on the typical length and frequency of the context. The firewall then divides the typical context length by the shortest literal part of the pattern and multiplies the base score of the pattern by this value. Finally, the firewall lowers the score if the context appears frequently and raises the score if the context appears infrequently.

# Custom Signature Contexts

Custom signature contexts are available for both string and integer context types.

- String Contexts
- Integer Contexts
- Context Qualifiers

## String Contexts

String Contexts are a type of custom signature context. They are used for Pattern Match operators.

- dhcp-req-chaddr
- dhcp-req-ciaddr
- dhcp-rsp-chaddr
- dhcp-rsp-ciaddr
- dns-req-addition-section
- dns-req-answer-section
- dns-req-authority-section
- dns-req-header
- dns-req-protocol-payload
- dns-req-section
- dns-rsp-addition-section
- dns-rsp-answer-section
- dns-rsp-authority-section
- dns-rsp-header
- dns-rsp-protocol-payload
- dns-rsp-ptr-answer-data
- dns-rsp-queries-section
- email-headers
- file-data
- file-elf-body
- file-flv-body
- file-html-body
- file-java-body
- file-mov-body
- file-office-content
- file-pdf-body

- file-riff-body
- file-swf-body
- file-tiff-body
- file-unknown-body
- ftp-req-params
- ftp-req-protocol-payload
- ftp-rsp-protocol-payload
- ftp-rsp-banner
- ftp-rsp-message
- gdbremote-req-context
- gdbremote-rsp-context
- giop-req-message-body
- giop-rsp-message-body
- h225-payload
- http-req-cookie
- http-req-headers
- http-req-host-header
- http-req-host-ipv4-address-found
- http-req-host-ipv6-address-found
- http-req-message-body
- http-req-mime-form-data
- http-req-ms-subdomain
- http-req-origin-headers
- http-req-params
- http-req-uri
- http-req-uri-path
- http-req-user-agent-header
- http-rsp-headers
- http-rsp-non-2xx-response-body
- http-rsp-reason
- icmp-req-code
- icmp-req-data
- icmp-req-type
- icmp-req-protocol-payload
- icmp-rsp-data
- icmp-rsp-protocol-payload

- icmp-req-possible-custom-payload
- ike-req-headers
- ike-rsp-headers
- ike-req-payload-text
- ike-rsp-payload-text
- imap-req-cmd-line
- imap-req-first-param
- imap-req-params-after-first-param
- imap-req-protocol-payload
- imap-rsp-protocol-payload
- irc-req-params
- irc-req-prefix
- jpeg-file-scan-data
- jpeg-file-segment-data
- jpeg-file-segment-header
- ldap-req-searchrequest-baseobject
- ldap-rsp-searchresentry-objectname
- ms-ds-smb-req-share-name
- ms-ds-smb-req-v1-create-filename
- ms-ds-smb-req-v2-create-filename
- msrpc-req-bind-data
- mssql-db-req-body
- netbios-dg-req-protocol-payload
- netbios-dg-rsp-protocol-payload
- netbios-ns-req-protocol-payload
- netbios-ns-rsp-protocol-payload
- nettcp-req-context
- oracle-req-data-text
- pe-dos-headers
- pe-file-header
- pe-optional-header
- pe-section-header
- pe-body-data
- pop3-req-protocol-payload
- pop3-rsp-protocol-payload
- pre-app-req-data

- pre-app-rsp-data
- rtmp-req-message-body
- rtsp-req-headers
- rtsp-req-uri-path
- sip-req-headers
- snmp-req-community-text
- smtp-req-argument
- smtp-req-protocol-payload
- smtp-rsp-protocol-payload
- smtp-rsp-content
- ssh-req-banner
- ssh-rsp-banner
- ssl-req-certificate
- ssl-req-chello-sni
- ssl-req-client-hello
- ssl-req-protocol-payload
- ssl-req-random-bytes
- ssl-rsp-cert-subjectpublickey
- ssl-rsp-certificate
- ssl-rsp-protocol-payload
- ssl-rsp-server-hello
- tcp-context-free
- telnet-req-client-data
- telnet-rsp-server-data
- udp-context-free
- unknown-req-tcp-payload
- unknown-rsp-tcp-payload
- unknown-req-udp-payload
- unknown-rsp-udp-payload

## dhcp-req-chaddr

Identifies the DHCP request client hardware address.

**Additional Details**

None

**Context Capture**

This context provides the highlighted text.

```
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xfb1fc570
  Seconds elapsed: 0
▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: Vmware_00:0C:29 (01:23:45:67:89:AB)
  Client hardware address padding: 00 00 00 00 00 00 00 00 00 00
```

## dhcp-req-ciaddr

Identifies the DHCP request client IP address.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
  Hops: 0
  Transaction ID: 0xfb1fc570
  Seconds elapsed: 0
▶ Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0
  Your (client) IP address: 0.0.0.0
  Next server IP address: 0.0.0.0
  Relay agent IP address: 0.0.0.0
  Client MAC address: Vmware_00:0C:29 (01:23:45:67:89:AB)
  Client hardware address padding: 00 00 00 00 00 00 00 00 00 00
  Server host name not given
  Boot file name not given
```

## dhcp-rsp-chaddr

Identifies the DHCP response client hardware address.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
    Hops: 0
    Transaction ID: 0xfb1fc570
    Seconds elapsed: 0
  ▶ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 10.0.0.61
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: Vmware_00:0C:29 (01:23:45:67:89:AB)
    Client hardware address padding: 00 00 00 00 00 00 00 00 00 00
    Server host name not given
    Boot file name not given
```

## dhcp-rsp-ciaddr

Identifies the DHCP response client IP address.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
▼ Dynamic Host Configuration Protocol (ACK)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0xfb1fc570
    Seconds elapsed: 0
  ▶ Bootp flags: 0x0000 (Unicast)
    Client IP address: 0.0.0.0
    Your (client) IP address: 10.0.0.61
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
```

## dns-req-addition-section

Additional records section if found in a DNS request (normal DNS requests should not have an additional records section).

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    [Request In: 3]
    [Time: 0.832133000 seconds]
    Transaction ID: 0xf76f
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 6
    Authority RRs: 0
    Additional RRs: 6
  ▽ Queries
    ▷ google.com: type MX, class IN
  ▷ Answers
  ▽ Additional records
    ▽ smtp4.google.com: type A, class IN, addr 216.239.37.26
        Name: smtp4.google.com
        Type: A (Host address)
        Class: IN (0x0001)
        Time to live: 10 minutes
        Data length: 4
        Addr: 216.239.37.26 (216.239.37.26)
```

## dns-req-answer-section

Answer section if found in a DNS request (normal DNS requests should not have an answer section).

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    Transaction ID: 0x2720
  ▷ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ▷ Queries
  ▽ Answers
    ▽ agentdns.com: type SOA, class IN, mname agentdns.com
        Name: agentdns.com
        Type: SOA (Start of zone of authority)
        Class: IN (0x0001)
        Time to live: 1 hour
        Data length: 51
        Primary name server: agentdns.com
        Responsible authority's mailbox: siteops.topproducersystems.com
        Serial number: 28
        Refresh interval: 15 minutes
        Retry interval: 10 minutes
        Expiration limit: 1 day
        Minimum TTL: 1 hour
```

## dns-req-authority-section

Authority section if found in a DNS request (normal DNS requests should not have an authority section).

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    Transaction ID: 0x19e4
  ▷ Flags: 0x8583 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▷ Queries
  ▽ Authoritative nameservers
    ▽ 5.10.in-addr.arpa: type SOA, class IN, mname ns0.eng.paloaltonetworks.local
        Name: 5.10.in-addr.arpa
        Type: SOA (Start of zone of authority)
        Class: IN (0x0001)
        Time to live: 1 day
        Data length: 83
        Primary name server: ns0.eng.paloaltonetworks.local
        Responsible authority's mailbox: root.ns0.eng.paloaltonetorks.local
        Serial number: 2012112700
        Refresh interval: 2 hours
        Retry interval: 10 minutes
        Expiration limit: 14 days
        Minimum TTL: 5 minutes
```

## dns-req-header

Full DNS request header (12 bytes) with the transaction ID, query flags, number of questions, and the Resource Record (RR) values in a DNS request.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (query)
    [Response In: 112]
    Transaction ID: 0xf9a6
  ▽ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
```

## dns-req-protocol-payload

The payload of a DNS request.

### Context Capture

```
▼ Domain Name System (query)
    Transaction ID: 0x0004
  ▶ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▶ Queries
    [Response In: 2]
```

## dns-req-section

This context matches the DNS questions of a DNS query so that patterns can be written against one or more domains in a given DNS query.

### Additional Details

This context is a direct pattern match against the format of a DNS query, so patterns must adhere to the DNS question structure. A recommended approach to create a DNS pattern is to capture the DNS request with Wireshark and copy the DNS Request field (make sure to remove the ending period in the request).

### Context Capture

This example illustrates how to build a signature for a DNS query for the domain www.thebayareagamers.com.

The signature pattern is:
\x 03 77 77 77 10 74 68 65 62 61 79 61 72 65 61 67 61 6d 65 72 73 03 63 6f 6d\x

| Pattern | Description |
|---|---|
| \x | Indicates this pattern is a hex pattern match |
| 03 | Indicates that the next 3 bytes are to be matched |
| 77 77 77 | "www" |
| | [The period in the domain name is omitted.] |
| 10 | Indicates that the next 16 bytes (10 hex) are to be matched |
| 74 68 65 62 61 79 61 72 65 61 67 61 6d 65 72 73 | "thebayareagamers" |
| 03 | Indicates that the next 3 bytes are to be matched |
| 63 6f 6d | "com" |
| \x | Ends hex pattern match |

The Wireshark representation of the above table. Everything highlighted yellow and blue is provided by this context. The blue section is where the hexadecimal string is pulled from for the above table.

```
▽ Domain Name System (query)
    [Response In: 16]
    Transaction ID: 0x8521
  ▷ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ www.thebayareagamers.com: type A, class IN
        Name: www.thebayareagamers.com
        Type: A (Host address)
        Class: IN (0x0001)

0000  b4 0c 25 ec d1 10 b8 f6  b1 1c 72 4d 08 00 45 00   ..%..... ..rM..E.
0010  00 46 a8 10 00 00 ff 11  40 f7 c0 a8 01 e7 08 08   .F...... @.......
0020  08 08 d2 e5 00 35 00 32  26 01 85 21 01 00 00 01   .....5.2 &..!....
0030  00 00 00 00 00 00 03 77  77 77 10 74 68 65 62 61   .......w ww.theba
0040  79 61 72 65 61 67 61 6d  65 72 73 03 63 6f 6d 00   yareagam ers.com.
0050  00 01 00 01                                         ....
```

## dns-rsp-addition-section

Additional records section of a DNS response.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (query)
    [Response In: 2]
    Transaction ID: 0x2720
  ▷ Flags: 0x0000 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 1
  ▷ Queries
  ▽ Additional records
    ▽ <Root>: type OPT
        Name: <Root>
        Type: OPT (EDNS0 option)
        UDP payload size: 4096
        Higher bits in extended RCODE: 0x0
        EDNS0 version: 0
      ▽ Z: 0x8000
          Bit 0 (DO bit): 1 (Accepts DNSSEC security RRs)
          Bits 1-15: 0x0 (reserved)
        Data length: 0
```

## dns-rsp-answer-section

All of the DNS Answers section with the exception of PTR records. PTR records are matched in a separate context.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    Transaction ID: 0x2720
  ▷ Flags: 0x8400 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 1
  ▷ Queries
  ▽ Answers
    ▽ agentdns.com: type SOA, class IN, mname agentdns.com
        Name: agentdns.com
        Type: SOA (Start of zone of authority)
        Class: IN (0x0001)
        Time to live: 1 hour
        Data length: 51
        Primary name server: agentdns.com
        Responsible authority's mailbox: siteops.topproducersystems.com
        Serial number: 28
        Refresh interval: 15 minutes
        Retry interval: 10 minutes
        Expiration limit: 1 day
        Minimum TTL: 1 hour
```

## dns-rsp-authority-section

The complete authority section of a DNS response.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    [Request In: 826]
    [Time: 0.051163000 seconds]
    Transaction ID: 0x44c3
  ▷ Flags: 0x8183 Standard query response, No such name
    Questions: 1
    Answer RRs: 0
    Authority RRs: 1
    Additional RRs: 0
  ▷ Queries
  ▽ Authoritative nameservers
    ▽ <Root>: type SOA, class IN, mname a.root-servers.net
        Name: <Root>
        Type: SOA (Start of zone of authority)
        Class: IN (0x0001)
        Time to live: 8 minutes, 19 seconds
        Data length: 64
        Primary name server: a.root-servers.net
        Responsible authority's mailbox: nstld.verisign-grs.com
        Serial number: 2013072301
        Refresh interval: 30 minutes
        Retry interval: 15 minutes
        Expiration limit: 7 days
        Minimum TTL: 1 day
```

## dns-rsp-header

Full DNS response header, which includes the transaction ID, query flags, the number of questions, and the Resource Record (RR) values.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    [Request In: 273]
    [Time: 0.015170000 seconds]
    Transaction ID: 0xd20a
  ▽ Flags: 0x8180 Standard query response, No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
  ▷ Answers
```

## dns-rsp-protocol-payload

*This is the description*

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▼ Domain Name System (response)
    Transaction ID: 0x0004
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 4
    Additional RRs: 4
  ▶ Queries
  ▶ Answers
  ▶ Authoritative nameservers
  ▶ Additional records
    [Request In: 1]
    [Time: 0.021210000 seconds]
```

## dns-rsp-ptr-answer-data

FQDN for a type PTR DNS response.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    [Request In: 69]
    [Time: 0.088866000 seconds]
    Transaction ID: 0x05dd
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▷ Queries
  ▽ Answers
    ▽ 177.64.98.207.in-addr.arpa: type PTR, class IN, corv-carl-gw.nero.net
        Name: 177.64.98.207.in-addr.arpa
        Type: PTR (Domain name pointer)
        Class: IN (0x0001)
        Time to live: 6 hours
        Data length: 23
        Domain name: corv-carl-gw.nero.net
```

## dns-rsp-queries-section

Name, type, and class of the queries section in a DNS response.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Domain Name System (response)
    [Request In: 109]
    [Time: 0.006633000 seconds]
    Transaction ID: 0xf9a6
  ▷ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ exploit-db.com: type A, class IN
        Name: exploit-db.com
        Type: A (Host address)
        Class: IN (0x0001)
```

## email-headers

All email headers and the plain text email body. Attachments are not included in this context as they are provided elsewhere.

### Additional Details

None

### Context Capture

This context provides the text in bold.


```
Microsoft Mail Internet Headers Version 2.0
Received: from mail.litwareinc.com ([10.54.108.101]) by mail.proseware.com with Microsoft
SMTPSVC(6.0.3790.0);
Wed, 12 Dec 2007 13:39:22 -0800
Received: from mail ([10.54.108.23] RDNS failed) by mail.litware.com with Microsoft SMTPSVC(6.0.3790.0);
Wed, 12 Dec 2007 13:38:49 -0800
From: "Kelly J. Weadock" <kelly@litware.com>
To: <anton@proseware.com>
Cc: <tim@cpandl.com>
Subject: Review of staff assignments
Date: Wed, 12 Dec 2007 13:38:31 -0800
Message-ID: <MAILbbnewS5TqCRL00000013@mail.litware.com>
X-OriginalArrivalTime: 12 Dec 2007 21:38:50.0145 (UTC)

Hey,

Check out this picture.

Kelly

Content-Type: image/gif; name="world1.gif"
Content-Description: world1.gif
Content-Disposition: attachment; filename="world1.gif"; size=292;
        creation-date="Wed, 12 DEC 2007 07:29:14 GMT";
        modification-date=" Wed, 12 DEC 2007 07:29:14 GMT"
Content-ID: <LKAJDF9282LKSDKA@litware.com>
Content-Transfer-Encoding: base64

R0lGODlhFAAWAKEAAP///8z//wCZMwAAACH+TlRoaXMgYXJ0IGlzIGluIHRoZSBwdWJsaWMgZG9t
YWluLiBLZXZpbiBldWdoZXMsIGtldmluaEBlaXQuY29tLCBTZXB0ZW1iZXIgMTk5NQAh+QQBAAAB
ACwAAAAAFAAWAAACY4yPqTrtm5qYtMEGBNiaWzRMHEVIwgBm5lieR7hqsiqjQSjG3I7C9LgznXw5
nUwjAaqEliSs2Vl2nKWgIlfbsHJTV3bJJNkGLG10arspwZ20mlYVum++8PBCBn8gBseDD7hQAAA7
```


## file-data

Covers the data of transferred files.

### Additional Details

This context supports the following file types:

- 7z
- ABR
- ACE
- ANI
- ARJ header
- ASF
- BAT
- BMP
- CAB
- CAFF
- CHM
- Cineon
- CorelDRAW
- CRX
- CSV
- DER
- DEX
- DMG
- DOC
- DWF
- DWG
- EICAR
- ELF
- Email headers
- EMF
- EPS
- FFmpeg
- FLAC
- FLV
- Font
- GDS
- GIF
- GZip
- HDF
- HLP
- ICO

- IFF
- IVR
- JarPack
- Java
- JPEG
- JS
- PL
- HTA
- LNK
- LZH
- M3U
- Mach
- MAKI
- MDB
- MDI
- MFT
- MIDI
- MOV
- MP2T
- MP3
- MPEG
- MVG
- MSOFFICE
- OGG
- OOXML
- Pcap
- PDF
- PE
- PGP
- PICT
- PKG
- PLS
- PNG
- Powershell
- PSD
- QVF

- RA
- RAR
- RIFF
- RLA
- RTF
- RWS
- SAMI
- SGI
- SH
- Shockwavelte #
- Shockwavebte #
- Softimage# PIC
- Soundbank
- SVG
- SWC
- SWF
- SWZ
- TIFF
- TNEF
- VBS
- WebM
- WebP
- WMF
- WOFF
- WPC
- WRI
- ZIP

**Context Capture**

This context captures the following information for a given file type (here GIF87a).

```
▼ Compuserve GIF, Version: GIF87a
    Version: GIF87a
    Screen width: 16
    Screen height: 16
  ▶ Global settings: (Global color table present) (1 bit per color) (1 bit per pixel)
    Background color index: 0
    Global color map: ffffffffffff
  ▶ Image
    Trailer (End of the GIF stream)
```

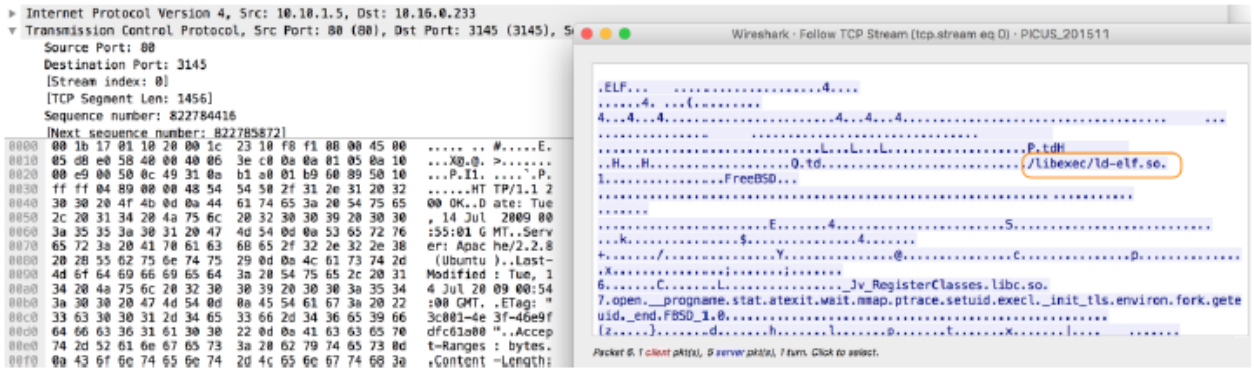
## file-elf-body

Identifies an executable and linkable formatted (ELF) file type contained in a protocol or application response and checks the ELF file body.

### Additional Details

None

### Context Capture




## file-flv-body

Full body of a flash video file minus the first 9 bytes, which are reserved for the header.

### Additional Details

Here is a screenshot from Wikipedia detailing the 9-byte header:

| Field | Data Type | Example | Description |
|---|---|---|---|
| Signature | byte[3] | "FLV" | Always "FLV" |
| Version | uint8 | 0x01 | Currently 0x01 for known FLV files |
| Flags | uint8 bitmask | 0x05 | Bitmask: 0x04 is audio, 0x01 is video (so 0x05 is audio+video) |
| Header Size | uint32_be | 0x00 0x00 0x00 0x09 | Always 9 for known FLV files |

### Context Capture

Using a cli hex-editor named xxd, we can view the header of the flash file.

```
Macbook:~ noob$ xxd -l 9 flash_video.flv
0000000: 464c 5601 0500 0000 09          FLV......
```

Every byte after the 9th is provided by this context. Only the first 50 bytes were printed here as an example.

```
Macbook:~ noob$ xxd -l 50 flash_video.flv
0000000: 464c 5601 0500 0000 0900 0000 0012 0003  FLV.............
0000010: 4b00 0000 0000 0000 0200 0a6f 6e4d 6574  K..........onMet
0000020: 6144 6174 6108 0000 000f 0008 6475 7261  aData.......dura
0000030: 7469
```

## file-html-body

Full body of a HTML file, minus the first 8 bytes as they're reserved for the header.

