TECHNICAL NOTE

STORMSHIELD NETWORK SECURITY

CUSTOM CONTEXT-BASED PROTECTION SIGNATURES

Product concerned: SNS 3.x, SNS 4.x

Document last updated: May 9, 2023

Reference: sns-en-custom_context_based_protection_signatures_technical_note

Table of contents

Before we begin 3

Requirements 4
- Checking whether Stormshield signatures are present and up to date 4
- Checking whether custom patterns have been excluded from Auto Update 4
- Enabling the use of custom patterns 4
- Defining the Active Update server 5
- Enabling the retrieval of custom patterns via Active Update 5

Structure of a custom pattern file 5
- Remarks and restrictions 5
- Contents of a context-based signature file 6

Defining custom patterns 9
- Defining a pattern that uses only mandatory fields 9
  - Meanings of the various fields in this example 9
- Examples of additional fields that can be added to this definition 10

Implementing custom patterns on the acceptance testing firewall 10
- Transferring the custom pattern file to the acceptance testing firewall 10
- Verifying the validity of the custom pattern definition file 10
- Compiling custom patterns 10
- Enabling custom patterns in the intrusion prevention engine 11

Testing custom patterns 11
- Checking for the presence of custom patterns in the web administration interface 11
- Testing custom patterns 11

Deploying patterns on the Active Update server 11

Downloading the custom pattern on client firewalls 12

Editing custom patterns 13
- Editing the CustomPatterns.in file 13
- Validating and compiling custom patterns 13
- On the Active Update server 13

Deleting custom patterns 14
- On the development workstation 14
- On the acceptance testing firewall 14
- On the Active Update server 14

Further reading 15

Before we begin

Custom context-based protection signatures (patterns) are to be analyzed by the firewall for applications developed in-house or in addition to signatures developed by Stormshield.

These patterns are based on regular expressions (known as "variants") that make it possible to locate character strings in the data contained in exchanged network packets. The associated alarms can then block or allow the traffic detected, depending on the settings defined in the custom pattern (which can be subsequently modified on each firewall in the **Configuration** > **Application protection** > **Applications and protections** module).

The example illustrated in this technical note consists of detecting the string "perdu.org" in a TCP or UDP request and automatically deploying this signature in a pool of firewalls. It involves four categories of equipment: a workstation for development, an acceptance testing firewall for custom context-based protection signatures, an Active Update server for the automatic distribution of signatures, and client firewalls.

Even though the file that defines custom signatures can be written directly on the acceptance testing firewall, one of the advantages of the development workstation is the availability of many tools for validating regular expressions, which can be found online or installed locally.

Further on in this document, custom context-based protection signatures will be referred to as custom patterns.

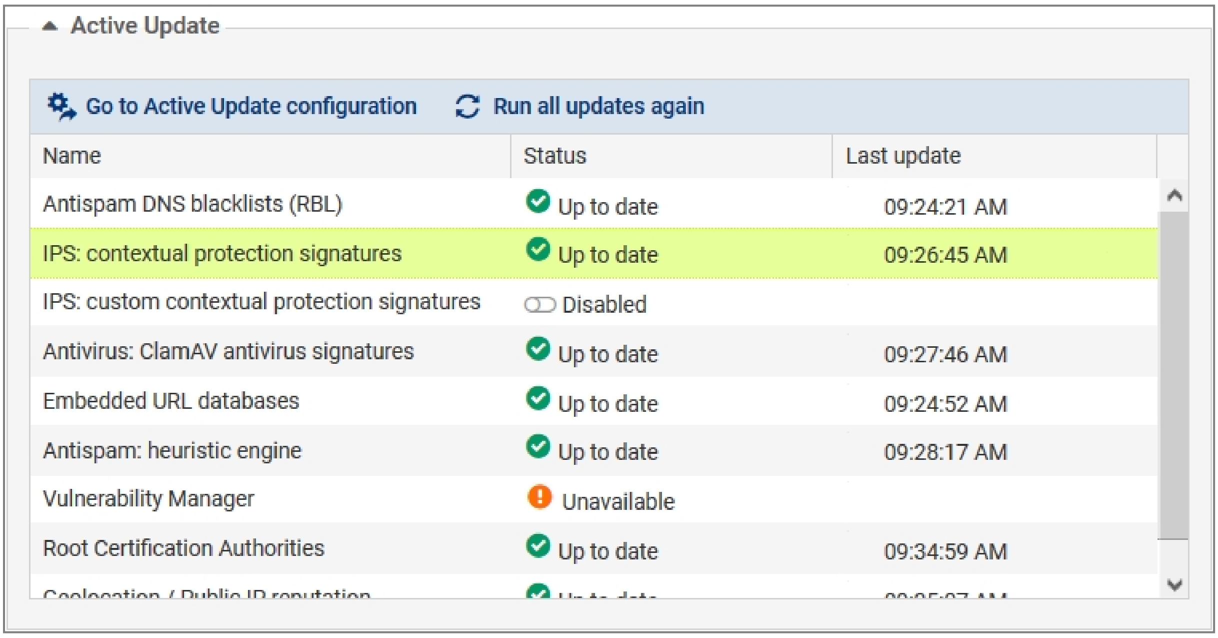
Take note that custom patterns may reveal information that is ordinarily hidden in the firewall's logs.