### Additional Details

None

### Context Capture

xxd is a cli-based hex editor; every byte after the 8th byte is provided by this context. Only the first 50 bytes were printed here as an example.

```
Macbook:~ noob$ xxd -l 50 The_legend_of_random.html
0000000: 3c21 444f 4354 5950 4520 6874 6d6c 2050  <!DOCTYPE html P
0000010: 5542 4c49 4320 222d 2f2f 5733 432f 2f44  UBLIC "-//W3C//D
0000020: 5444 2058 4854 4d4c 2031 2e30 2054 7261  TD XHTML 1.0 Tra
0000030: 6e73                                     ns
```

## file-java-body

Full body of a Java file, minus the first 4 bytes, which is always 0xCAFEBABE ("cafebabe").

### Additional Details

None

### Context Capture

The first 4 bytes of the Java file printed by the cli-based hex editor, xxd. Every byte after the 4th is provided by this context. Only the first 25 bytes were printed in the above example.

```
Macbook:~ noob$ xxd -l 4 java_file.class
0000000: cafe babe                                ....
```

## file-mov-body

Full body of a MOV file, minus the first 8 bytes as they're reserved for the header.

### Additional Details

None

#### Context Capture

xxd is a cli-based hex editor; every byte after the 8th is provided by this context. Only the first 50 bytes were printed in this example.

```
Macbook:~ noob$ xxd -l 50 /System/Library/Compositions/Yosemite.mov
0000000: 0000 0020 6674 7970 7174 2020 2005 0300  ... ftypqt   ...
0000010: 7174 2020 0000 0000 0000 0000 0000 0000  qt ............
0000020: 0000 10ae 6d6f 6f76 0000 006c 6d76 6864  ....moov...lmvhd
0000030: 0000
```

## file-office-content

Full body of a Microsoft Office Document file, minus the first 8 bytes as they're reserved for the header.

#### Additional Details

None

#### Context Capture

xxd is a cli-based hex editor, every byte after the 8th is provided by this context. Only the first 50 bytes were printed in this example.

```
Macbook:~ noob$ xxd -l 50 Word_Document.doc
0000000: d0cf 11e0 a1b1 1ae1 0000 0000 0000 0000  ................
0000010: 0000 0000 0000 0000 3e00 0300 feff 0900  ........>.......
0000020: 0600 0000 0000 0000 0000 0000 2000 0000  ............ ...
0000030: b20f
```

## file-pdf-body

The full body of a PDF file minus the first 8 bytes, which are reserved for the header.

#### Additional Details

Compressed data is provided as decompressed data by the decoder.

#### Context Capture

xxd is a cli-based hex editor, every byte after the 8th is provided by this context. Only the first 50 bytes were printed in this example.

```
Macbook:~ noob$ xxd -l 50 WildFire_Administrators_Guide-5.1.pdf
0000000: 2550 4446 2d31 2e36 0d25 e2e3 cfd3 0d0a  %PDF-1.6.%......
0000010: 3431 3332 2030 206f 626a 0d3c 3c2f 4c69  4132 0 obj.<</Li
0000020: 6e65 6172 697a 6564 2031 2f4c 2031 3237  nearized 1/L 127
0000030: 3834                                     84
```

## file-riff-body

Full body of a RIFF file, minus the first 8 bytes as they're reserved for the header.

#### Additional Details

None

#### Context Capture

xxd is a cli-based hex editor; every byte after the 8th is provided by this context. Only the first 50 bytes were printed in this example.

```
Macbook:~ noob$ xxd -l 50 /pentest/misc/exiftool/t/images/RIFF.avi
0000000: 5249 4646 b63b 2a00 4156 4920 4c49 5354  RIFF.;*.AVI LIST
0000010: 4601 0000 6864 726c 6176 6968 3800 0000  F...hdrlavih8...
0000020: 6a04 0100 c824 0300 0000 0000 1000 0100  j....$..........
0000030: e900                                     ..
```

## file-swf-body

Full body of a SWF file, minus the first 8 bytes as they're reserved for the header.

#### Additional Details

None

#### Context Capture

xxd is a cli-based hex editor; every byte after the 8th is provided by this context. Only the first 50 bytes were printed in this example.

```
Macbook:~ noob$ xxd -l 50 Cinema.swf
0000000: 4357 530a bef9 3c00 78da c4bd 0778 1bc7  CWS...<.x....x..
0000010: d52e 8c99 c562 b128 043b 2952 9229 773b  .....b.(.;)R.)w;
0000020: b624 cb89 132b 8e1d 8aa4 2426 5431 49c9  .$...+....$&T1I.
0000030: f697                                     ..
```

## file-tiff-body

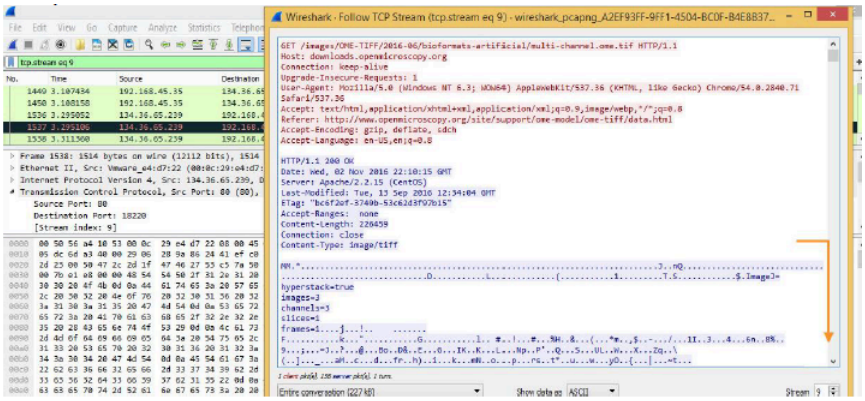
When the firewall detects a tagged image file format (TIFF) file, this context returns data contained within the body of the file.

#### Additional Details

None

### Context Capture



## file-unknown-body

This context provides data after the first 8 bytes and up to 7 packets of an unknown file we couldn't otherwise identify.

### Additional Details

None

### Context Capture

xxd is a cli-based hex editor; every byte after the 8th is provided up until 7 bytes is seen. In this example the first 8 bytes are numbered to easily show what wouldn't be matched. Next are "A's" followed by "shellcode" in hex. You could block this file by adding '\x7368656c6c636f6465\x' in the "Pattern" field of the custom signature.

```
Macbook:~ noob$ xxd file.bin
0000000: 1122 3344 5566 7788 4141 4141 4141 4141  ."3DUfw.AAAAAAAA
0000010: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
0000020: 7368 656c 6c63 6f64 65                   shellcode
```

## ftp-req-params

Parameters following an FTP command.

### Additional Details

None

### Context Capture

The context provides the text highlighted in yellow. Qualifiers: This context can use FTP command and FTP vendor ID qualifiers to limit signatures to specific FTP commands and known FTP clients.


## ftp-req-protocol-payload

The payload of an FTP request.

**Context Capture**


## ftp-rsp-protocol-payload

The payload of an FTP response.

**Context Capture**

```
▼ File Transfer Protocol (FTP)
  ▼ 331 Anonymous login ok, send your complete email address as your password.\r\n
      Response code: User name okay, need password (331)
      Response arg: Anonymous login ok, send your complete email address as your password.
  [Current working directory: ]
```

## ftp-rsp-banner

FTP welcome banner shown before authentication.

**Additional Details**

None

**Context Capture**

```
▽ File Transfer Protocol (FTP)
  ▽ 220 Welcome to FTP service!\r\n
      Response code: Service ready for new user (220)
      Response arg: Welcome to FTP service!
```

## ftp-rsp-message

FTP server response code and the code itself. Note, that the code and the space can be used as part of the required 7-byte anchor.

**Additional Details**

None

**Context Capture**

This context matches the text highlighted in yellow.

```
▷ Transmission Control Protocol, Src Port: ftp (21),
▽ File Transfer Protocol (FTP)
  ▽ 230 Login successful.\r\n
      Response code: User logged in, proceed (230)
      Response arg: Login successful.
```

## gdbremote-req-context

GDB is a process debugger that has the ability to debug across the network. This context provides the request data.

**Additional Details**

None

**Context Capture**

After capturing the GDB network data, follow the TCP stream to view the data. In this instance, everything in red is the request data, and that is what this context provides.


## gdbremote-rsp-context

GDB is a process debugger that has the ability to debug across the network. This context provides the response data.

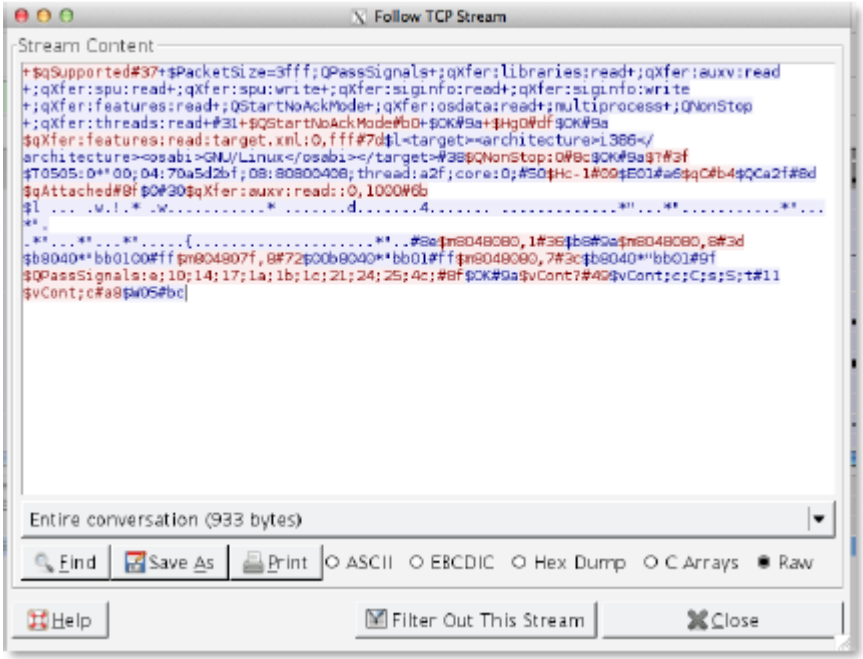
### Additional Details

None

### Context Capture

After capturing the GDB network data, I followed the TCP stream to view the data. In this instance, everything in blue is what this context provides




## giop-req-message-body

Everything in the GIOP request.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▷ General Inter-ORB Protocol
▽ General Inter-ORB Protocol Request
  ▷ ServiceContextList
    Request id: 2
    Response expected: 1
    Object Key length: 40
    Object Key: 3a5c626977316b6261743030373a49545f6461656d6f6e3a...
    Operation length: 15
    Request operation: getIIOPDetails
    Requesting Principal Length: 9
    Requesting Principal: hhamblen.
    Stub data (40 bytes)
```

## giop-rsp-message-body

Data after the GIOP header in a GIOP response.

### Additional Details

None

**Context Capture**

This context provides the text highlighted in yellow.

```
▷ General Inter-ORB Protocol
▽ General Inter-ORB Protocol Reply
  ▶ ServiceContextList
    Request id: 2
    Reply status: No Exception (0)
    Stub data (23 bytes)
```

## h225-payload

Extracts any data contained in an H.225.0 (App-ID: h.225) request.

**Additional Details**

None

**Context Capture**

```
▶ Internet Protocol Version 4, Src: 10.180.6.135, Dst: 146.220.202.132
▶ User Datagram Protocol, Src Port: 1719 (1719), Dst Port: 1719 (1719)
▼ H.225.0 RAS
  ▼ RasMessage: registrationRequest (3)
    ▼ registrationRequest
        requestSeqNum: 4185
        protocolIdentifier: 0.0.8.2250.0.5 (Version 5)
        0... .... discoveryComplete: False
        callSignalAddress: 0 items
      ▶ rasAddress: 1 item
      ▶ terminalType
        gatekeeperIdentifier: GATEKEEPER
      ▶ endpointVendor
        timeToLive: 30
        1... .... keepAlive: True
        endpointIdentifier: 000121628418870112227887250172022653634661719
        0... .... willSupplyUUIEs: False
        0... .... maintainConnection: False
        supportsAltGK: NULL
      ▶ featureSet
        0... .... supportsAssignedGK: False
    [The response to this request is in frame 37]

0000  00 1b 17 00 01 31 00 21  a1 6c 3b 53 08 00 45 00   .....1.! .l;S..E.
0010  00 f2 00 00 40 00 3d 11  ce 5f 0a b4 06 87 92 dc   ....@.=. ._......
0020  ca 84 06 b7 06 b7 00 de  16 ee 0e 40 10 58 06 00   ........ ...@.X..
0030  08 91 4a 00 05 00 00 01  00 0a b4 06 87 06 b7 22   ..J..... ......."
0040  c0 0b 0b 03 87 0b 53 43  4f 50 49 41 5f 56 43 32   ......SC OPIA_VC2
0050  34 30 07 32 2e 35 2e 31  2e 32 32 01 20 00 47 00   40.2.5.1 .22. .G.
0060  41 00 54 00 45 00 4b 00  45 00 45 00 50 00 45 00   A.T.E.K. E.E.P.E.
```

## http-req-cookie

Returns the Cookie header value contained in an HTTP request header.

#### Additional Details

None

#### Context Capture

```
▼ Hypertext Transfer Protocol
  ▶ GET /favicon.ico HTTP/1.1\r\n
    Host: 172.16.8.156\r\n
    User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:8.0) Gecko/20100101 Firefox/8.0\r\n
    Accept: image/png,image/*;q=0.8,*/*;q=0.5\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Connection: keep-alive\r\n
  ▼ Cookie: TSLSecret=CVE-2012-0053; TSLNotSoSecret=TSL20120127-03\r\n
      Cookie pair: TSLSecret=CVE-2012-0053
      Cookie pair: TSLNotSoSecret=TSL20120127-03
    \r\n
    [Full request URI: http://172.16.8.156/favicon.ico]
    [HTTP request 2/3]
    [Prev request in frame: 7]
    [Response in frame: 15]
    [Next request in frame: 17]
```

## http-req-headers

HTTP request header, not including the method, path, HTTP version, or host as those are provided elsewhere.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow. This context can use HTTP header field and HTTP method qualifiers to limit signatures to HTTP headers with specific values for select header fields and for specific HTTP methods.

```
▽ Hypertext Transfer Protocol
  ▷ POST /blog/wp-admin/admin-ajax.php HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 321\r\n
    Origin: http://thelegendofrandom.com\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko)
    Content-type: application/x-www-form-urlencoded\r\n
    Accept: */*\r\n
    Referer: http://thelegendofrandom.com/blog/\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f\r\n
```

## http-req-host-header

Host field in a HTTP request header.

#### Additional Details

The pattern match searches for a pattern only within the highlighted field. By default it does not apply line start and line end anchors; as a result, the pattern 'example.com' will match with <anytext>example.com<anytext>. To initiate an exact match search, you must add a <space> before the pattern and '\r\n' after the pattern on PAN-OS 9.1 and earlier. Starting with PAN-OS 10.0 you can use the following anchor characters: ^ and $ to specify a string start and end.

#### Context Capture

This context provides the text highlighted in yellow. This context can use HTTP header field and HTTP method qualifiers to limit signatures to HTTP headers with specific values for select header fields and for specific HTTP methods.

```
▽ Hypertext Transfer Protocol
  ▷ GET /blog/wp-content/uploads/2012/10/15.png HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko)
```

## http-req-host-ipv4-address-found

When an HTTP request host header contains an IPv4 address, the value is set to 1.

#### Additional Details

None

#### Context Capture

```
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: 10.2.156.28\r\n
    User-Agent: curl/7.58.0\r\n
    Accept: */*\r\n
    \r\n
    [Full request URI: http://10.2.156.28/]
    [HTTP request 1/1]
    [Response in frame: 8]
```

## http-req-host-ipv6-address-found

When an HTTP request host header contains an IPv6 address, the value is set to 1.

#### Additional Details

None

**Context Capture**

```
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: [2607:f8b0:4005:80a::200e]\r\n
    User-Agent: curl/7.58.0\r\n
    Accept: */*\r\n
    \r\n
    [Full request URI: http://[2607:f8b0:4005:80a::200e]/]
    [HTTP request 1/1]
    [Response in frame: 8]
```

## http-req-message-body

Body content of an HTTP request when the body content cannot be recognized as URL encoded or MIME type data using the Content-type field. For signatures concerning URLs, reference http-req-params; and for MIME type data, reference http-req-mime-form-data.

### Additional Details

None

### Context Capture

This context provides the full body. I followed the TCP stream in Wireshark and only chose a portion of the body for the signature to match. Qualifiers: This context can use the HTTP method qualifier to limit signatures to HTTP headers with specific HTTP methods.

```
Stream Content
POST /veridata HTTP/1.1
Host: 1GSNNipOH
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; en-GB)
Accept: */*
Content-type: text/xml; charset=utf-8
SOAPAction: ""
Connection: keep-alive
Content-Length: 21032

<?xml version="1.0" encoding="utf-8"?><soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema' xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Header><ns1:authorization-handle
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next' soapenv:mustUnderstand="0" xmlns:ns1='http://www.goldengate.com/
veridata'>5565832287636947043</ns1:authorization-
handle><ns2:mWDQgJXSzKjmKRlFdTcvGFffuZdRSTewFAcbfZTtVrzBfKvQxdjhgAsWkHADpJMvVxBfbiuiesbssthThrdRDoXTYycLVdHdSLSinrcQJGriBSNHeu
NxLExPYOiGzGhycCyKnxFxaqbxkdzwmicuLgsbbUUUhqevaXsigYpFoKyvqqXhERsbakHpDINYFLZFThxJHdCNyPPLhHAFuGVUhbDfJYbREtYLoAmKwuxRWGMdnLEs
yzMYveXlzTlMhZSsMONzVkqZTtzCNQsOqbWmkYaJlcdAFKRFBSdbjpBdGrWbwsLDSkyjxjuwbrQTcySXqTdKKQzQLcMfWYBwKMcCHbvjrLndglBiCqVpplWtFqhPqI
FMnriLQACVBkXJipzSMzkNaNQKHBHVtoxFRsyMkVwYYtFsTYavzPanbXEkDDzeSQAWlFYJnBNbzrHGKvVhHpMzKnJISvZsBzoASldobJohHHkYoLoHepfZVLERhtv
```

## http-req-mime-form-data

MIME header data in the body of an HTTP request, not including embedded file contents.

### Additional Details

None

### Context Capture

This context provides the data highlighted in yellow.

```
▷ Hypertext Transfer Protocol
▽ MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "oOoOoOo"
    [Type: multipart/form-data]
    First boundary: --oOoOoOo\r\n
  ▽ Encapsulated multipart part:
      Content-Disposition: form-data; name="Filedata"; filename="nTYltlmY.php"\r\n\r\n
    ▷ Data (1798 bytes)
    Boundary: \r\n--oOoOoOo\r\n
    Encapsulated multipart part:
▽ [Malformed Packet: MIME multipart]
  ▽ [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
      [Message: Malformed Packet (Exception occurred)]
      [Severity level: Error]
      [Group: Malformed]
```

## http-req-ms-subdomain

Identifies the request headers/params which can be used to identify office365-enterprise-access. The example below (X-User-Identity) is one of several headers that can be used to identify the office365-enterprise account.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
<Accept: application/mapi-http\r\n>
User-Agent: Microsoft Office/16.0 (Windows NT 6.1; Microsoft Outlook 16.0.6528; Pro)\r\n
<User-Agent: Microsoft Office/16.0 (Windows NT 6.1; Microsoft Outlook 16.0.6528; Pro)\r\n>
X-MS-CookieUri-Requested: t\r\n
X-FeatureVersion: 1\r\n
Client-Request-Id: {4E3223DE-DB95-45BD-9D43-B42507D4D287}\r\n
X-ClientApplication: Outlook/16.0.6568.1463\r\n
X-ClientInfo: {605B28C7-8DD5-45B5-BC43-748072EEB3BE}:24\r\n
X-User-Identity: [redacted] org\r\n
X-RequestId: {69840C3D-E6D0-4794-BB04-FFDDEBFCEA24}:2\r\n
X-RequestType: Connect\r\n
▼ Content-Length: 374\r\n
```

## http-req-origin-headers

Identifies strings used to match against strings from the origin field. You must operate PAN-OS 8.1 or later to use this field.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
Content-Type: application/json+protobuf\r\n
<Content-Type: application/json+protobuf\r\n>
Accept: */*\r\n
<Accept: */*\r\n>
Origin: https://hangouts.google.com\r\n
X-Client-Data: CIS2yQEIo7bJAQjEtskBCKmdygEIlqzKAQiZwsoBCPjHygEInd3KAQjUnMsBCOScywEIqJ3LAQ==\r\n
Sec-Fetch-Site: same-site\r\n
Sec-Fetch-Mode: cors\r\n
Sec-Fetch-Dest: empty\r\n
Referer: https://hangouts.google.com/\r\n
<Referer: https://hangouts.google.com/\r\n>
Accept-Encoding: gzip, deflate, br\r\n
```

## http-req-params

Query string as well as parameters in the HTTP body for a POST method (after the '?').

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow. Qualifiers: This context can use the HTTP method qualifier to limit signatures to HTTP headers with specific HTTP methods.

```
▽ Hypertext Transfer Protocol
  ▷ GET /blog/wp-admin/images/wordpress-logo.png?ver=20120216 HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Referer: http://thelegendofrandom.com/blog/wp-login.php\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f; wordpress_test_cookie=WP+Cookie+
    \r\n
    [Full request URI: http://thelegendofrandom.com/blog/wp-admin/images/wordpress-logo.png?ver=20120216]
```

## http-req-uri

The URI path and parameters in a HTTP header request.

### Additional Details

Available only in PAN-OS 10.0 and later releases.

### Context Capture

This context provides the text highlighted in yellow. Qualifiers: This context can use the HTTP method qualifier to limit signatures to HTTP headers with specific HTTP methods.

```
▽ Hypertext Transfer Protocol
  ▷ GET /blog/wp-admin/images/wordpress-logo.png?ver=20120216 HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Referer: http://thelegendofrandom.com/blog/wp-login.php\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f; wordpress_test_cookie=WP+Cookie+
    \r\n
    [Full request URI: http://thelegendofrandom.com/blog/wp-admin/images/wordpress-logo.png?ver=20120216]
```

## http-req-uri-path

Path in a HTTP request header (up to and including the '?').

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

Qualifiers: This context can use the HTTP method qualifier to limit signatures to HTTP headers with specific HTTP methods.

```
▽ Hypertext Transfer Protocol
  ▷ GET /blog/wp-admin/images/wordpress-logo.png?ver=20120216 HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Referer: http://thelegendofrandom.com/blog/wp-login.php\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f; wordpress_test_cookie=WP+Cookie+
    \r\n
    [Full request URI: http://thelegendofrandom.com/blog/wp-admin/images/wordpress-logo.png?ver=20120216]
```

## http-req-user-agent-header

The user agent field in an HTTP request header.

### Context Capture

This context covers the area called out in red.

```
▼ Hypertext Transfer Protocol
  ▼ GET / HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /
      Request Version: HTTP/1.1
    Host: 127.0.0.1:8000\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.6
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exch
    Sec-Fetch-Site: none\r\n
    Sec-Fetch-Mode: navigate\r\n
    Sec-Fetch-User: ?1\r\n
    Sec-Fetch-Dest: document\r\n
    Accept-Encoding: gzip, deflate, br\r\n
    Accept-Language: en-US,en;q=0.9\r\n
  ▶ Cookie: wp-settings-time-1=1586134406; wp-settings-1=mfold%3Do; csrftoken=3ghnm6c7Nh9GTk3szE9AmwWYCWv2NcnO2iLlkitnIso
    \r\n
    [Full request URI: http://127.0.0.1:8000/]
    [HTTP request 1/1]
    [Response in frame: 11]
```

## http-rsp-headers

Full HTTP response header, not including the HTTP banner.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Hypertext Transfer Protocol
  ▽ HTTP/1.1 200 OK\r\n
    ▷ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Request Version: HTTP/1.1
      Status Code: 200
      Response Phrase: OK
    X-DB-Timeout: 120\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    Content-Type: text/plain\r\n
    Date: Mon, 22 Jul 2013 23:18:55 GMT\r\n
  ▷ Content-Length: 15\r\n
```

## http-rsp-non-2xx-response-body

Body of non-2xx HTTP response, excluding HTTP 406 (Not Acceptable) responses.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
GET /exception HTTP/1.1
Host: tacdevbox:5000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.0 403 FORBIDDEN
Content-Type: text/html; charset=utf-8
Content-Length: 13773
Server: Werkzeug/0.11.9 Python/2.7.8
Date: Mon, 13 Mar 2017 22:26:02 GMT

<!DOCTYPE html>
<html>
```

## http-rsp-reason

The HTTP response status reason.

### Context Capture

```
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.0 404 File not found\r\n
    Server: SimpleHTTP/0.6 Python/3.7.6\r\n
    Date: Tue, 02 Jun 2020 16:53:18 GMT\r\n
    Connection: close\r\n
    Content-Type: text/html;charset=utf-8\r\n
  ▶ Content-Length: 469\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.000694000 seconds]
    [Request in frame: 117]
    [Request URI: http://127.0.0.1:8000/admin]
```

## icmp-req-code

Identifies the ICMP request message code number.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
▶ Ethernet II, Src: VMware_81:41:76 (00:50:56:81:41:76), Dst: PaloAlto_e8:c0:12 (b4:0c:25:e8:c0:12)
▶ Internet Protocol Version 4, Src: 10.46.174.10, Dst: 142.250.189.238
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x510f [correct]
    [Checksum Status: Good]
    Identifier (BE): 31427 (0x7ac3)
    Identifier (LE): 50042 (0xc37a)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
  ▶ [No response seen]
    Timestamp from icmp data: Oct  8, 2021 14:04:32.000000000 PDT
    [Timestamp from icmp data (relative): 30.061097000 seconds]
  ▶ Data (48 bytes)
```

## icmp-req-data

Identifies the ICMP payload request message.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
▶ Ethernet II, Src: 00:66:aa:d1:32:c2 (00:66:aa:d1:32:c2), Dst: 00:44:66:fc:29:af (00:44:66:fc:29:af)
▶ Internet Protocol Version 4, Src: 10.47.116.62, Dst: 8.8.8.8
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x662a [correct]
    [Checksum Status: Good]
    Identifier (BE): 46405 (0xb545)
    Identifier (LE): 17845 (0x45b5)
    Sequence Number (BE): 13 (0x000d)
    Sequence Number (LE): 3328 (0x0d00)
  ▶ [No response seen]
    Timestamp from icmp data: Oct 13, 2021 10:45:33.554195000 PDT
    [Timestamp from icmp data (relative): 0.000095000 seconds]
  ▼ Data (48 bytes)
      Data: 08090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b…
      [Length: 48]
```

## icmp-req-type

Identifies the ICMP request message type number.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
▶ Ethernet II, Src: VMware_81:41:76 (00:50:56:81:41:76), Dst: PaloAlto_e8:c0:12 (b4:0c:25:e8:c0:12)
▶ Internet Protocol Version 4, Src: 10.46.174.10, Dst: 142.250.189.238
▼ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x510f [correct]
    [Checksum Status: Good]
    Identifier (BE): 31427 (0x7ac3)
    Identifier (LE): 50042 (0xc37a)
    Sequence Number (BE): 1 (0x0001)
    Sequence Number (LE): 256 (0x0100)
  ▶ [No response seen]
    Timestamp from icmp data: Oct  8, 2021 14:04:32.000000000 PDT
    [Timestamp from icmp data (relative): 30.061097000 seconds]
  ▶ Data (48 bytes)
```

## icmp-req-protocol-payload

The payload of an ICMP request.

### Context Capture

```
▼ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x325c [correct]
    [Checksum Status: Good]
    Identifier (BE): 512 (0x0200)
    Identifier (LE): 2 (0x0002)
    Sequence number (BE): 8448 (0x2100)
    Sequence number (LE): 33 (0x0021)
    [Request frame: 1]
    [Response time: 216.795 ms]
  ▼ Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f707172737475767761…
      [Length: 32]
```

## icmp-rsp-data

Identifies the ICMP payload response message.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
▼ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x6863 [correct]
    [Checksum Status: Good]
    Identifier (BE): 0 (0x0000)
    Identifier (LE): 0 (0x0000)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
    [Request frame: 1]
    [Response time: 0.750 ms]
  ▶ Data (33 bytes)

0000  00 0c 29 47 fd 3f 00 0c  29 36 84 ba 08 00 45 00   ··)G·?·· )6····E·
0010  00 3d ed 0f 00 00 40 01  8d 16 c0 a8 3f b0 c0 a8   ·=····@· ····?···
0020  3f 99 00 00 68 63 00 00  00 00 74 65 73 74 2e 74   ?···hc·· ··test.t
0030  78 74 3a 3a 2d 33 35 31  31 35 39 35 31 39 3a 3a   xt::-351 159519::
0040  39 12 13 14 15 12 11 13  12 12 12                  9····· ···
```

## icmp-rsp-protocol-payload

The payload of an ICMP response.

### Context Capture

```
▼ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0x325c [correct]
    [Checksum Status: Good]
    Identifier (BE): 512 (0x0200)
    Identifier (LE): 2 (0x0002)
    Sequence number (BE): 8448 (0x2100)
    Sequence number (LE): 33 (0x0021)
    [Request frame: 1]
    [Response time: 216.795 ms]
  ▼ Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f707172737475767761…
      [Length: 32]
```

## icmp-req-possible-custom-payload

This is not a context but a value that you can add to your custom signature to detect custom payloads in ICMP requests.

#### Context Capture

None

## ike-req-headers

Full IKE header from the requester, including the initiator's SPI, next payload, major version, minor version, exchange type, flags, message ID, and length.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow.

```
> Frame 1: 863 bytes on wire (6904 bits), 863 bytes captured (6904 bits)
> Ethernet II, Src: CheckPoi_31:43:30 (00:a0:8e:31:43:30), Dst: Cisco_5d:18:1a (00:09:b6:5d:18:1a)
> Internet Protocol Version 4, Src: 134.154.229.8, Dst: 130.150.239.180
> User Datagram Protocol, Src Port: 500, Dst Port: 500
v Internet Security Association and Key Management Protocol
    Initiator SPI: 86973c9ad71c18ec
    Responder SPI: 0000000000000000
    Next payload: Security Association (1)
  > Version: 1.0
    Exchange type: Aggressive (4)
  > Flags: 0x00
    Message ID: 0x00000000
    Length: 805
  > Payload Security Association: (1)
  > Payload: Key Exchange (4)
  > Payload: Nonce (10)
```

## ike-rsp-headers

Full IKE header from the responder, including the responder's SPI, next payload, major version, minor version, exchange type, flags, message ID, and length.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow.

```
Internet Security Association and Key Management Protocol
    Initiator SPI: 0000000000000000
    Responder SPI: 0000000000000000
    Next payload: Notification (11)
  > Version: 1.0
    Exchange type: Informational (5)
  > Flags: 0x00
    Message ID: 0xfb8c726e
    Length: 40
  v Payload: Notification (11)
        Next payload: NONE / No Next Payload  (0)
        Reserved: 00
        Payload length: 12
        Domain of interpretation: ISAKMP (0)
```

## ike-req-payload-text

Full security association request payload, including the proposal and transform substructures.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
v Internet Security Association and Key Management Protocol
    Initiator SPI: 86973c9ad71c18ec
    Responder SPI: 0000000000000000
    Next payload: Security Association (1)
  > Version: 1.0
    Exchange type: Aggressive (4)
  > Flags: 0x00
    Message ID: 0x00000000
    Length: 805
  > Payload: Security Association (1)
  > Payload: Key Exchange (4)
  > Payload: Nonce (10)
  > Payload: Identification (5)
  > Payload: Vendor ID (13) : XAUTH
  > Payload: Vendor Id (13) : RFC 3706 DPD (Dead Peer Detection)
  > Payload: Vendor ID (10) : CISCO-UNITY 1.0
```

## ike-rsp-payload-text

Full security association response payload, including the proposal and transform substructures.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
> Version: 1.0
  Exchange type: Informational (5)
> Flags: 0x00
  Message ID: 0xfb8c726e
  Length: 40
v Payload: Notification (11)
      Next payload: NONE / No Next Payload  (0)
      Reserved: 00
      Payload length: 12
      Domain of interpretation: ISAKMP (0)
      Protocol ID: ISAKMP (1)
      SPI Size: 0
      Notify Message Type: INVALID-EXCHANGE-TYPE (7)
      Notification DATA: <MISSING>
```

## imap-req-cmd-line

IMAP command used.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Internet Message Access Protocol
  ▽ Line: LOGIN root vpn123\r\n
      Request Tag: LOGIN
      Request: root vpn123
```

## imap-req-first-param

First parameter to an IMAP command.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow. This context can use the IMAP command qualifier to limit signatures to specific IMAP commands.

```
▽ Internet Message Access Protocol
  ▽ Line: LOGIN root vpn123\r\n
      Request Tag: LOGIN
      Request: root vpn123
```

## imap-req-params-after-first-param

Every parameter to an IMAP command, not including the first parameter.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Internet Message Access Protocol
  ▽ Line: LOGIN root vpn123\r\n
      Request Tag: LOGIN
      Request: root vpn123
```

## imap-req-protocol-payload

The payload of an IMAP request.

### Context Capture

```
▼ Internet Message Access Protocol
  ▼ Line: A0 ID ("foo")\n
      Request Tag: A0
      Request Command: ID
      Request: ID ("foo")
      [Response In: 8]
```

## imap-rsp-protocol-payload

The payload of an IMAP response.

### Context Capture

```
▼ Internet Message Access Protocol
    Line: * OK [CAPABILITY IMAP4rev1 SASL-IR LOGIN-REFERRALS ID ENABLE IDLE LITERAL+ AUTH=PLAIN] Dovecot ready.\r\n
```

## irc-req-params

Argument after the actual IRC command and space.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Internet Relay Chat
  ▽ Request: NICK nickblock
      Command: NICK
    ▷ Command parameters
```

## irc-req-prefix

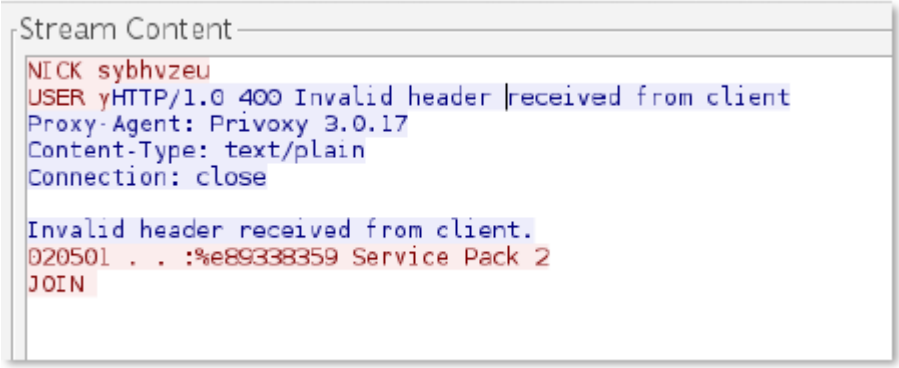
Data before an IRC command, typically used to indicate the true origin of a message.

### Additional Details

None

### Context Capture

You can see by following the TCP stream in Wireshark that there is data in between the IRC commands. It appears this message was Proxied.




## jpeg-file-scan-data

This context provides all of the scan data within a JPEG file.

### Additional Details

None

## jpeg-file-segment-data

This context provides all of the segment data within a JPEG file.

### Additional Details

None

## jpeg-file-segment-header

This context provides the segment header data within a JPEG file.

### Additional Details

None

## ldap-req-searchrequest-baseobject

Identifies the base object for the LDAP searchRequest entry.

### Additional Details

None

#### Context Capture

This context provides the highlighted text.

```
    messageID: 2
  ▼ protocolOp: searchRequest (3)
    ▼ searchRequest
        baseObject: dc=buh,dc=is,dc=keysight,dc=com
        scope: wholeSubtree (2)
        derefAliases: neverDerefAliases (0)
        sizeLimit: 0
        timeLimit: 0
        typesOnly: False
```

## ldap-rsp-searchresentry-objectname

Identifies the objectName for the LDAP searchResEntry.

#### Additional Details

None

#### Context Capture

This context provides the highlighted text.

```
▼ Lightweight Directory Access Protocol
  ▼ LDAPMessage searchResEntry(8) "CN=Computers,dc=teva,dc=test" [49 results]
      messageID: 8
    ▼ protocolOp: searchResEntry (4)
      ▼ searchResEntry
          objectName: CN=Computers,dc=teva,dc=test
        ▶ attributes: 16 items
      [Response To: 4]
```

## ms-ds-smb-req-share-name

Full path to a file that is read or written using SMB.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow.

```
▽ SMB (Server Message Block Protocol)
  ▷ SMB Header
  ▽ Trans2 Response (0x32)
      Subcommand: FIND_FIRST2 (0x0001)
      [Level of Interest: Find File Both Directory Info (260)]
      [Search Pattern: \Departments\Public\Dallas_Sites\Dallas Trip\Site 2\*]
      Word Count (WCT): 10
      Total Parameter Count: 10
      Total Data Count: 1276
      Reserved: 0000
      Parameter Count: 10
```

## ms-ds-smb-req-v1-create-filename

This field identifies the SMBv1 NT Create AndX filename.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▼ SMB (Server Message Block Protocol)
  ▶ SMB Header
  ▼ NT Create AndX Request (0xa2)
      [FID: 0x4000]
      Word Count (WCT): 24
      AndXCommand: No further commands (0xff)
      Reserved: 00
      AndXOffset: 0
      Reserved: 00
      File Name Len: 9
    ▶ Create Flags: 0x00000016
      Root FID: 0x00000000
    ▶ Access Mask: 0x0002019f
      Allocation Size: 0
    ▶ File Attributes: 0x00000000
    ▶ Share Access: 0x00000003, Read, Write
      Disposition: Open (if file exists open it, else fail) (1)
    ▶ Create Options: 0x00000040
      Impersonation: Impersonation (2)
    ▶ Security Flags: 0x03, Context Tracking, Effective Only
      Byte Count (BCC): 9
      File Name: \BROWSER
```

```
0000  58 49 3b eb 27 10 de ad  01 40 a8 c0 08 00 45 00   XI;·'··· ·@····E·
0010  00 88 5f cd 00 00 ff 06  75 94 c0 a8 37 77 c0 a8   ··_····· u···7w··
0020  2d 46 ba 17 01 bd cb d5  92 2d fd 58 3a b8 50 18   -F¯····· ·-·X:·P·
0030  ff ff 1a 00 00 00 00 00  00 5c ff 53 4d 42 a2 00   ····· ·\·SMB··
0040  00 00 00 00 01 20 00 00  00 00 00 00 00 00 00 00   ····· ·· ·····
0050  00 00 00 08 03 d6 00 08  10 27 18 ff 00 00 00 00   ····· ·'·····
0060  09 00 16 00 00 00 00 00  00 00 9f 01 02 00 00 00   ····· ·····
0070  00 00 00 00 00 00 00 00  00 00 03 00 00 00 01 00   ····· ·····
0080  00 00 40 00 00 00 02 00  00 00 03 09 00 5c 42 52   ··@····· ·····\BR
0090  4f 57 53 45 52 00                                  OWSER·
```

## ms-ds-smb-req-v2-create-filename

This field identifies the SMBv2/SMBv3 Create filename.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▼ SMB2 (Server Message Block Protocol version 2)
  ▶ SMB2 Header
  ▼ Create Request (0x05)
    ▶ StructureSize: 0x0039
      Oplock: No oplock (0x00)
      Impersonation level: Impersonation (2)
      Create Flags: 0x0000000000000000
      Reserved: 0000000000000000
    ▶ Access Mask: 0x00100081
    ▶ File Attributes: 0x00000000
    ▶ Share Access: 0x00000007, Read, Write, Delete
      Disposition: Open (if file exists open it, else fail) (1)
    ▶ Create Options: 0x00000001
    ▶ Filename: test
      Blob Offset: 0x00000080
      Blob Length: 88
    ▶ ExtraInfo SMB2_CREATE_DURABLE_HANDLE_REQUEST SMB2_CREATE_QU

0030  ff ff 29 bc 00 00 00 00  01 38 fe 53 4d 42 40 00   ··)····· ·8·SMB@·
0040  01 00 00 00 00 00 05 00  01 00 08 00 00 00 d8 00   ........ ........
0050  00 00 2a 00 00 00 00 00  00 00 ff fe 00 00 01 00   ··*····· ........
0060  00 00 35 00 00 d8 01 04  00 00 d2 51 a9 2b b9 2d   ··5····· ···Q·+·-
0070  28 d5 7f 78 11 fa d1 3f  cb 51 39 00 00 00 02 00   (··x···? ·Q9·····
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 81 00 10 00 00 00  00 00 07 00 00 00 01 00   ........ ........
00a0  00 00 01 00 00 00 78 00  08 00 80 00 00 00 58 00   ......x· ......X·
00b0  00 00 74 00 65 00 73 00  74 00 28 00 00 00 10 00   ··t·e·s· t·(·····
00c0  04 00 00 00 18 00 10 00  00 00 44 48 6e 51 00 00   ........ ··DHnQ··
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 18 00 00 00 10 00  04 00 00 00 18 00 00 00   ........ ........
00f0  00 00 4d 78 41 63 00 00  00 00 00 00 00 00 10 00   ··MxAc·· ........
0100  04 00 00 00 18 00 00 00  00 00 51 46 69 64 00 00   ........ ··QFid··
0110  00 00 fe 53 4d 42 40 00  01 00 00 00 00 00 0f 00   ···SMB@· ........
```

## msrpc-req-bind-data

Data payload of a MS RPC Bind request.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow. The easiest way to find a pattern to match is to look at the hex representation of the payload and pick at least 7 bytes to match on as seen above.

```
▽ Transmission Control Protocol, Src Port: 45622 (45622), Dst Port: netbios-ssn (139)
    Source port: 45622 (45622)
    Destination port: netbios-ssn (139)
    [Stream index: 0]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 468    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header length: 20 bytes
  ▷ Flags: 0x018 (PSH, ACK)
    Window size value: 16384
    [Calculated window size: 16384]

00a0  00 00 00 00 00 00 00 00  00 00 00 00 03 00 00 00   ........ ........
00b0  01 00 00 00 40 00 40 00  02 00 00 00 01 11 00 00   ....@.@. ........
00c0  5c 00 73 00 72 00 76 00  73 00 76 00 63 00 00 00   \.s.r.v. s.v.c...
00d0  0a 2f 00 fe 00 00 40 00  00 00 00 ff ff ff ff 80   ./....@. ........
00e0  00 48 00 00 00 48 00 b6  00 00 00 00 00 49 00 ee   .H...H.. .....I..
00f0  05 00 0b 03 10 00 00 00  10 02 00 00 01 00 00 00   ........ ........
0100  b8 10 b8 10 00 00 00 00  01 00 00 00 00 00 01 00   ........ ........
0110  c8 4f 32 4b 70 16 d3 01  12 78 5a 47 bf 6e e1 88   .O2Kp... .xZG.n..
0120  03 00 00 00 04 5d 88 8a  eb 1c c9 11 9f e8 08 00   .....]..
0130  2b 10 48 60 02 00 00 00  0e ff 00 de de 00 40 00   +.H`.... ......@.
0140  00 00 00 ff ff ff ff 80  00 48 00 00 00 ff 01 ca   ........ .H......
0150  01 00 00 00 00 49 00 ee  ed 1e 94 7c 90 81 c4 ff   .....I.. ...|....
0160  ef ff ff 44 31 c9 83 a9  dd d9 ae d9 74 24 f4 5b   ...D1... ....t$.[
0170  81 73 13 a9 d1 80 f5 83  eb fc e2 f4 55 39 c4 f5   .s...... ....U9..
0180  a9 d1 0b b0 95 5a fc f0  d1 d0 6f 7e e6 c9 0b aa   .....Z.. ..o~....
0190  89 d0 6b bc 22 e5 06 f4  47 e0 40 6c 05 55 40 81   ..k."... G.@l.U@.
01a0  ae 10 4a f8 a8 13 6b 01  92 85 a4 f1 dc 34 0b aa   ..J...k. .....4..
01b0  8d d0 6b 93 22 dd cb 7e  f6 cd 81 1e 22 cd 0b f4   ..k."..~ ....."...
01c0  42 58 dc d1 ad 12 b1 35  cd 5a c0 c5 2c 11 f8 f9   BX.....5 .Z..,...
01d0  22 91 8c 7e d9 cd 2d 7e  c1 d9 6b fc 22 51 30 f5   "..~..-~ ..k."Q0.
01e0  a9 d1 0b 9d 95 0e b1 03  c9 87 09 0d 2a 11 fb a5   ........ ....*...
01f0  c1 af 58 17 da b9 18 0b  23 df d7 0a 4e b2 e1 99   ..X..... #...N...
0200  ca ff e5 8d cc d1 90 f5  90                        ........ .
```

## mssql-db-req-body

Request to a Microsoft SQL server, excluding the request header.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Tabular Data Stream
    Type: SQL batch (1)
  ▷ Status: 0x01
    Length: 112
    Channel: 0
    Packet Number: 1
    Window: 0
  ▽ TDS Query Packet
      Query: DELETE FROM ainfluencia WHERE refnac05 = '204050001'
```

## netbios-dg-req-protocol-payload

The payload of a NetBIOS Datagram Service request.

### Context Capture

```
▼ NetBIOS Datagram Service
    Message Type: Direct_group datagram (17)
  ▶ Flags: 0x02, This is first fragment, Node Type: B node
    Datagram ID: 0x8661
    Source IP: 192.168.2.16
    Source Port: 138
    Datagram length: 187 bytes
    Packet offset: 0 bytes
    Source name: ESURNIR-LAPTOP<20> (Server service)
    Destination name: MSHOME<1e> (Browser Election Service)
```

## netbios-dg-rsp-protocol-payload

The payload of a NetBIOS Datagram Service response.

### Context Capture

```
▼ NetBIOS Datagram Service
    Message Type: Direct_group datagram (17)
  ▶ Flags: 0x02, This is first fragment, Node Type: B node
    Datagram ID: 0x8665
    Source IP: 192.168.2.16
    Source Port: 138
    Datagram length: 201 bytes
    Packet offset: 0 bytes
    Source name: ESURNIR-LAPTOP<00> (Workstation/Redirector)
    Destination name: <01><02>__MSBROWSE__<02><01> (Browser)
```

## netbios-ns-req-protocol-payload

The payload of a NetBIOS Name Service request.

**Context Capture**

```
▼ NetBIOS Name Service
    Transaction ID: 0x2f53
  ▼ Flags: 0x0110, Opcode: Name query, Recursion desired, Broadcast
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Name query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... ...1 .... = Broadcast: Broadcast packet
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ POTATO<00>: type NB, class IN
        Name: POTATO<00> (Workstation/Redirector)
        Type: NB (32)
        Class: IN (1)
```

## netbios-ns-rsp-protocol-payload

The payload of a NetBIOS Name Service response.

**Context Capture**

```
▼ NetBIOS Name Service
    Transaction ID: 0x2f53
  ▼ Flags: 0x8580, Response, Opcode: Name query, Authoritative, Recursion desired, Recursion available, Reply code: No error
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Name query (0)
      .... .1.. .... .... = Authoritative: Server is an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 1... .... = Recursion available: Server can do recursive queries
      .... .... ...0 .... = Broadcast: Not a broadcast packet
      .... .... .... 0000 = Reply code: No error (0)
    Questions: 0
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▼ Answers
    ▼ POTATO<00>: type NB, class IN
        Name: POTATO<00> (Workstation/Redirector)
        Type: NB (32)
        Class: IN (1)
        Time to live: 3 days
        Data length: 6
      ▶ Name flags: 0x0000, ONT: B-node (B-node, unique)
        Addr: 192.168.1.65
```

## nettcp-req-context

Checks the RequestContext field in Net.TCP (App-ID: net.tcp) requests.

**Additional Details**

None

## oracle-req-data-text

When the firewall detects an Oracle request and the request type is DATA, this context returns the data contained in the request.

#### Additional Details

None

#### Context Capture

```
▶ Frame 8: 210 bytes on wire (1680 bits), 210 bytes captured (1680 bits)
▶ Ethernet II, Src: Vmware_3b:fa:b1 (00:0c:29:3b:fa:b1), Dst: Vmware_8f:ca:a1 (00:0c:29:8f:ca:a1)
▶ Internet Protocol Version 4, Src: 192.168.10.17, Dst: 192.168.10.10
▶ Transmission Control Protocol, Src Port: 1255 (1255), Dst Port: 1521 (1521), Seq: 1011314681, Ack: 320022589, Len: 156
▼ Transparent Network Substrate Protocol
    Packet Length: 156
    Packet Checksum: 0x0000
    Packet Type: Data (6)
    Reserved Byte: 00
    Header Checksum: 0x0000
  ▼ Secure Network Services
    ▶ Data Flag: 0x0000
    ▼ Data (146 bytes)
        Data: deadbeef00920a200100000400000400030000000000400...
        [Length: 146]
```

## pe-dos-headers

The DOS MZ header and the DOS stub are located in the first 64 bytes of the PE file.

#### Additional Details

None

#### Context Capture

This context provides the data in bold.

| PE File Structure |
|---|
| **DOS MZ Header + DOS Stub – first 64 bytes** |
| PE File Header – next 20 bytes |
| PE Optional Header – next 224 bytes |
| PE Section Header – next 40 bytes each |
| PE Body Data – Rest of the file |

## pe-file-header

The PE file header is 20 bytes long and starts at the 65th byte of the PE file.

#### Additional Details

None

#### Context Capture

This context provides the data in bold.

| PE File Structure |
| --- |
| DOS MZ Header + DOS Stub – first 64 bytes |
| **PE File Header – next 20 bytes** |
| PE Optional Header – next 224 bytes |
| PE Section Header – next 40 bytes each |
| PE Body Data – Rest of the file |

## pe-optional-header

The optional header of a PE file is typically 224 bytes long and starts at the 86th byte of the PE file.

#### Additional Details

None

#### Context Capture

This context provides the data in bold.

| PE File Structure |
| --- |
| DOS MZ Header + DOS Stub – first 64 bytes |
| PE File Header – next 20 bytes |
| **PE Optional Header – next 224 bytes** |
| PE Section Header – next 40 bytes each |
| PE Body Data – Rest of the file |

## pe-section-header

This context provides the section headers for a PE file.

#### Additional Details

These headers are 40 bytes each. Some typical sections with headers are “idata”, “rsrc”, “data”, “text”, and “src”. However, each PE file may not include each section and the sections are not guaranteed to be in any specific order.

#### Context Capture

This context provides the data in bold.

| PE File Structure |
| --- |
| DOS MZ Header + DOS Stub – first 64 bytes |
| PE File Header – next 20 bytes |
| PE Optional Header – next 224 bytes |
| **PE Section Header – next 40 bytes each** |
| PE Body Data – Rest of the file |

## pe-body-data

This context provides the body data of a PE file, which includes everything inside the file sections themselves.

### Additional Details

None

### Context Capture

This context provides the data in bold.

| PE File Structure |
| --- |
| DOS MZ Header + DOS Stub – first 64 bytes |
| PE File Header – next 20 bytes |
| PE Optional Header – next 224 bytes |
| PE Section Header – next 40 bytes each |
| **PE Body Data – Rest of the file** |
| |

## pop3-req-protocol-payload

The payload of a POP3 request.

### Context Capture

```
▼ Post Office Protocol
  ▼ AUTH PLAIN\r\n
      Request command: AUTH
      Request parameter: PLAIN
```

## pop3-rsp-protocol-payload

The payload of a POP3 response.

### Context Capture

```
▼ Post Office Protocol
  ▼ +OK POP server ready H mimap13 0MW5rZ-1VayeZ2jFp-00XVZd\r\n
      Response indicator: +OK
      Response description: POP server ready H mimap13 0MW5rZ-1VayeZ2jFp-00XVZd
```

## pre-app-req-data

This field provides the request data before the firewall App-ID has identified the traffic.

### Additional Details

Firewall traffic that cannot be identified by App-ID due to inadequate data for signature matching is designated as **insufficient data** in the application field of the traffic logs.

**Context Capture**

```
▶ Frame 27: 951 bytes on wire (7608 bits), 951 bytes captured (7608 bits)
▶ Ethernet II, Src: PcsCompu_1e:f8:f0 (08:00:27:1e:f8:f0), Dst: PcsCompu_bd:00:0f (0
▶ Internet Protocol Version 4, Src: 192.168.1.226, Dst: 192.168.1.151
▶ Transmission Control Protocol, Src Port: 37138, Dst Port: 4567, Seq: 68, Ack: 13,
▼ Data (885 bytes)
0000  08 00 27 bd 00 0f 08 00  27 1e f8 f0 08 00 45 00   ··'····· '·····E·
0010  03 a9 52 7e 40 00 40 06  60 07 c0 a8 01 e2 c0 a8   ··R~@·@· `·******
0020  01 97 91 12 11 d7 90 f7  34 7f 80 8d 8b 09 80 18   ******** 4*******
0030  01 f6 0b c8 00 00 01 01  08 0a 87 89 c9 a8 0b b3   ******** ********
0040  8e 1e 65 6e 70 30 73 33  3a 20 66 6c 61 67 73 3d   ··enp0s3 : flags=
0050  34 31 36 33 3c 55 50 2c  42 52 4f 41 44 43 41 53   4163<UP, BROADCAS
0060  54 2c 52 55 4e 4e 49 4e  47 2c 4d 55 4c 54 49 43   T,RUNNIN G,MULTIC
0070  41 53 54 3e 20 20 6d 74  75 20 31 35 30 30 0a 20   AST>  mt u 1500·
0080  20 20 20 20 20 20 20 69  6e 65 74 20 31 39 32 2e          i net 192.
0090  31 36 38 2e 31 2e 32 32  36 20 20 6e 65 74 6d 61   168.1.22 6  netma
00a0  73 6b 20 32 35 35 2e 32  35 35 2e 32 35 35 2e 30   sk 255.2 55.255.0
00b0  20 20 62 72 6f 61 64 63  61 73 74 20 31 39 32 2e     broadc ast 192.
```

## pre-app-rsp-data

This field provides the response data before the firewall App-ID has identified the traffic.

### Additional Details

Firewall traffic that cannot be identified by App-ID due to inadequate data for signature matching is designated as **insufficient data** in the application field of the traffic logs.

**Context Capture**

```
    [Calculated window size: 65280]
    [Window size scaling factor: 128]
    Checksum: 0xee81 [correct]
    [Checksum Status: Good]
    [Calculated Checksum: 0xee81]
    Urgent pointer: 0
  ▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▶ [SEQ/ACK analysis]
  ▶ [Timestamps]
    TCP payload (9 bytes)
▼ Data (9 bytes)
    Data: 6966636f6e6669670a
    [Length: 9]
0000  08 00 27 1e f8 f0 08 00  27 bd 00 0f 08 00 45 00   ··'····· '·····E·
0010  00 3d 0a 4b 40 00 40 06  ab a6 c0 a8 01 97 c0 a8   ·=·K@·@· ********
0020  01 e2 11 d7 91 12 80 8d  8b 00 90 f7 34 7f 80 18   ******** ····4···
0030  01 fe ee 81 00 00 01 01  08 0a 0b b3 8e 1e 87 89   ******** ********
0040  bd 75 69 66 63 6f 6e 66  69 67 0a                  ·uifconf ig·
```

## rtmp-req-message-body

RTMP body up until twenty packets have been sent.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Real Time Messaging Protocol (AMF0 Command connect('StreamPlayer/'))
    Response to this call in frame: 19
  ▷ RTMP Header
  ▽ RTMP Body
    ▽ String 'connect'
        AMF type: String (2)
        String length: 7
        String: connect
    ▽ Number 1
        AMF type: Number (0)
        Number: 1
    ▽ Object (9 items)
        AMF type: Object (3)
      ▷ Property 'app' String 'StreamPlayer/'
      ▷ Property 'flashVer' String 'WIN 9,0,47,0'
      ▷ Property 'swfUrl' String 'file://'
      ▷ Property 'tcUrl' String 'rtmp://fc432.streamedia.info/StreamPlayer/'
      ▷ Property 'fpad' Boolean false
      ▷ Property 'audioCodecs' Number 615
      ▷ Property 'videoCodecs' Number 124
      ▷ Property 'videoFunction' Number 1
      ▷ Property 'pageUrl' Undefined
        End Of Object Marker
```

## rtsp-req-headers

Full RTSP request headers, not including the command line.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow. Qualifier: This context can use the RTSP method qualifier to limit signatures to specific RTSP methods.

```
▽ Real Time Streaming Protocol
  ▷ Request: DESCRIBE rtsp://media.real.com:554/showcase/ads/spicealicious_15.rm RTSP/1.0\r\n
    CSeq: 2\r\n
    User-Agent: RealMedia Player (HelixDNAClient)/10.0.0.0 (win32)\r\n
    Accept: application/sdp\r\n
    Session: 315291390-1
    Bandwidth: 1544000\r\n
    ClientID: WinNT_5.1_6.0.12.1483_RealPlayer_RN30RD_en-us_686\r\n
    [truncated] Cookie: Edition=ED=us&CV=16TR=0&SET=user_session; realNav=|movies|; RNFeatures
    GUID: 00000000-0000-0000-0000-000000000000\r\n
    Language: en-us\r\n
    PlayerCookie: RNPlayer|RNFeatures|RNTS\r\n
    RegionData: 94063\r\n
    Require: com.real.retain-entity-for-setup\r\n
    SupportsMaximumASMBandwidth: 1\r\n
    \r\n
```

## rtsp-req-uri-path

Path of an RTSP request, not including the command line.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow. Qualifier: This context can use the RTSP method qualifier to limit signatures to specific RTSP methods.

```
▽ Real Time Streaming Protocol
  ▷ Request: DESCRIBE rtsp://media.real.com:554/showcase/ads/spicealicious_15.rm RTSP/1.0\r\n
    CSeq: 2\r\n
    User-Agent: RealMedia Player (HelixDNAClient)/10.0.0.0 (win32)\r\n
    Accept: application/sdp\r\n
    Session: 315291390-1
    Bandwidth: 1544000\r\n
    ClientID: WinNT_5.1_6.0.12.1483_RealPlayer_RN3ORD_en-us_686\r\n
    [truncated] Cookie: Edition=ED=us&CV=1&TR=0&SET=user_session; realNav=|movies|; RNFeatures
    GUID: 00000000-0000-0000-0000-000000000000\r\n
    Language: en-us\r\n
    PlayerCookie: RNPlayer|RNFeatures|RNTS\r\n
    RegionData: 94063\r\n
    Require: com.real.retain-entity-for-setup\r\n
    SupportsMaximumASMBandwidth: 1\r\n
    \r\n
```

## sip-req-headers

This field identifies the message header for a sip request.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted content.

```
▼ Message Header
  ▶ Via: SIP/2.0/UDP 172.168.208.164:3345;branch=z9hG4bKamCgoNQenQJ0EWGE;rport
  ▶ Contact: <sip:16177064456_scpnpnacv3PvsWLXBC4Iaq89PygN7hq70G4bzGhjYA1@172.168.208.164:3345;rinstance=
    Max-Forwards: 70
  ▶ From: <sip:16177064456_scpnpnacv3PvsWLXBC4Iaq89PygN7hq70G4bzGhjYA1@enflick.layered.net>;tag=5475E5E8
    Allow: OPTIONS, INVITE, ACK, REFER, CANCEL, BYE, NOTIFY
    Supported: replaces, path
    User-Agent: Enflick Softphone/8.4.1
  ▶ To: <sip:16177064456_scpnpnacv3PvsWLXBC4Iaq89PygN7hq70G4bzGhjYA1@enflick.layered.net>
    Expires: 600
    Call-ID: 0B804B9946C31DC17163858F0B69C6B6AB346020
    [Call-ID: 0B804B9946C31DC17163858F0B69C6B6AB346020]
```

## snmp-req-community-text

This context tracks the value of the variable field, “community” in the SNMP request header.

#### Additional Details

None

#### Context Capture

```
◢ User Datagram Protocol, Src Port: 1029 (1029), Dst Port: 161 (161)
    Source Port: 1029
    Destination Port: 161
    Length: 51
  ▷ Checksum: 0x6633 [validation disabled]
    [Stream index: 1]
◢ Simple Network Management Protocol
    version: v2c (1)
    community: sevenbyte
  ◢ data: get-request (0)
    ◢ get-request
        request-id: 0
        error-status: noError (0)
        error-index: 0
```

## smtp-req-argument

Argument of a SMTP command.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow. This context can use the SMTP method qualifier to limit signatures to specific SMTP methods.

```
▽ Simple Mail Transfer Protocol
  ▽ Command Line: HELO relay.example.org\r\n
      Command: HELO
      Request parameter: relay.example.org
```

## smtp-rsp-content

SMTP server response content.

#### Additional Details

None

#### Context Capture

This context provides the text highlighted in yellow.

```
▽ Simple Mail Transfer Protocol
  ▽ Response: 554 5.7.1 <Jason@pan.com>: Relay access denied\r\n
      Response code: Transaction failed (554)
      Response parameter: 5.7.1 <Jason@pan.com>: Relay access denied
```

## smtp-req-protocol-payload

The payload of an SMTP request.

### Context Capture

```
▼ Simple Mail Transfer Protocol
  ▼ Command Line: MAIL FROM:<owa[redacted].com>\r\n
      Command: MAIL
      Request parameter: FROM:<owa[redacted].com>
```

## smtp-rsp-protocol-payload

The payload of an SMTP response.

### Context Capture

```
▼ Simple Mail Transfer Protocol
  ▼ Response: 221 2.0.0 Service closing transmission channel\r\n
      Response code: <domain> Service closing transmission channel (221)
      Response parameter: 2.0.0 Service closing transmission channel
```

## ssh-req-banner

SSH banner of the client, not including comments.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ SSH Protocol
    Protocol: SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7\r\n
```

## ssh-rsp-banner

SSH banner of the server, not including comments.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ SSH Protocol
    Protocol: SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7\r\n
```

## ssl-req-certificate

Certificate request message of a SSL negotiation when initiated from the client.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Secure Sockets Layer
  ▽ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1105
    ▽ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 1101
        Certificates Length: 1098
      ▽ Certificates (1098 bytes)
          Certificate Length: 1095
        ▷ Certificate (pkcs-9-at-emailAddress=mwalter@paloaltonetworks.com,id-at-
  ▽ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
      Content Type: Handshake (22)
```

## ssl-req-chello-sni

Detects and identifies the SNI (Server Name Indication) contained within the client hello message of an SSL negotiation.

### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
Transport Layer Security
  TLSv1.3 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 512
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 508
      Version: TLS 1.2 (0x0303)
      Random: 85226f225c2366df08b4818c857104c9d61a691f9cd2e5a3c2fee55d2868db7e
      Session ID Length: 32
      Session ID: e44eda9e609b7aef6733fedb83b2439c73612aaa38aee49df1994d0cf3497b00
      Cipher Suites Length: 32
      Cipher Suites (16 suites)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 403
      Extension: Reserved (GREASE) (len=0)
      Extension: server_name (len=21)
        Type: server_name (0)
        Length: 21
        Server Name Indication extension
          Server Name list length: 19
          Server Name Type: host_name (0)
          Server Name length: 16
          Server Name: en.wikipedia.org
      Extension: extended_master_secret (len=0)
      Extension: renegotiation_info (len=1)
      Extension: supported_groups (len=10)
```

## ssl-req-client-hello

Client hello message of a SSL negotiation.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
Secure Sockets Layer
  TLSv1.1 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 188
    Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 184
      Version: TLS 1.1 (0x0302)
      Random
      Session ID Length: 0
      Cipher Suites Length: 72
      Cipher Suites (36 suites)
      Compression Methods Length: 1
      Compression Methods (1 method)
      Extensions Length: 71
      Extension: server_name
      Extension: renegotiation_info
      Extension: elliptic_curves
      Extension: ec_point_formats
      Extension: SessionTicket TLS
      Extension: next_protocol_negotiation
      Extension: Unknown 30031
```

## ssl-req-protocol-payload

The payload of an SSL request.

### Context Capture

```
▼ Transport Layer Security
  ▼ TLSv1 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 158
    ▶ Handshake Protocol: Client Hello
```

## ssl-req-random-bytes

Random bytes field in the SSL client hello.

### Additional Details

None

### Context Capture

This value is already hexadecimal; you'll need to write the pattern in your signature as such (enclosed in \x).

```
▽ Secure Sockets Layer
  ▽ SSL Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 173
    ▽ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 169
        Version: TLS 1.1 (0x0302)
      ▽ Random
          gmt_unix_time: Jun 20, 2013 17:37:23.000000000 PDT
          random_bytes: 68a5abf0e56aa8ebcad6c91b1102aba5cbc171b8661ec856...
        Session ID Length: 0
        Cipher Suites Length: 72
      ▷ Cipher Suites (36 suites)
```

## ssl-rsp-cert-subjectpublickey

Certificate subject public key that's part of an SSL server hello handshake.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
▽ Secure Sockets Layer
  ▽ TLSv1.1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.1 (0x0302)
      Length: 4284
    ▽ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 4280
        Certificates Length: 4277
      ▽ Certificates (4277 bytes)
          Certificate Length: 1520
        ▽ Certificate (id-at-commonName=www.mint.com,id-at-organizationalUnitName=IPS,id-
          ▽ signedCertificate
              version: v3 (2)
              serialNumber : 0x3ffa4514571de904b443fca78a473f81
            ▷ signature (shaWithRSAEncryption)
            ▷ issuer: rdnSequence (0)
            ▷ validity
            ▷ subject: rdnSequence (0)
            ▽ subjectPublicKeyInfo
              ▷ algorithm (rsaEncryption)
                Padding: 0
                subjectPublicKey: 3082010a0282010100cf47639ebb7f40c3497553b06cd30c...
```

## ssl-rsp-certificate

Certificate response message of a SSL negotiation from the server.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
▽ Secure Sockets Layer
  ▽ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 4304
    ▽ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 4300
        Certificates Length: 4297
      ▽ Certificates (4297 bytes)
          Certificate Length: 1540
        ▷ Certificate (id-at-commonName=wwws.mint.com,id-at-organizationalUnitName=Technology Operations
          Certificate Length: 1512
        ▷ Certificate (id-at-commonName=VeriSign Class 3 Extended Validation SSL CA,id-at-organizational
          Certificate Length: 1236
        ▷ Certificate (id-at-commonName=VeriSign Class 3 Public Primary Certification ,id-at-organizatio
  ▷ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

## ssl-rsp-protocol-payload

The payload of an SSL response.

### Context Capture

```
▼ Transport Layer Security
  ▼ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 53
    ▶ Handshake Protocol: Server Hello
  ▼ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 758
    ▶ Handshake Protocol: Certificate
  ▼ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 4
    ▶ Handshake Protocol: Server Hello Done
```

## ssl-rsp-server-hello

Server hello message of a SSL negotiation.

### Additional Details

None

### Context Capture

This context provides the text highlighted in yellow.

```
▽ Secure Sockets Layer
  ▽ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 74
    ▽ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: TLS 1.0 (0x0301)
      ▽ Random
          gmt_unix_time: Jan 14, 1991 01:24:51.000000000 PST
          random_bytes: c5709a69958a0383ef906d43e210fbc4544177289980474f...
        Session ID Length: 32
        Session ID: 61f94743978f78cd07946b7ee342ac72061bd74f4c838506...
        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
        Compression Method: null (0)
```

## tcp-context-free

The entire payload of a TCP packet.

### Additional Details

Available only on PAN-OS 10.0 or later.

- *Using this context is not recommended and results in severe performance degradation.*
- *Upon upgrade to PAN-OS 10.2 and later, PA-220 and PA-800 appliances support a maximum pattern length of 64 characters.*

**Context Capture**

```
    TCP payload (329 bytes)
▼ Hypertext Transfer Protocol
  ▶ POST /_search?pretty HTTP/1.1\r\n
    Host: ffutKcgCAQJs:9200\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)\r
    Accept: */*\r\n
    Content-Type: application/json; charset=UTF-8\r\n
    Connection: keep-alive\r\n
  ▶ Content-Length: 102\r\n
    \r\n
    [Full request URI: http://ffutKcgCAQJs:9200/_search?pretty]
    [HTTP request 1/1]
    File Data: 102 bytes
▶ JavaScript Object Notation: application/json
```

## telnet-req-client-data

All telnet data for traffic originating from the client.

**Additional Details**

None

**Context Capture**

This context matches the text highlighted in yellow.

```
▽ Telnet
    Data: show system info
```

## telnet-rsp-server-data

All telnet data for traffic originating from the server.

**Additional Details**

None

**Context Capture**

This context matches the text highlighted in yellow.

```
▽ Telnet
    Data:
    Data: hostname: PA-4060-2\r\n
    Data: ip-address: 10.2.133.14\r\n
    Data: netmask: 255.255.0.0\r\n
    Data: default-gateway: 10.2.0.1\r\n
    Data: ipv6-address: 2001:438:0:12:225:90ff:fe13:840/64\r\n
    Data: ipv6-link-local-address: fe80::225:90ff:fe13:840/64\r\n
    Data: ipv6-default-gateway: \r\n
```

## udp-context-free

The entire payload of a UDP packet.

### Additional Details

Available only on PAN-OS 10.0 or later.

- *Using this context is not recommended and results in severe performance degradation.*
- *Upon upgrade to PAN-OS 10.2 and later, PA-220 and PA-800 appliances support a maximum pattern length of 64 characters.*

### Context Capture

```
▶ User Datagram Protocol, Src Port: 1261, Dst Port: 5060
▼ Session Initiation Protocol (INVITE)
    ▶ Request-Line: INVITE sip:200@127.0.0.1 SIP/2.0
    ▶ Message Header
    ▶ Message Body
```

## unknown-req-tcp-payload

Full TCP payload for unknown TCP traffic originating from the client.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
▷ Transmission Control Protocol, Src Port: 19041 (19041) Dst Port: 55589 (55589), Seq: 99, Ack: 18, Len: 940
▽ Data (940 bytes)
    Data: 000d0a686f73746e616d653a2050412d343036302d320d0a...
    [Length: 940]
```

## unknown-rsp-tcp-payload

Full TCP payload for unknown TCP traffic originating from the server.

### Additional Details

None

#### Context Capture

This context matches the text highlighted in yellow.

```
▷ Transmission Control Protocol, Src Port: 19041 (19041) Dst Port: 55589 (55589), Seq: 99, Ack: 18, Len: 940
▽ Data (940 bytes)
    Data: 000d0a686f73746e616d653a2050412d343036302d320d0a...
    [Length: 940]
```

### unknown-req-udp-payload

Full UDP payload for unknown UDP traffic originating from the “client”, which is the initiator of UDP communications.

#### Additional Details

None

#### Context Capture

This context matches the text highlighted in yellow.

```
▷ User Datagram Protocol, Src Port: 31565 (31565), Dst Port: 29902 (29902)
▽ Data (91 bytes)
    Data: 5c93b4b444bdc29c9a1f216fd55df1063fd4df84fbc0425e...
```

### unknown-rsp-udp-payload

Full UDP payload for unknown UDP traffic originating from the “server”, which is opposite the “client”.

#### Additional Details

None

#### Context Capture

This context matches the text highlighted in yellow.

```
▷ User Datagram Protocol, Src Port: 18442 (18442), Dst Port: 24761 (24761)
▽ Data (160 bytes)
    Data: b2772f2d50ae822f0a5f8cbb0c76c4b9e076360d86eba921...
    [Length: 160]
```

## Integer Contexts

Integer Contexts are a type of custom signature context. They are used for equality operators: less than, greater than, and equal to. They are available for custom IPS signatures, but not custom application signatures.

- dnp3-req-func-code
- dnp3-req-object-type

- dns-rsp-tcp-over-dns
- dns-rsp-txt-found
- ftp-req-params-len
- http-req-connect-method
- http-req-content-length
- http-req-cookie-length
- http-req-header-length
- http-req-host-ipv4-address-found
- http-req-host-ipv6-address-found
- http-req-dst-port
- http-req-param-length
- http-req-no-host-header
- http-req-no-version-string-small-pkt
- http-req-simple-request
- http-req-uri-path-length
- http-req-uri-tilde-count-num
- http-rsp-code
- http-rsp-content-length
- http-rsp-total-headers-len
- iccp-req-func-code
- ike-req-payload-type
- ike-rsp-payload-type
- ike-req-payload-length
- ike-rsp-payload-length
- ike-version
- imap-req-cmd-param-len
- imap-req-first-param-len
- imap-req-param-len-from-second
- irc-req-protocol-payload
- irc-rsp-protocol-payload
- open-vpn-req-protocol-payload
- pfcp-req-msg-type
- pfcp-rsp-msg-type
- smtp-req-helo-argument-length
- smtp-req-mail-argument-length
- smtp-req-rcpt-argument-length

- sctp-req-ppid
- ssl-req-client-hello-ext-type
- ssl-req-client-hello-missing-sni
- ssl-rsp-version
- stun-req-attr-type
- panav-rsp-zip-compression-ratio

## dnp3-req-func-code

DNP3 Application Layer request and response headers contain function codes. The function codes include read, write, select, operate, and direct_operate. The dnp3-req-func-code context identifies these function codes which are 1 byte in length.

### Additional Details

None

### Context Capture

In this example, the function code 'Select' has hex value 0x03. In the custom application, a decimal equivalent of 3 will have to be defined.

```
▶ Frame 4: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
▶ Ethernet II, Src: Woonsang_04:05:06 (01:02:03:04:05:06), Dst: 06:05:04:03:02:01 (06:05:04:03:02:01)
▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▶ Transmission Control Protocol, Src Port: 64825 (64825), Dst Port: 20000 (20000), Seq: 2, Ack: 2, Len: 35
▼ Distributed Network Protocol 3.0
  ▶ Data Link Layer, Len: 26, From: 4, To: 3, DIR, PRM, Unconfirmed User Data
  ▶ Transport Control: 0xc1, Final, First(FIR, FIN, Sequence 1)
  ▶ Application data chunks
  ▼ Application Layer: (FIR, FIN, Sequence 1, Select)
    ▶ Application Control: 0xc1, First, Final(FIR, FIN, Sequence 1)
      Function Code: Select (0x03)
    ▼ SELECT Request Data Objects
      ▼ Object(s): Control Relay Output Block (Obj:12, Var:01) (0x0c01), 1 point
        ▶ Qualifier Field, Prefix: 2-Octet Indexing, Code: 16-bit Single Field Quantity
        ▶ Number of Items: 1
        ▶ Point Number 1 [Latch On] [NUL]
```

```
0000  c1 03 0c 01 28 01 00 01  00 03 01 64 00 00 00 64   ....(... ...d...d
0010  00 00 00 00                                        ....
```

## dnp3-req-object-type

This context can be used to identify group and variation objects in the DNP3 library.

### Additional Details

The dnp3-req-object-type context is a 2-byte hex value.

### Context Capture

In this case, the hex is 0x0c01 and the custom application will take a decimal value of 3073.

```
▶ Frame 4: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)
▶ Ethernet II, Src: PaloAlto_01:10:20 (00:1b:17:01:10:20), Dst: Dell_10:f8:f1 (00:1c:23:10:f8:f1)
▶ Internet Protocol Version 4, Src: 10.16.0.233, Dst: 10.1.4.8
▶ Transmission Control Protocol, Src Port: 10207 (10207), Dst Port: 20000 (20000), Seq: 2941955442, Ack: 1772930245, Len: 35
▼ Distributed Network Protocol 3.0
  ▶ Data Link Layer, Len: 26, From: 1, To: 1024, DIR, PRM, Unconfirmed User Data
  ▶ Transport Control: 0xce, Final, First(FIR, FIN, Sequence 14)
  ▶ Application data chunks
  ▼ Application Layer: (FIR, FIN, Sequence 10, Direct Operate)
    ▶ Application Control: 0xca, First, Final(FIR, FIN, Sequence 10)
      Function Code: Direct Operate (0x05)
    ▼ DIRECT OPERATE Request Data Objects
      ▼ Object(s): Control Relay Output Block (Obj:12, Var:01) (0x0c01), 1 point
        ▼ Qualifier Field, Prefix: 2-Octet Indexing, Code: 16-bit Single Field Quantity
            .010 .... = Index Prefix: 2-Octet Indexing (2)
            .... 1000 = Qualifier Code: 16-bit Single Field Quantity (8)
        ▶ Number of Items: 1
        ▶ Point Number 0 [Latch On] [NUL]

0000  ca 05 0c 01 28 01 00 00  00 03 01 e8 03 00 00 e8   ....(... ........
0010  03 00 00 00                                        ....
```

## dns-rsp-tcp-over-dns

Checks multiple conditions of a DNS response to detect TCP-over-DNS.

### Additional Details

If conditions indicating TCP-over-DNS are detected, the dns-rsp-tcp-over-dns field is set to 1.

### Context Capture

```
▶ User Datagram Protocol, Src Port: 53 (53), Dst Port: 65490 (65490)
▼ Domain Name System (response)
    [Request In: 1]
    [Time: 2.712467000 seconds]
    Transaction ID: 0x113c
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▼ Queries
    ▼ [truncated]6e440111ce3092b9a54d6963726f736f66742057696e646f7773205b5665.7273696f6e20362e312e373630315d0d0a436f70797269676874202863329.203230303920 4d6963726f
      Name: 6e440111ce3092b9a54d6963726f736f66742057696e646f7773205b5665.7273696f6e20362e312e373630315d0d0a436f70797269676874202863329.203230303920 4d6963726f736
      [Name Length: 225]
      [Label Count: 8]
      Type: TXT (Text strings) (16)
      Class: IN (0x0001)
  ▼ Answers
    ▶ [truncated]6e440111ce3092b9a54d6963726f736f66742057696e646f7773205b5665.7

0000  00 70 76 69 66 00 00 1b  17 00 01 11 08 00 45 00   .pvif... ......E.
0010  01 2e 00 00 40 00 40 11  23 ab 0a 0f 00 c0 0a 0c   ....@.@. #.......
0020  01 32 00 35 ff d2 01 1a  ba 29 11 3c 01 80 00 01   .2.5.... .).<....
0030  00 01 00 00 00 00 3c 36  65 34 34 30 31 31 31 63   ......<6 e440111c
0040  65 33 30 39 32 62 39 61  35 34 64 36 39 36 33 37   e3092b9a 54d69637
0050  32 36 66 37 33 36 66 36  36 37 34 32 30 35 37 36   26f736f6 67420576
0060  39 36 65 36 34 36 66 37  37 37 33 32 30 35 62 35   96e646f7 773205b5
0070  36 36 35 3c 37 32 37 33  36 39 36 66 36 65 32 30   665<7273 696f6e20
0080  33 36 32 65 33 31 32 65  33 37 33 36 33 30 33 31   362e312e 37363031

Wireshark · Follow UDP Stream (udp.stream eq 0) · dns-tf
.<.........<6e440111ce3092b9a54d6963726f736f66742057696e646f7773205b5665<72
73696f6e20362e312e373630315d0d0a436f70797269676874202863329<203230303920 4d696
3726f736f66742043 6f72706f726174696f6e2e2020.416c6c20726967687473.a
bovine1234.mooo.com......<.........<6e440111ce3092b9a54d6963726f736f66742057
696e646f7773205b5665<7273696f6e20362e312e373630315d0d0a436f70797269676874202
863329<203230303920 4d6963726f736f66742043 6f72706f726174696f6e2e2020.416c6c20
726967687473.a
bovine1234.mooo.com............Q}...b9fb0111ceb9a530ed

1 client pkt(s), 1 server pkt(s), 1 turn.
Entire conversation (517 bytes)   Show data as ASCII   Stream 0
```

## dns-rsp-txt-found

Checks the Answer section of a DNS response, and checks if the Type field is set to TXT.

### Additional Details

None

#### Context Capture

In this case, set the dns-rsp-text-found to 1 if TXT has to be identified as the DNS Type field.

```
▶ User Datagram Protocol, Src Port: 53 (53), Dst Port: 80 (80)
▼ Domain Name System (response)
    Transaction ID: 0x40c5
  ▶ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 2
    Additional RRs: 1
  ▶ Queries
  ▼ Answers
    ▼ txt403.gamesreality.com: type TXT, class IN
        Name: txt403.gamesreality.com
        Type: TXT (Text strings) (16)
        Class: IN (0x0001)
        Time to live: 2808
        Data length: 3909
```

## ftp-req-params-len

Length of the arguments to an FTP command, not including the command itself.

#### Additional Details

None

#### Context Capture

This context provides the length of the text highlighted.This context can use FTP command and FTP vendor ID qualifiers to limit signatures to specific FTP commands and known FTP clients.

```
▽ File Transfer Protocol (FTP)
  ▽ SIZE test12345\r\n
      Request command: SIZE
      Request arg: test12345
```

## http-req-connect-method

Identifies the connect method used for the http-request. If the connect method is used, then the value of this context is set to 1.

#### Additional Details

None

### Context Capture

This context provides the highlighted text.

```
▼ Hypertext Transfer Protocol
  ▶ [Expert Info (Warning/Security): Unencrypted HTTP protocol detected over encrypted port, could indicate a dangerous misconfiguration.]
  ▼ CONNECT www.[illegible] com:80 HTTP/1.1\r\n
    ▶ [Expert Info (Chat/Sequence): CONNECT www.[illegible] com:80 HTTP/1.1\r\n]
      Request Method: CONNECT
      Request URI: www.[illegible] com:80
      Request Version: HTTP/1.1
    Host: www.[illegible] com:80\r\n
    <Host: www.[illegible] com:80\r\n>
    User-Agent: [illegible]
    <User-Agent: [illegible]
    Proxy-Connection: keep-alive\r\n
```

## http-req-content-length

Content length of a HTTP request.

### Additional Details

None

### Context Capture

This context provides the integer highlighted in yellow.

```
▽ Hypertext Transfer Protocol
  ▽ POST /blog/wp-admin/admin-ajax.php HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): POST /blog/wp-admin/admin-ajax.php HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /blog/wp-admin/admin-ajax.php
      Request Version: HTTP/1.1
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
  ▽ Content-Length: 321\r\n
      [Content length: 321]
```

## http-req-cookie-length

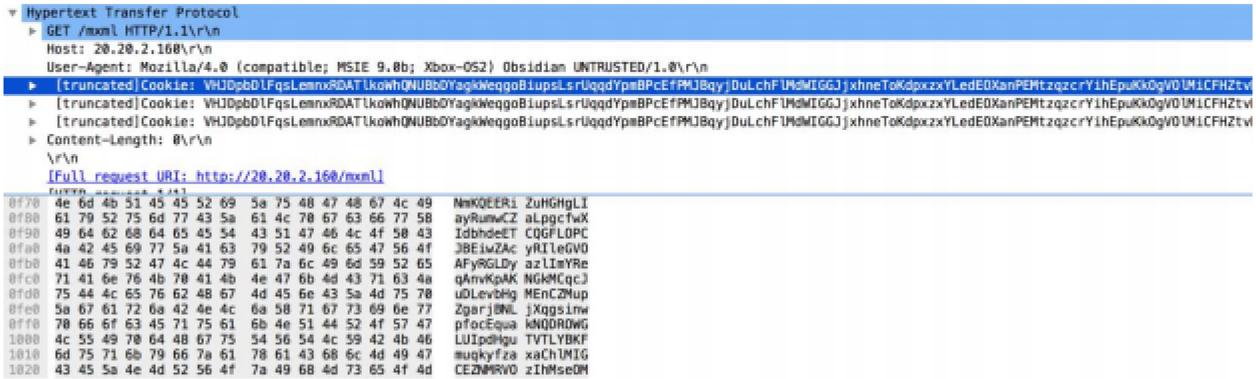
Identifies the Cookie header in an HTTP request header, and detects the number of bytes in the cookie string.

### Additional Details

None

### Context Capture



```
▼ Hypertext Transfer Protocol
  ▶ GET /mxml HTTP/1.1\r\n
    Host: 20.20.2.160\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 9.0b; Xbox-OS2) Obsidian UNTRUSTED/1.0\r\n
  ▶  [truncated]Cookie: VHJDpbDlFqsLemnxRDATlkoWhQNUBbDYagkWeqgoBiupsLsrUqqdYpmBPcEfPMJBqyjDuLchFlMdWIGGJjxhneToKdpxzxYLedEOXanPEMtzqzcrYihEpuKkOgVOlMiCFHZtv
  ▶  [truncated]Cookie: VHJDpbDlFqsLemnxRDATlkoWhQNUBbDYagkWeqgoBiupsLsrUqqdYpmBPcEfPMJBqyjDuLchFlMdWIGGJjxhneToKdpxzxYLedEOXanPEMtzqzcrYihEpuKkOgVOlMiCFHZtv
  ▶  [truncated]Cookie: VHJDpbDlFqsLemnxRDATlkoWhQNUBbDYagkWeqgoBiupsLsrUqqdYpmBPcEfPMJBqyjDuLchFlMdWIGGJjxhneToKdpxzxYLedEOXanPEMtzqzcrYihEpuKkOgVOlMiCFHZtv
  ▶ Content-Length: 0\r\n
    \r\n
    [Full request URI: http://20.20.2.160/mxml]
0f70  4e 6d 4b 51 45 45 52 69  5a 75 48 47 48 67 4c 49   NmKQEERi ZuHGHgLI
0f80  61 79 52 75 6d 77 43 5a  61 4c 70 67 63 66 77 58   ayRumwCZ aLpgcfwX
0f90  49 64 62 68 64 65 45 54  43 51 47 46 4c 4f 50 43   IdbhdeET CQGFLOPC
0fa0  4a 42 45 69 77 5a 41 63  79 52 49 6c 65 47 56 4f   JBEiwZAc yRIleGVO
0fb0  41 46 79 52 47 4c 44 79  61 7a 6c 49 6d 59 52 65   AFyRGLDy azlImYRe
0fc0  71 41 6e 76 4b 70 41 4b  4e 47 6b 4d 43 71 63 4a   qAnvKpAK NGkMCqcJ
0fd0  75 44 4c 65 76 62 48 67  4d 45 6e 43 5a 4d 75 70   uDLevbHg MEnCZMup
0fe0  5a 67 61 72 6a 42 4e 4c  6a 58 71 67 73 69 6e 77   ZgarjBNL jXqgsinw
0ff0  70 66 6f 63 45 71 75 61  6b 4e 51 44 52 4f 57 47   pfocEqua kNQDROWG
1000  4c 55 49 70 64 48 67 75  54 56 54 4c 59 42 4b 46   LUIpdHgu TVTLYBKF
1010  6d 75 71 6b 79 66 7a 61  78 61 43 68 6c 4d 49 47   muqkyfza xaChlMIG
1020  43 45 5a 4e 4d 52 56 4f  7a 49 68 4d 73 65 4f 4d   CEZNMRVO zIhMseOM
```

## http-req-dst-port

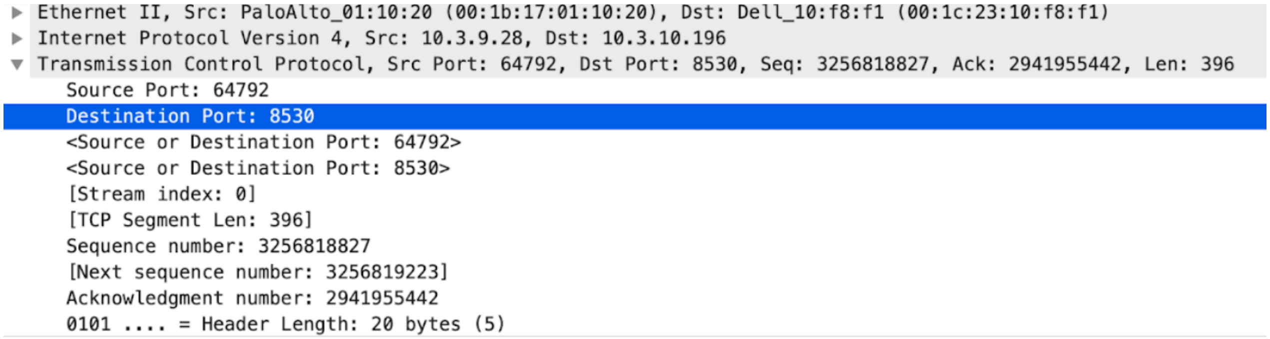
Identifies and detects the destination port for an HTTP request.

### Additional Details

None

### Context Capture

This context provides the highlighted text.



```
▶ Ethernet II, Src: PaloAlto_01:10:20 (00:1b:17:01:10:20), Dst: Dell_10:f8:f1 (00:1c:23:10:f8:f1)
▶ Internet Protocol Version 4, Src: 10.3.9.28, Dst: 10.3.10.196
▼ Transmission Control Protocol, Src Port: 64792, Dst Port: 8530, Seq: 3256818827, Ack: 2941955442, Len: 396
    Source Port: 64792
    Destination Port: 8530
    <Source or Destination Port: 64792>
    <Source or Destination Port: 8530>
    [Stream index: 0]
    [TCP Segment Len: 396]
    Sequence number: 3256818827
    [Next sequence number: 3256819223]
    Acknowledgment number: 2941955442
    0101 .... = Header Length: 20 bytes (5)
```

## http-req-header-length

Length of a HTTP request header, excluding method, path, and HTTP version.

### Additional Details

None

### Context Capture

Qualifiers: This context can use HTTP header field and HTTP method qualifiers to limit signatures to HTTP headers with specific values for select header fields and for specific HTTP methods.

```
▽ Hypertext Transfer Protocol
  ▷ POST /blog/wp-admin/admin-ajax.php HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
  ▷ Content-Length: 321\r\n
    Origin: http://thelegendofrandom.com\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko)
    Content-type: application/x-www-form-urlencoded\r\n
    Accept: */*\r\n
    Referer: http://thelegendofrandom.com/blog/\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f\r\n
```

## http-req-param-length

Length of the URL query string.

### Additional Details

None

### Context Capture

This context provides the length of the text highlighted in yellow (everything after the '?').

```
▽ Hypertext Transfer Protocol
  ▷ GET /blog/wp-admin/images/wordpress-logo.png?ver=20120216 HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Referer: http://thelegendofrandom.com/blog/wp-login.php\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f; wordpress_test_cookie=WP+Cookie+
    \r\n
    [Full request URI: http://thelegendofrandom.com/blog/wp-admin/images/wordpress-logo.png?ver=20120216]
```

## http-req-no-host-header

If this field is set to 1, an HTTP request with no host header has been found.

### Additional Details

None

### Context Capture

You can compare the topmost detected request to the normal request directly below it.

```
▼ Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    User-Agent: curl/7.47.0\r\n
    Accept: */*\r\n
    \r\n
```

You can compare the example above to a normal request:

```
▼ Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.example.com\r\n
    User-Agent: curl/7.47.0\r\n
    Accept: */*\r\n
    \r\n
```

## http-req-no-version-string-small-pkt

If this field is set to 1, an HTTP request that is less than 50 bytes and is missing the HTTP version string "HTTP/x.y" has been found.

### Additional Details

None

### Context Capture

You can compare the topmost detected request to the normal request directly below it.

```
▼ Hypertext Transfer Protocol
  ▶ GET /\r\n
    \r\n
```

You can compare the example above to a normal request:

```
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: 192.168.56.101\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Saf
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.8,ja;q=0.6\r\n
    \r\n
```

## http-req-simple-request

If this field is set to 1, an HTTP simple request missing the HTTP version string "HTTP/x.y" has been found.

### Additional Details

None

### Context Capture

You can compare the topmost detected request to the normal request directly below it.

```
▼ Hypertext Transfer Protocol
  > GET /?secret=somestring\r\n
    \r\n
```

You can compare the example above to a normal request:

```
▼ Hypertext Transfer Protocol
  ▶ GET / HTTP/1.1\r\n
    Host: 192.168.56.101\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.90 Saf
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.8,ja;q=0.6\r\n
    \r\n
```

## http-req-uri-path-length

Length of the URI path, not including the query string (up to and including the '?').

### Additional Details

None

### Context Capture

Qualifiers: This context can use the HTTP method to limit signatures in HTTP headers with specific HTTP methods.

```
▼ Hypertext Transfer Protocol
  ▷ GET /blog/wp-admin/images/wordpress-logo.png?ver=20120216 HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Referer: http://thelegendofrandom.com/blog/wp-login.php\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f; wordpress_test_cookie=WP+Cookie+
    \r\n
    [Full request URI: http://thelegendofrandom.com/blog/wp-admin/images/wordpress-logo.png?ver=20120216]
```

## http-req-uri-tilde-count-num

Number of "~" characters in the path (same path that http-req-uri-path provides).

### Additional Details

None

### Context Capture

The encoded characters below are included in this context.

Qualifiers: This context can use the HTTP method qualifier to limit signatures to HTTP headers with specific HTTP methods.

- %3A
- %u003A
- %u0589
- %u2236
- %u007E
- %u0303
- %u223C
- %uFF5E

## http-rsp-code

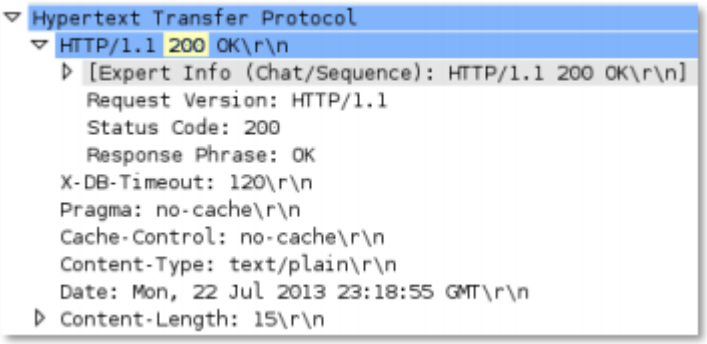
The number corresponding to the HTTP response code.

### Additional Details

None

### Context Capture

This context provides the integer highlighted in yellow.




## http-rsp-content-length

Content length of a HTTP response.

### Additional Details

None

### Context Capture

This context provides the integer highlighted in yellow.

```
▽ Hypertext Transfer Protocol
  ▽ HTTP/1.1 200 OK\r\n
    ▷ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Request Version: HTTP/1.1
      Status Code: 200
      Response Phrase: OK
    X-DB-Timeout: 120\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    Content-Type: text/plain\r\n
    Date: Mon, 22 Jul 2013 23:18:55 GMT\r\n
  ▷ Content-Length: 15\r\n
```

## http-rsp-total-headers-len

Length of the HTTP response headers, not including the HTTP status banner.

### Additional Details

None

### Context Capture

This context provides the content-length of the text highlighted in yellow.

```
▽ Hypertext Transfer Protocol
  ▽ HTTP/1.1 200 OK\r\n
    ▷ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Request Version: HTTP/1.1
      Status Code: 200
      Response Phrase: OK
    X-DB-Timeout: 120\r\n
    Pragma: no-cache\r\n
    Cache-Control: no-cache\r\n
    Content-Type: text/plain\r\n
    Date: Mon, 22 Jul 2013 23:18:55 GMT\r\n
  ▷ Content-Length: 15\r\n
```

## iccp-req-func-code

ICCP function codes such as read, write, identify, and rename can be identified using the iccp-req-func-code context.

### Additional Details

This context identifies the 1-byte function code value. In this case, the read function code has a hex value of 0xa4 and the corresponding decimal value is 164, which has to be entered while creating the custom application.

### Context Capture

```
▸ Frame 12: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
▸ Ethernet II, Src: PaloAlto_01:10:20 (00:1b:17:01:10:20), Dst: Dell_10:f8:f1 (00:1c:23:10:f8:f1)
▸ Internet Protocol Version 4, Src: 10.16.0.233, Dst: 10.1.4.8
▸ Transmission Control Protocol, Src Port: 63427 (63427), Dst Port: 102 (102), Seq: 3256819059, Ack: 2941955603, Len: 50
▸ TPKT, Version: 3, Length: 50
▸ ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
▸ ISO 8327-1 OSI Session Protocol
▸ ISO 8327-1 OSI Session Protocol
▸ ISO 8823 OSI Presentation Protocol
▾ MMS
  ▾ confirmed-RequestPDU
      invokeID: 1
    ▾ confirmedServiceRequest: read (4)
      ▾ read
        ▾ variableAccessSpecificatn: listOfVariable (0)
          ▸ listOfVariable: 1 item

0000  00 1c 23 10 f8 f1 00 1b  17 01 10 20 08 00 45 00   ..#..... ... ..E.
0010  00 5a df 2d 40 00 80 06  02 6f 0a 10 00 e9 0a 01   .Z.-@... .o......
0020  04 08 f7 c3 00 66 c2 1f  1d 73 af 5a ae 13 50 18   .....f.. .s.Z..P.
0030  ff ff b0 c7 00 00 03 00  00 32 02 f0 80 01 00 01   ........ .2......
0040  00 61 25 30 23 02 01 03  a0 1e a0 1c 02 01 01 a4   .a%0#... ........
0050  17 a1 15 a0 13 30 11 a0  0f 80 0d 54 41 53 45 32   .....0.. ...TASE2
0060  5f 56 65 72 73 69 6f 6e                            _Version
```

## ike-req-payload-type

Indicates the IKE payload type (identified by the *Next payload* entry) following the header in the requester's IKE message.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
⌄ Internet Security Association and Key Management Protocol
    Initiator SPI: 86973c9ad71c18ec
    Responder SPI: 0000000000000000
    Next payload: Security Association (1)
  > Version: 1.0
    Exchange type: Aggressive (4)
  > Flags: 0x00
    Message ID: 0x00000000
    Length: 805
  ⌄ Payload: Security Association (1)
      Next payload: Key Exchange (4)
      Reserved: 00
      Payload length: 556
      Domain of interpretation: IPSEC (1)
    > Situation: 00000001
    > Payload: Proposal (2) # 1
```

## ike-rsp-payload-type

Indicates the IKE payload type (identified by the *Next payload* entry) following the header in the responder's IKE message.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
Internet Security Association and Key Management Protocol
    Initiator SPI: 0000000000000000
    Responder SPI: 0000000000000000
    Next payload: Notification (11)
    Version: 1.0
    Exchange type: Informational (5)
    Flags: 0x00
    Message ID: 0xfb8c726e
    Length: 40
    Payload: Notification (11)
        Next payload: NONE / No Next Payload  (0)
        Reserved: 00
        Payload length: 12
        Domain of interpretation: ISAKMP (0)
```

## ike-req-payload-length

Indicates the length of a single payload entry contained inside of the requester's IKE packet, which itself may contain multiple payloads.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
    Length: 372
  ⌄ Payload: Security Association (1)
      Next payload: Key Exchange (4)
      Reserved: 00
      Payload length: 60
      Domain of interpretation: IPSEC (1)
    > Situation: 00000001
    > Payload: Proposal (2) # 1
  ⌄ Payload: Key Exchange (4)
      Next payload: Nonce (10)
      Reserved: 00
      Payload length: 132
      Key Exchange Data: a72d8a5c26eeab336cc303aba79f8b5a93fb18d0f591be81…
  > Payload: Nonce (10)
```

## ike-rsp-payload-length

Indicates the length of a single payload entry contained inside of the responder's IKE packet, which itself may contain multiple payloads.

### Additional Details

None

### Context Capture

This context matches the text highlighted in yellow.

```
⌄ Internet Security Association and Key Management Protocol
    Initiator SPI: 0000000000000000
    Responder SPI: 0000000000000000
    Next payload: Notification (11)
  > Version: 1.0
    Exchange type: Informational (5)
  > Flags: 0x00
    Message ID: 0xfb8c726e
    Length: 40
  ⌄ Payload: Notification (11)
      Next payload: NONE / No Next Payload  (0)
      Reserved: 00
      Payload length: 12
      Domain of interpretation: ISAKMP (0)
```

## ike-version

Indicates the version of the IKE protocol used in the exchange.

### Additional Details

None

#### Context Capture

This context matches the text highlighted in yellow.

```
▶ Frame 4: 226 bytes on wire (1808 bits), 226 bytes captured (1808 bits)
▶ Ethernet II, Src: Fortinet_3c:11:5d (08:5b:0e:3c:11:5d), Dst: PaloAlto_05:8e:10 (b4:0c:25:05:8e:10)
▶ Internet Protocol Version 6, Src: 2003:51:6012::4, Dst: 2003:51:6012::2
▶ User Datagram Protocol, Src Port: 500, Dst Port: 500
▼ Internet Security Association and Key Management Protocol
    Initiator SPI: 63c268011254b771
    Responder SPI: 850b3a1845f947cc
    Next payload: Key Exchange (4)
  ▶ Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▶ Flags: 0x00
    Message ID: 0x00000000
    Length: 164
  ▶ Payload: Key Exchange (4)
  ▶ Payload: Nonce (10)
```

## imap-req-cmd-param-len

Total length of all parameters of an IMAP command.

#### Additional Details

None

#### Context Capture

This context provides the length of the text highlighted in yellow.

Qualifiers: This context can use the IMAP command qualifier to limit signatures to specific IMAP commands.

```
▽ Internet Message Access Protocol
  ▽ Line: LOGIN root vpn123\r\n
      Request Tag: LOGIN
      Request: root vpn123
```

## imap-req-first-param-len

Length of the first parameter of an IMAP command.

#### Additional Details

None

#### Context Capture

This context provides the length of the text highlighted in yellow.

Qualifiers: This context can use the IMAP command qualifier to limit signatures to specific IMAP commands.

```
▽ Internet Message Access Protocol
  ▽ Line: LOGIN root vpn123\r\n
      Request Tag: LOGIN
      Request: root vpn123
```

## imap-req-param-len-from-second

Total length of all parameters of an IMAP command, not including the first.

#### Additional Details

None

#### Context Capture

This context provides the length of the text highlighted in yellow

This context can use the IMAP command qualifier to limit signatures to specific IMAP commands.

```
▽ Internet Message Access Protocol
  ▽ Line: LOGIN root vpn123\r\n
      Request Tag: LOGIN
      Request: root vpn123
```

## irc-req-protocol-payload

The payloads of the prefix, commands, and parameters of an IRC request. Does not cover the entire payload.

#### Context Capture

This context covers the sections called out in red.

```
elay Chat
: CAP LS
and: CAP
and parameters
arameter: LS
```

## irc-rsp-protocol-payload

The payloads of the prefix, commands, and parameters of an IRC response. Does not cover the entire payload.

#### Context Capture

This context covers the sections called out in red.

```
▼ Internet Relay Chat
  ▼ Response: [redacted].org/DWGS MODE #spunet +l 58
      Prefix: [redacted].org/DWGS
      Command: MODE
    ▼ Command parameters
        Parameter: #spunet
        Parameter: +l
        Parameter: 58
```

## open-vpn-req-protocol-payload

The payload of an OpenVPN request.

### Context Capture

```
▼ OpenVPN Protocol
  ▼ Type: 0x31 [opcode/key_id]
      0011 0... = Opcode: P_DATA_V1 (0x06)
      .... .001 = Key ID: 1
  ▼ Data (52 bytes)
      Data: ff8849511f2a6f998e26828502b215def7c7dd0df3c3119a…
```

## pfcp-req-msg-type

Indicates the PFCP message type value in the requester's Packet Forwarding Control Protocol (PFCP) message header.

### Additional Details

None

### Context Capture

This context matches the highlighted text.

```
> Internet Protocol Version 4, Src: 192.168.65.23, Dst: 192.168.165.14
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805
v Packet Forwarding Control Protocol
  > Flags: 0x20
    Message Type: PFCP Association Setup Request (5)
    Length: 21
    Sequence Number: 1
    Spare: 0
  > Node ID : IPv4 address: 192.168.65.23
  > Recovery Time Stamp : May 26, 2020 21:36:46.000000000 UTC
    [Response In: 2]
```

## pfcp-rsp-msg-type

Indicates the PFCP message type value in the responder's Packet Forwarding Control Protocol (PFCP) message header.

### Additional Details

None

#### Context Capture

This context matches the highlighted text.

```
> Internet Protocol Version 4, Src: 192.168.165.14, Dst: 192.168.65.23
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805
v Packet Forwarding Control Protocol
  > Flags: 0x20
    Message Type: PFCP Association Setup Response (6)
    Length: 32
    Sequence Number: 1
    Spare: 0
  > Node ID : IPv4 address: 192.168.165.14
  > Cause : Request accepted(success)
  > Recovery Time Stamp : May 26, 2020 21:36:46.000000000 UTC
  > UP Function Features :
    [Response To: 1]
    [Response Time: 0.002180000 seconds]
```

## smtp-req-helo-argument-length

Length of the argument to the SMTP “HELO” command.

#### Additional Details

None

#### Context Capture

This context provides the length of the text highlighted in yellow.

```
v Simple Mail Transfer Protocol
  v Command Line: HELO relay.example.org\r\n
      Command: HELO
      Request parameter: relay.example.org
```

## smtp-req-mail-argument-length

Length of the argument to the SMTP “MAIL FROM” command.

#### Additional Details

None

#### Context Capture

This context provides the length of the text highlighted in yellow.

```
v Simple Mail Transfer Protocol
  v Command Line: MAIL FROM: Martin@pan.com\r\n
      Command: MAIL
      Request parameter: FROM: Martin@pan.com
```

## smtp-req-rcpt-argument-length

Length of the argument to the SMTP “RCPT TO” command.

### Additional Details

None

### Context Capture

This context provides the length of the text highlighted in yellow.

```
▽ Simple Mail Transfer Protocol
  ▽ Command Line: RCPT TO: Jason@pan.com\r\n
      Command: RCPT
      Request parameter: TO: Jason@pan.com
```

## sctp-req-ppid

This context matches an SCTP Payload Protocol Identifier (PPID).

### Additional Details

A PPID is a 32-bit unsigned integer value that represents an application (upper layer) specified protocol identifier. It identifies the type of information being carried in a SCTP DATA chunk.

### Context Capture

```
▶ Frame 1: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)
▶ Ethernet II, Src: ACC_4a:00:35 (08:00:03:4a:00:35), Dst: TattileS_00:5e:46 (00:a0:80:00:5e:46)
▶ Internet Protocol Version 4, Src: 10.28.6.43, Dst: 10.28.6.44
▼ Stream Control Transmission Protocol, Src Port: 16384 (16384), Dst Port: 2944 (2944)
    Source port: 16384
    Destination port: 2944
    Verification tag: 0x00016f0a
    [Assocation index: 0]
    Checksum: 0x6db01882 (not verified)
  ▼ DATA chunk(ordered, complete segment, TSN: 671236933, SID: 0, SSN: 41149, PPID: 7, payload length: 75 bytes)
    ▶ Chunk type: DATA (0)
    ▶ Chunk flags: 0x03
      Chunk length: 91
    ▶ Transmission sequence number: 671236933
      Stream identifier: 0x0000
      Stream sequence number: 41149
      Payload protocol identifier: H.248/MEGACO (7)
      Chunk padding: 67
▶ MEGACO

0000  00 a0 80 00 5e 46 08 00  03 4a 00 35 08 00 45 00   ....^F.. .J.5..E.
0010  00 7c 14 1c 00 00 3b 84  4a 54 0a 1c 06 2b 0a 1c   .|....;. JT...+..
0020  06 2c 40 00 0b 80 00 01  6f 0a 6d b0 18 82 00 03   .,@..... o.m.....
0030  00 5b 28 02 43 45 00 00  a0 bd 00 00 00 07 4d 45   .[(.CE.. ......ME
```

## ssl-req-client-hello-ext-type

Detects the extension type listed in the TLS client hello message.

### Additional Details

None

#### Context Capture

This context provides the highlighted text, in this case, the encrypted Server Name extension present in the TLS Client Hello message. To detect this extension, specify ssl-req-client-hello-ext-type equals 65486.

```
▶ Compression Methods (1 method)
  Extensions Length: 594
▶ Extension: extended_master_secret (len=0)
▶ Extension: renegotiation_info (len=1)
▶ Extension: supported_groups (len=14)
▶ Extension: ec_point_formats (len=2)
▶ Extension: session_ticket (len=0)
▶ Extension: application_layer_protocol_negotiation (len=14)
▶ Extension: status_request (len=5)
▶ Extension: key_share (len=107)
▶ Extension: supported_versions (len=5)
▶ Extension: signature_algorithms (len=24)
▶ Extension: psk_key_exchange_modes (len=2)
▼ Extension: encrypted_server_name (len=366)
    Type: encrypted_server_name (65486)
    Length: 366
    Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
  ▶ Key Share Entry: Group: x25519, Key Exchange length: 32
    Record Digest Length: 32
    Record Digest: ca6763c1a0b429f6f7f9841a16cb119eb00b9562775b979625cb8a9bff3dc297
```

## ssl-req-client-hello-missing-sni

When this field is set to 1, an SSL client hello without the presence of an SNI (Server Name Indication) entry during the SSL negotiation process, is detected.

#### Additional Details

None

#### Context Capture

The following SSL client hello examples show requests with and without an SNI, respectively.

**With SNI Entry:**

```
Transport Layer Security
  TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 299
    Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 295
        Version: TLS 1.2 (0x0303)
        Random: c246dec20e37a3e9ab00b371bebf14ecb5299e4945ad4a2440c2ff22c64836d9
        Session ID Length: 32
        Session ID: bf6ccb295725ad6963d93960e78bb2845b12754fa7a4254655695861478f0aa4
        Cipher Suites Length: 62
      > Cipher Suites (31 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 160
      v Extension: server_name (len=17)
          Type: server_name (0)
          Length: 17
        > Server Name Indication extension
      > Extension: ec_point_formats (len=4)
      > Extension: supported_groups (len=12)
      > Extension: session_ticket (len=0)
      > Extension: encrypt_then_mac (len=0)
      > Extension: extended_master_secret (len=0)
      > Extension: signature_algorithms (len=42)
      > Extension: supported_versions (len=5)
```

### Without SNI Entry:

```
Transport Layer Security
  TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 278
    Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 274
        Version: TLS 1.2 (0x0303)
        Random: f78e38a07dcba75cdad98bebba1b8e019e3a406c8ed83a52b69f7ac9a607e5d4
        Session ID Length: 32
        Session ID: 6f6c90ce9a9cd4b58ef06ddae1338476fbca64c895fee288c6c03d777c01ec3b
        Cipher Suites Length: 62
      > Cipher Suites (31 suites)
        Compression Methods Length: 1
      > Compression Methods (1 method)
        Extensions Length: 139
      > Extension: ec_point_formats (len=4)
      > Extension: supported_groups (len=12)
      > Extension: session_ticket (len=0)
      > Extension: encrypt_then_mac (len=0)
      > Extension: extended_master_secret (len=0)
      > Extension: signature_algorithms (len=42)
      > Extension: supported_versions (len=5)
      > Extension: psk_key_exchange_modes (len=2)
      > Extension: key_share (len=38)
```

## ssl-rsp-version

Detects the SSL version listed in the SSL server hello handshake.

### Additional Details

None

**Context Capture**

```
▶ Frame 6: 200 bytes on wire (1600 bits), 200 bytes captured (1600 bits)
▶ Ethernet II, Src: CheckPoi_31:43:30 (00:a0:8e:31:43:30), Dst: CiscoInc_5d:18:1a (00:09:b6:5d:18:1a)
▶ Internet Protocol Version 4, Src: 134.154.194.20, Dst: 67.170.250.42
▶ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 3103 (3103), Seq: 1340392748, Ack: 3065106964, Len: 146
▼ Secure Sockets Layer
  ▼ SSLv3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 74
    ▶ Handshake Protocol: Server Hello
  ▶ SSLv3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  ▶ SSLv3 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

## stun-req-attr-type

This context identifies the 2-byte attribute type value in STUN server requests and responses.

### Additional Details

In this case, the hex is 0x0003 and the custom application will take a decimal equivalent value of 3.

**Context Capture**

```
▶ Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits)
▶ Ethernet II, Src: Dell_f4:e9:b8 (00:14:22:f4:e9:b8), Dst: Netscreen_18:1e:d5 (00:10:db:18:1e:d5)
▶ Internet Protocol Version 4, Src: 10.0.0.31, Dst: 64.0.27.226
▶ User Datagram Protocol, Src Port: 2566 (2566), Dst Port: 1376 (1376)
▼ Simple Traversal of UDP Through NAT
    Message Type: Binding Request (0x0001)
    Message Length: 0x0008
    Message Transaction ID: 0b590000432b00002954000028700000
  ▼ Attributes
    ▼ Attribute: CHANGE-REQUEST
        Attribute Type: CHANGE-REQUEST (0x0003)
        Attribute Length: 4
        .... .... .... .0.. = Change IP: Not set
        .... .... .... ..0. = Change Port: Not set

0000  00 10 db 18 1e d5 00 14  22 f4 e9 b8 08 00 45 00   ........ ".....E.
0010  00 38 3f 4d 00 00 80 11  95 67 0a 00 00 1f 40 00   .8?M.... .g....@.
0020  1b e2 0a 06 05 60 00 24  e9 e6 00 01 00 08 0b 59   .....`.$ .......Y
0030  00 00 43 2b 00 00 29 54  00 00 28 70 00 00 00 03   ..C+..)T ..(p....
0040  00 04 00 00 00 00                                  ......
```

## panav-rsp-zip-compression-ratio

This context detects the zip compression ratio of files downloaded over HTTP.

### Additional Details

The data compression ratio compares the uncompressed size and the compressed size of a file. This context can be used to identify a zip bomb or files with large data compression ratios.

**Context Capture**

```
6193 13.227947   192.168.55.11   192.168.45.35   TCP    66 80 → 45323 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
6194 13.228009   192.168.45.35   192.168.55.11   TCP    54 45323 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
6195 13.228139   192.168.45.35   192.168.55.11   HTTP  435 GET /zipbomb.zip HTTP/1.1
6196 13.263129   192.168.55.11   192.168.45.35   TCP  1514 [TCP segment of a reassembled PDU]
6197 13.290827   192.168.55.11   192.168.45.35   TCP  1514 [TCP segment of a reassembled PDU]
6198 13.290878   192.168.45.35   192.168.55.11   TCP    54 45323 → 80 [ACK] Seq=382 Ack=2921 Win=131328 Len=0
6199 13.317527   192.168.55.11   192.168.45.35   TCP  1514 [TCP segment of a reassembled PDU]
6200 13.346888   192.168.55.11   192.168.45.35   TCP  1514 [TCP segment of a reassembled PDU]

▷ Frame 6195: 435 bytes on wire (3480 bits), 435 bytes captured (3480 bits) on interface 0
▷ Ethernet II, Src: Vmware_a4:10:53 (00:50:56:a4:10:53), Dst: Vmware_e4:d7:22 (00:0c:29:e4:d7:22)
▷ Internet Protocol Version 4, Src: 192.168.45.35, Dst: 192.168.55.11
▷ Transmission Control Protocol, Src Port: 45323 (45323), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 381
◢ Hypertext Transfer Protocol
  ◢ GET /zipbomb.zip HTTP/1.1\r\n
    ▷ [Expert Info (Chat/Sequence): GET /zipbomb.zip HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /zipbomb.zip

0000  00 0c 29 e4 d7 22 00 50  56 a4 10 53 08 00 45 00   ..)..".P V..S..E.
0010  01 a5 5b 12 40 00 80 06  00 00 c0 a8 2d 23 c0 a8   ..[.@... ....-#..
0020  37 0b b1 0b 00 50 c5 cf  58 c9 07 d3 b5 f0 50 18   7....P.. X.....P.
0030  02 01 e7 16 00 00 47 45  54 20 2f 7a 69 70 62 6f   ......GE T /zipbo
0040  6d 62 2e 7a 69 70 20 48  54 54 50 2f 31 2e 31 0d   mb.zip H TTP/1.1.
0050  0a 48 6f 73 74 3a 20 31  39 32 2e 31 36 38 2e 35   .Host: 1 92.168.5
0060  35 2e 31 31 0d 0a 43 6f  6e 6e 65 63 74 69 6f 6e   5.11..Co nnection
0070  3a 20 6b 65 65 70 2d 61  6c 69 76 65 0d 0a 55 70   : keep-a live..Up
0080  67 72 61 64 65 2d 49 6e  73 65 63 75 72 65 2d 52   grade-In secure-R
0090  65 71 75 65 73 74 73 3a  20 31 0d 0a 55 73 65 72   equests:  1..User
00a0  2d 41 67 65 6e 74 3a 20  4d 6f 7a 69 6c 6c 61 2f   -Agent:  Mozilla/
```

# Context Qualifiers

Qualifiers lessen the chance of false positives by restricting the locations where the firewall can find a given pattern. In other words, a signature matches only when the firewall detects the pattern inside a specific qualifier, which corresponds to a specific context. For example, you might use the http-method qualifier to specify that a http-req-uri-path pattern matters when found inside a HTTP GET method.

```
◢ Hypertext Transfer Protocol
  ▷ GET /blog/wp-admin/images/wordpress-logo.png?ver=20120216 HTTP/1.1\r\n
    Host: thelegendofrandom.com\r\n
    Connection: keep-alive\r\n
    Accept: image/webp,*/*;q=0.8\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
    Referer: http://thelegendofrandom.com/blog/wp-login.php\r\n
    Accept-Encoding: gzip,deflate,sdch\r\n
    Accept-Language: en-US,en;q=0.8\r\n
    Cookie: slimstat_tracking_code=303072.6d7ea8bf09c9eb247db980c56004556f; wordpress_test_cookie=WP+Cookie+
    \r\n
    [Full request URI: http://thelegendofrandom.com/blog/wp-admin/images/wordpress-logo.png?ver=20120216]
```

FTP Command Qualifiers

FTP Vendor ID Qualifiers

HTTP Header Field Qualifiers

HTTP Method Qualifiers

IMAP Command Qualifiers

RTSP Method Qualifiers

SMTP Method Qualifiers

### FTP Command Qualifiers

FTP command qualifiers can be added to custom signatures that use FTP-related contexts to limit a match condition to specific FTP commands.

| | | | | | | |
|---|---|---|---|---|---|---|
| ABOR | ACCT | ALLO | APPE | AUTH | CDUP | CWD |
| DELE | EHLO | ERPT | HELO | LIST | MDTM | MKD |
| MODE | NLIST | OPTS | PASS | PASV | PBSZ | PORT |
| PWD | QUIT | REIN | REST | RETR | RMD | RNFR |
| RNTO | SITE | SIZE | SMNT | STAT | STOR | STOU |
| STRU | SYST | TEST | TYPE | UNKNOWN-COMMAND | UNLOCK | USER |
| XCRC | XMD5 | XSHA1 | | | | |

**FTP Vendor ID Qualifiers**

FTP vendor ID qualifiers can be added to custom signatures that use FTP-related contexts to limit a match condition to specific FTP clients.

| | | | | | |
|---|---|---|---|---|---|
| CEASERFTP | EASY_FILE_SHARING_FTP | FTP_CLIENTPA_FTP | FREEFTPD | MICROSOFTFTP | NETTERM |
| PROFTPD | SERV_U | UNKNOWN_FTP_SERVER | VSFTPD | WARFTPD | WS_FTP |
| WUFTP | | | | | |

**HTTP Header Field Qualifiers**

HTTP header field qualifiers can be added to custom signatures that use HTTP-related contexts to limit a match condition to HTTP headers that have specific values for select header fields.

| | | | | | |
|---|---|---|---|---|---|
| ACCEPT_LANGUAGE | AUTHORIZATION | CONTENT_ENCODING | CONTENT_LENGTH | CONTENT_TYPE | HOST |
| IF_MOD_SINCE | SUBSCRIBE_HDR | TRANSFER_ENCODING | UNKNOWN_HDR | X_FORWARD_FOR | |

**HTTP Method Qualifiers**

HTTP method qualifiers can be added to custom signatures that use HTTP-related contexts to limit a match condition to HTTP headers that use specific HTTP methods.

| | | | | | | |
|---|---|---|---|---|---|---|
| BCOPY | BDELETE | BITS_POST | BMOVE | BPROPFIND | BROPPATCH | CCM_POST |
| CONNECT | COPY | DELETE | GET | HEAD | LINK | LOCK |
| MCKCOL | MOVE | NOTIFY | OPTIONS | POLL | POST | PROPFIND |
| PROPPATCH | PROXY_SUCCESS | PUT | RPC_CONNECT | SEARCH | SMS_POST | SOURCE |

| | | | | | | |
|---|---|---|---|---|---|---|
| SUBSCRIBE | TRACE | TRACK | UNKNOWN_METHOD | UNLINK | UNLOCK | UNSUBSCRIBE |

**IMAP Command Qualifiers**

IMAP command qualifiers can be added to custom signatures that use IMAP-related contexts to limit a match condition to specific IMAP commands.

| | | | | | | |
|---|---|---|---|---|---|---|
| APPEND | AUTHENTICATE | CAPABILITY | CHECK | CLOSE | COPY | CREATE |
| DELETE | EXAMINE | EXPUNGE | FETCH | FIND | IDLE | LIST |
| LOGIN | LSUB | NOOP | RENAME | SEARCH | SELECT | STARTTLS |
| STATUS | SUBSCRIBE | UNKNOWN_COMMAND | UNSUBSCRIBE | | | |

**RTSP Method Qualifiers**

RTSP method qualifiers can be added to custom signatures that use RTSP-related contexts to limit a match condition to specific RTSP methods.

| | | | | |
|---|---|---|---|---|
| ANNOUNCES | DESCRIBE | GET_PARAMETER | OPTIONS | PAUSE |
| PLAY | RECORD | REDIRECT | SET_PARAMETER | SETUP |
| SETUP_PARAMETER | TEAR_DOWN | UNKNOWN_METHOD | | |

**SMTP Method Qualifiers**

SMTP method qualifiers can be added to custom signatures that use SMTP-related contexts to limit a match condition to specific SMTP methods.

| | | | | | | |
|---|---|---|---|---|---|---|
| AUTH | BDAT | DATA | EHLO | HELO | MAIL | QUIT |
| RCPT | RSET | SAML | SEND | SOML | STARTTLS | UNKNOWN_CMD |
| USER | VRFY | XEXCH50 | XEXPS | XLINK2STATE | XTELLMAIL | |

# IPS Signature Converter Plugin for Panorama

Snort and Suricata are open-source intrusion prevention system (IPS) tools that use uniquely formatted rules to detect threats. The IPS Signature Converter enables you to leverage these rules for immediate threat protection by translating the IPS signatures into custom Palo Alto Networks threat signatures. You can then register the signatures on Palo Alto Networks firewalls in specified device groups and enforce policy using Vulnerability Protection and Anti-Spyware Security Profiles.

Additionally, you can export rules that list IP address indicators of compromise (IOC) and use the resultant text file as an external dynamic list to enforce policy on the entries contained in the list.

# About the IPS Signature Converter Plugin

The IPS Signature Converter plugin for Panorama provides an automated solution for converting rules from a third-party intrusion prevention system (IPS)—Snort or Suricata—into custom Palo Alto Networks threat signatures. You can then register these custom signatures on firewalls that belong to device groups you specify and use the signatures to enforce policy in Vulnerability Protection and Anti-Spyware Security Profiles.

Snort and Suricata are open-source IPS tools that use uniquely formatted rules to detect threats. Organizations that share threat intelligence often distribute security advisories with these rules to help you implement the appropriate protections on your firewall. The IPS Signature Converter plugin enables you to immediately act upon these advisories and protect your network against any threats you receive in Snort or Suricata format.

After you install the IPS Signature Converter plugin on Panorama, you can upload rules for conversion and import them to your device groups. You can also export rules containing indicators of compromise (IOC) to a text file that you can use as an external dynamic list to enforce policy on the entries contained in the list.

# Convert Rules Using the Panorama Web Interface

After you install the intrusion prevention system (IPS) signature converter plugin, you can use it to translate Snort and Suricata rules into custom Palo Alto Networks threat signatures. You can then register the custom signatures on Palo Alto Networks firewalls that belong to device groups that you specify and use these customer signatures in your Vulnerability Protection and Anti-Spyware Security Profiles.

Additionally, you can export rules that list IP address indicators of compromise (IOC) and use the resultant text file as an external dynamic list to enforce policy on the entries contained in the list.

The following example uses this Snort rule:

```
alert tcp any any -> any any (msg:"Malformed_UA"; content:"User-
Agent: Mozillar/"; depth:500; sid:99999999;)
```

**STEP 1 |** Select **Panorama** > **IPS Signature Converter** > **Manage**.

**STEP 2 |** **Upload Signatures**.

**STEP 3 |** Select one of two methods for uploading your rules:

- **Browse** to and select a text file.

  *You cannot convert binary file types, such as .pdf or .docx.*

- Paste the rules directly into the text box.

  *You can upload only 300 rules at a time for conversion.*


**STEP 4 |** Click **OK**.

Your signatures will populate at least one of the following tabs: **Succeeded**, **Succeeded with Warnings**, **Failed**, **Duplicates**, or **Existing Coverage**.

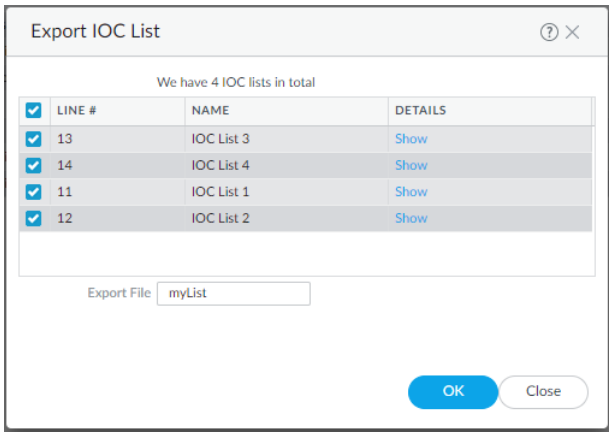
 

**STEP 5 |** (Optional) Export rules to an indicator of compromise (IOC) list.

Panorama converts a rule that does not contain the keywords `content` or PCRE into an **IOC List**. **Export IOC List** to group these rules into a text file that you can use as an external dynamic list for your Security policy rules.

1. Select **Export IOC List**.

   A dialog displays any rules that converted as **IOC List**.



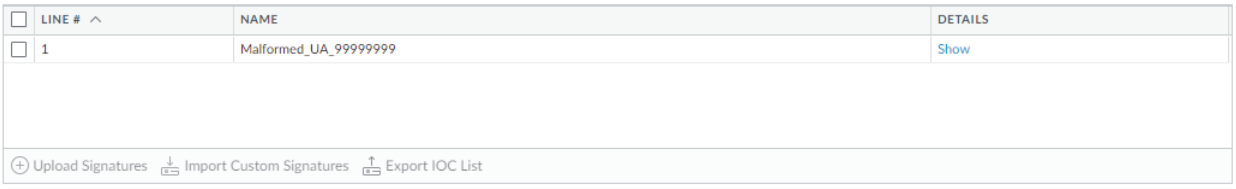

2. Select the rules that you want to export.
3. Enter the name of the file to which you want to export your rules.
4. Click **OK**.

   The exported text file will appear in your downloads folder.

**STEP 6 |** Commit converted signatures to Panorama.

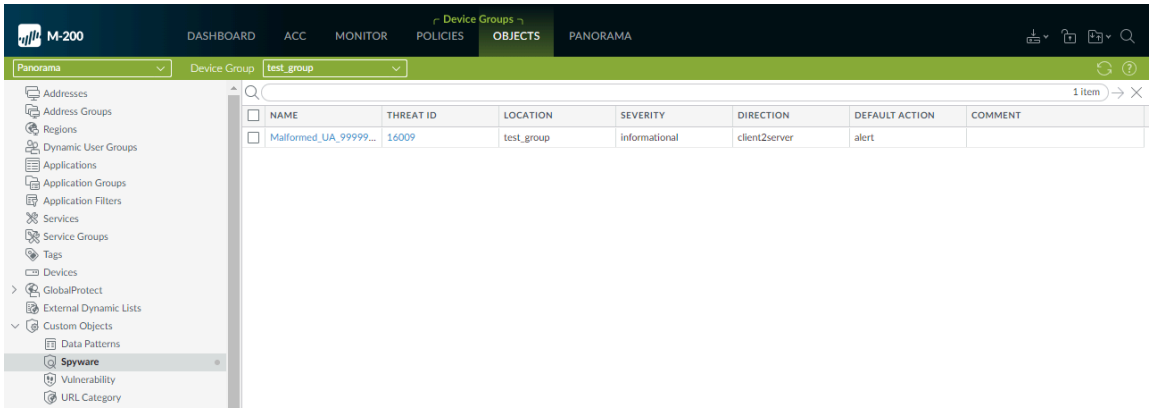
1. Select the signatures you want to upload.




2. **Import Custom Signatures**.
3. Select a Device Group from the drop-down.

   Select **Shared** to make the signatures available to all device groups.
4. Under the Destination column, select whether to commit the signatures as **Vulnerability** or **Spyware**.
5. Click **OK**.
6. In the top right of the screen, select and Commit to Panorama.
7. Verify that you successfully committed your signatures.

   1. Select **Objects** > **Custom Objects**.
   2. Select either **Spyware** or **Vulnerability**, depending on how you categorized your signatures in the previous step.



**STEP 7 |** Push the signatures to managed firewalls.

> *The firewalls must be running PAN-OS 10.0 or a later release with an active* Threat Prevention *license.*

**STEP 8 |** Test your signatures on a firewall in the device group to which you pushed the signatures.

# Convert Rules Using the Panorama CLI

In addition to the web interface, you can use the command-line interface (CLI) to convert Snort and Suricata rules into custom PAN-OS® threat signatures. This example uses the following Snort rule:

```
alert tcp $HOME_NET 2589 -> $EXTERNAL_NET any ( msg:"MALWARE-BACKDOOR
 - Dagger_1.4.0"; flow:to_client,established; content:"2|00 00 00
 06 00 00 00|Drives|24 00|",depth 16; metadata:ruleset community;
 classtype:misc-activity; sid:105; rev:14; )
```

You can not convert rule files through the CLI. If you want to convert a file with multiple rules in it, use the Panorama web interface.

The CLI Quick Start contains additional CLI commands.

**STEP 1 |** Encode the rule in Base64 format.

You can do this using a free, browser-based tool (example).

*Before encoding the rule, ensure there are no line breaks. Otherwise, the line breaks are encoded and cause the rule conversion in the subsequent step to fail.*

**STEP 2 |** Convert the encoded rule:

```
admin@demo-panorama-vm> request plugins ips-signature-converter
 convert b64-encode <base64_encoded_rule>

LINE# TITLE                    RESULT     TYPE        CONVERTER_MSG
1     Converted_MALWARE-       Succeed    Plain       None
      BACKDOOR -
      Dagger_1.4.0_105

Summary: Total:1, Succeed:1, Warnings:0, Existing Coverage:0,
 Duplicated:0, Failed:0
```

**STEP 3 |** (Optional) Change the signature type.

If your signature is for protection against spyware, you can set the type as **`spyware`** so that Panorama imports it as an Anti-Spyware signature. Otherwise, rules convert as `vulnerability` by default.

```
admin@demo-panorama-vm> request plugins ips-signature-converter
 set-properties signature-type <vulnerability/spyware>
 lines <line_number>

LINE# TITLE                              SIG_TYPE      ACTION
 SEVERITY
1     Converted_MALWARE-BACKDOOR -       spyware       alert
 low
      Dagger_1.4.0_105
```

**STEP 4 |** Import the signature to Panorama:

```
admin@demo-panorama-vm> request plugins ips-signature-
converter import-custom-signatures device-group <device_group>
 lines <line_number>

LINE# TITLE                           THREAT_ID STATUS  DETAIL
1     Converted_MALWARE-BACKDOOR -    16002     Success Import
 Succeeded
      Dagger_1.4.0_105
```

*If you do not specify a **device-group**, Panorama imports the signature to the Shared location.*

**STEP 5 |** Commit your changes to Panorama:

```
admin@demo-panorama-vm# commit

Commit job 707 is in progress. Use Ctrl+C to return to command
 prompt
...23%.59%80%.......90%.....100%
Configuration committed successfully
```

**STEP 6 |** Push the signatures to a device group:

```
@demo-panorama-vm> commit-all shared-policy device-
group <device_group>

Job enqueued with jobid 709
709
```

**STEP 7 |** Log in to a firewall in the device group that you specified in the previous step to verify that the push succeeded:

```
admin@PA-3220# show threats <vulnerability/spyware> <threat_id>
~
spyware {
  16002 {
    signature {
      standard {
        ips_converted_pattern {
          and-condition {
            "And Condition 1" {
              or-condition {
                "Or Condition 1" {
                  operator {
                    pattern-match {
                      pattern "2\x00 00 00 06 00 00 00\xDrives\x24
 00\x";
                      context tcp-context-free;
                      negate no;
```

```
                    }
                  }
                }
              }
            }
          }
          order-free no;
          scope session;
```

# Convert Rules Using the Panorama XML API

The Panorama XML API enables you to convert Snort and Suricata, open-source intrusion prevention system (IPS) rules to custom Palo Alto Networks threat signatures. You can then use the XML API to import the custom rules as Vulnerability Protection and Anti-Spyware Security profiles.

Because the PAN-OS® XML API uses a tree of XML nodes, you must specify the correct type and action in your API request along with the XPath Node Selection. See Explore the API to learn how to construct XML requests.

You can not convert rule files through the CLI. If you want to convert a file with multiple rules in it, use the Panorama web interface.

**STEP 1 |** Convert Snort or Suricata policy rules to Base64 URL encoded format.

You can use a free, browser-based tool (example.

This example uses the following Snort rule:

alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET CHAT Yahoo IM conference message"; flow: to_server,established; content:"YMSG"; nocase; depth: 4; content:"|00 1D|"; offset: 10; depth: 2; reference:url,doc.emergingthreats.net/2001258; classtype:policy-violation; sid:2001258; rev:7; metadata:created_at 2010_07_30, updated_at 2010_07_30;)

**STEP 2 |** Make a request to convert the rule to a custom PAN-OS threat signature.

```
curl -X POST 'https://
<firewall>/api/?key=
key&type=op&cmd=<request><plugins><ips-signature-
converter><convert><b64-encode>
YWxlcnQgdGNwICRIT01FX05FVCBhbnkgLT4gJEVYVEVSTkF
MX05FVCBhbnkgKG1zZzoiRVQgQ0hBVCBZYWh
vbyBJTSBjb25mZXJlbmNlIG1lc3NhZ2UiOyBmbG93OiB0b19zZXJ2ZXIs
ZXN0YWJsaXNoZWQ7IGNvbnRlbnQ6IllNU0ciOyBub2Nhc2U7IGRlcHRoO
iA0OyBjb250ZW50OiJ8MDAgMUR8Ijsgb2Zmc2V0OiAxMDsgZGVwdGg
6IDI7IHJlZmVyZW5jZTp1cmwsZG9jLmVtZXJnaW5ndGhyZWF0cy5uZXQvMjAwM
TI1ODsgY2xhc3N0eXBlOnBvbGljeS12aW9sYXRpb247IHNpZDoyMDAx
MjU4OyByZXY6NzsgbWV0YWRhdGE6Y3JlYXRlZF9hdCAyMDEwXzA3XzMwLCB1cGRh
```

```
        dGVkX2F0IDIwMTBfMDdfMzA7KQ==</b64-encode></convert></ips-
signature-converter></plugins></request>'
```

The response contains details about the rules (see previous details for more information):

```
        <response status="success">
            <result>
                <result>
                    <status>pass</status>
                    <msg>
                        <convert-result>
                            <extra-msg></extra-msg>
                            <failed-count>0/1</failed-count>
                            <failed></failed>
                            <duplicated-count>0/1</duplicated-
count>
                            <duplicated></duplicated>
                            <skipped-count>0/1</skipped-count>
                            <skipped></skipped>
                            <warned-count>1/1</warned-count>
                            <warned>
                                <entry name="1">
                                    <type>plain</type>
                                    <sig_type>vulnerability</
sig_type>
                                    <line>1</line>
                                    <title>Converted_ET CHAT Yahoo
 IM conference message_2001258</title>
```

```
                                    <action>alert</action>
                                    <severity>low</severity>
                                    <info>
                                        <entry name="0">

 <msg>[performance_impact] use of tcp-context-free (YMSG)</msg>
                                            <start_offset>127</
start_offset>
                                            <end_offset>131</
end_offset>
                                        </entry>
                                    </info>
                                </entry>
                            </warned>
                            <succeed-count>0/1</succeed-count>
                            <succeed></succeed>
                        </convert-result>
                    </msg>
                </result>
            </result>
        </response>
```

**STEP 3 |** Set the properties for rules that you converted.

Use the line number of a converted rule and set the properties. For example:

- Type set to **spyware**.
- Action when detected set to **alert**.
- Severity set to **low**.

```
curl -X POST 'https://
```

```
        <firewall>/api/?
type=op&key=LUFRPT0&cmd=<request><plugins><ips-signature-
converter><set-properties><default-action>alert</default-
action><lines>1</lines><severity>low</severity><signature-
type>spyware</signature-type></set-properties></ips-signature-
converter></plugins></request>'
```

The resulting success message:

```
<response status="success">
   <result>
       <result>
           <status>pass</status>
           <msg>
               <set-properties-result>
                   <entry name="1">
                       <line>1</line>
                       <sig_type>spyware</sig_type>
                       <action>alert</action>
                       <severity>low</severity>
                       <status>success</status>
                   </entry>
               </set-properties-result>
           </msg>
       </result>
   </result>
</response>
```

**STEP 4 |** (Optional) View the results of the converted rules.

The following request results in output that displays all successfully converted rules and the properties associated with each.

```
curl-X GET 'https://
<firewall>/api/?type=op&key=apikey&cmd=<show><plugins><ips-
signature-converter><results></results></ips-signature-converter></
plugins></show>
```

The resulting success message:

```
<response status="success">
    <result>
        <result>
            <status>pass</status>
            <msg>
                <line>1</line>
                <status>warned</status>
                <rule>alert tcp $HOME_NET any -
> $EXTERNAL_NET any (msg:"ET CHAT Yahoo IM conference
 message"; flow: to_server,established; content:"YMSG";
 nocase; depth: 4; content:"|00 1D|"; offset: 10; depth: 2;
 reference:url,doc.emergingthreats.net/2001258; classtype:policy-
violation; sid:2001258; rev:7; metadata:created_at 2010_07_30,
 updated_at 2010_07_30;)</rule>
                <type>plain</type>
                <sig_type>spyware</sig_type>
                <title>Converted_ET CHAT Yahoo IM conference
 message_2001258</title>
                <action>alert</action>
                <severity>low</severity>
```

```
                        <perf_score>10</perf_score>

                        <perf_level>high</perf_level>

                        <info>

                            <entry name="0">

                                <msg>[performance_impact] use of
 tcp-context-free (YMSG)</msg>

                                <start_offset>127</start_offset>

                                <end_offset>131</end_offset>

                            </entry>

                        </info>

                        <signatures>

                            <entry name="0">

                                <context>

                                    <!
[CDATA[<entry><signature><standard><entry
 name="ips_converted_pattern"><and-condition><entry name="And
 Condition 1"><or-condition><entry name="Or Condition
 1"><operator><pattern-match><pattern>YMSG</pattern><context>tcp-
context-free</context><negate>no</negate></pattern-match>

                        </operator>

                      </entry>

                    </or-condition>

                  </entry><entry name="And Condition 2"><or-
condition><entry name="Or Condition 1"><operator><pattern-
match><pattern>\x00 1D\x</pattern><context>tcp-context-free</
context><negate>no</negate></pattern-match>

                        </operator>

                      </entry>

                    </or-condition>

                  </entry>

                </and-condition><order-free>no</order-
free><scope>session</scope></entry>

            </standard>
```

```
          </signature><default-action><alert/></default-
action><reference><member>doc.emergingthreats.net/2001258</
member><member>Score: 10</member><member>Impact:
 high</member><member>Reason: use of tcp-context-free</
member></reference><threatname>Converted_ET CHAT Yahoo IM
 conference message_2001258</threatname><severity>low</
severity><direction>client2server</direction><affected-
host><server>yes</server></affected-host>
```

**STEP 5 |** Import the Spyware or Vulnerability rule to your device groups to use in a custom object.

Using the line number of a successfully converted rule, send a request that imports the rule to the shared device group.

```
       curl-X GET 'https://

        <firewall>/api/?key=

        key&type=op&cmd=<request><plugins><ips-signature-
converter><import-custom-sig><lines>

        1</lines></import-custom-sig></ips-signature-converter></
plugins></request>
```

The resulting success message using line one provides an ID number you can use to find the profile in the web interface.

```
        <response status="success">

        <result>

        <result>

        <status>pass</status>

        <msg>

        <import-result>

        <entry name="1">

        <line>1</line>

        <sid>42556</sid>

        <status>success</status>
```

```
        <msg>command succeeded</msg>
        </entry>
        </import-result>
        </msg>
        </result>
        </result>
        </response>
```

# Install the IPS Signature Converter Plugin

To convert intrusion prevention system (IPS) rules to custom PAN-OS® threat signatures, download and install the IPS Signature Converter on Panorama™.

If you have a Panorama high availability (HA) configuration, repeat the installation process on each Panorama peer. When you install the plugin on Panorama peers in an HA pair, install the plugin on the passive peer first and then on the active peer. After you install the plugin on the passive peer, that peer will transition to a non-functional state. When you install the plugin on the active peer, the passive peer will return to a functional state.

Before you install the plugin, ensure that you have the correct version requirements and the latest Applications and Threats content updates.

**STEP 1 |** Select **Panorama** > **Plugins**.

**STEP 2 |** Enter **`ips_signature_converter`** in the search bar.

**STEP 3 |** **Refresh** ( ) to retrieve the latest updates.

**STEP 4 |** **Download** (ACTIONS column) the plugin.

**STEP 5 |** Select the version of the plugin that you want to install.

**STEP 6 |** **Install** the plugin (ACTIONS column).

# CLI Quick Start

See the list of basic commands below for the intrusion prevention system (IPS) Signature Converter plugin on Panorama™. For more information about how to use the command line interface (CLI), see how to Get Started with the CLI.

| To do this... | Start here... |
| --- | --- |
| Convert, import, check performance impact, and configure the properties of signatures | `request plugins ips-signature-converter` |
| Set the type, default action, or severity of a signature | `request plugins ips-signature-converter set-properties <line numbers>` |
| View information about your converted signatures | `show plugins ips-signature-converter` |
| Delete all signatures (does not delete signatures that you imported to Panorama) | `clear plugins ips-signature-converter all` |

# Troubleshooting the IPS Signature Converter

If your rules fail to convert, use the following command in the Panorama command-line interface (CLI) to see a detailed summary of the failure:

```
admin@M-200-49> tail follow yes lines 1 plugins-log
 plugin_ips_signature_converter.log
```

The output consists of a list of the logs for each rule and a final summary of the status of their conversion.

**Rule Logs**

The output first lists the logs for each rule that you submitted for conversion. Each log contains the following fields.

| Field | Values |
|---|---|
| `Line` | The line number of the rule. |
| `result` | • `True`—The rule converted successfully.<br>• `False`—The rule failed to convert. |
| `type` | • `normal`—The rule contains a pattern to search packet payloads.<br>• `edl`—The rule is a list of suspect URLs, IP addresses, or domains. |
| `hash` | A unique identifier for each rule that successfully converted. This output is `None` if conversion failed. |
| `msg` | Details about a signature with a `result` of `failed` or `warned`. |

**Summary**

After listing the logs for each rule, the output displays a summary of the conversion results.

| Field | The number of rules that... |
|---|---|
| `Total` | Were submitted for conversion. |
| `Succeed` | Converted successfully. |

| Field | The number of rules that... |
| --- | --- |
| `Warned` | Converted successfully but contain minor syntax errors or that pose a risk, such as high performance impact or false-positive rate. |
| `Skipped` | Converted successfully and share a common vulnerabilities and exposures (CVE) identifier with a signature that already exists in the Palo Alto Networks Threat Vault. |
| `Duplicated` | Were repeated in the submission. |
| `Failed` | Failed to convert. |