Requirements

Only on the acceptance testing firewall:

Checking whether Stormshield signatures are present and up to date

In the web administration interface of the acceptance testing firewall, using the **Dashboard**, check whether the Stormshield protection signatures have been downloaded (**Active Update** component):



Name	Status	Last update
Antispam DNS blacklists (RBL)	✓ Up to date	09:24:21 AM
IPS: contextual protection signatures	✓ Up to date	09:26:45 AM
IPS: custom contextual protection signatures	Disabled	
Antivirus: ClamAV antivirus signatures	✓ Up to date	09:27:46 AM
Embedded URL databases	✓ Up to date	09:24:52 AM
Antispam: heuristic engine	✓ Up to date	09:28:17 AM
Vulnerability Manager	! Unavailable	
Root Certification Authorities	✓ Up to date	09:34:59 AM
Geolocation / Public IP reputation	✓ Up to date	09:35:07 AM

If this is not the case, manually launch a download by clicking on **Run all updates again**.

Checking whether custom patterns have been excluded from Auto Update

In **Configuration** > **System** > **Active Update**, check that the line "IPS: protection signatures" has been disabled in order to prevent custom patterns that are being modified from being overwritten by those retrieved from the Active Update server.

On the acceptance testing firewall and on client firewalls:

Enabling the use of custom patterns

In the **Configuration** > **System** > **CLI** menu, enter both of the following commands:

```
CONFIG SECURITYINSPECTION COMMON INIT CustomPatternsMatching=1
CONFIG SECURITYINSPECTION ACTIVATE
```

On client firewalls

Defining the Active Update server

1. In **Configuration** > **System** > **Active Update**, expand the **Advanced properties** panel.
2. In the **Custom pattern update servers** table, click on **Add** and enter your server's URL (e.g.: http://my_active_update_server/ActiveUpdate/). You can indicate up to 8 custom pattern update websites.
3. Apply changes.

For further detail on the installation of a customized Active Update server, please refer to the Stormshield knowledge base. Do note that you will need to authenticate to access the knowledge base.

Enabling the retrieval of custom patterns via Active Update

In **Configuration** > **System** > **Active Update**, double-click on the status of the line "IPS: protection signatures" and confirm.

Structure of a custom pattern file

The characteristics of a custom pattern are:

- A context (e.g.: tcpudp:hostname, smtp:client, dcerpc:request:data, etc.), and
- A unique ID for a given context.

Remarks and restrictions

- Custom patterns are exclusively *asq* patterns (cf. section Contents of a context-based signature file). It is a simple policy that is meant to activate a security policy and raise the associated alarm.
- The *probe* and *mix* contexts as well as those beginning with *http:javascript* are not allowed. The command `enpattern -l | grep -Ev "(mix|probe)"` makes it possible to list usable contexts,
- Pattern IDs must always be higher than 4096,
- Any given context accepts a maximum of 2048 signatures,
- Patterns may not contain more than 256 regular expressions (*variants*),
- All contexts that group definitions of custom patterns are grouped in a single file (named *CustomPatterns.in* in the example).

> **IMPORTANT**
> Context-based signatures may consume a lot of processor resources and memory, especially when the regular expressions that they contain do not impose any limits on the number of characters for a given search.

Contents of a context-based signature file

The most basic structure of the file defining custom context-based signatures is as follows:

- A "[global.context]" section, that is unique for each context, in which the revision number of signatures is specified:

Field name	Description	Possible values (meaning)
Revision=	Revision number of signatures.	Full value. **Example** 1.2, etc.

- For every custom context-based pattern, there is a "[identifier.context]" section that contains all the following mandatory fields (the fields in the section can be in any order):

Field name	Description	Possible values (meaning)
type=	Specifies the pattern's scope of application	asq
classification=	Category of the pattern. In the web administration interface (**Applications and protections > Display by inspection profile** module), this field makes it possible to: • associate the right icon, and • filter patterns according to this value using the available buttons.	0 (Protections) 1 (Applications) 2 (Malware)
action_fw=	Action applied by the alarm associated with the custom pattern. This field is made up of 4 values, separated by commas, without spaces, corresponding to 4 predefined security templates: Internet, Low, Medium and High.	pass block **Example** pass,pass,pass,pass pass,pass,block,block
level_fw=	Level assigned to the associated alarm. This field is made up of 4 values, separated by commas, without spaces, corresponding to 4 predefined security templates: Internet, Low, Medium and High.	ignore minor major **Example** ignore,minor,major,major major,major,major,major
description=	Short description of the pattern written in English. It appears in **Message** column of the **Applications and protections** module.	Customized text surrounded by quotation marks. **Example** "Access to perdu.org site"

ldescr=	Additional information about the signature, expressed in English. It is displayed in a tool tip, when scrolling the mouse over the description of the alarm (**Message** column in the **Applications and protection** module).	Customized text surrounded by quotation marks. **Example** "This custom signature is able to detect when a computer tries to connect to the website perdu.org"
1=	First regular expression used in the pattern.	Regular expression surrounded by quotation marks

This section may also contain the following optional fields:

Field name	Description	Possible values (meaning)
severity=	Level of severity assigned to the threat detected by the custom pattern.	0 (Information) 1 (Low) 2 (Moderate) 3 (High) 4 (Critical)
resource=	This field allows assigning the icon of the relevant application to the pattern. This icon appears to the right of the classification icon.	Customized text **Example** Facebook Googleplus Twitter
description_fr=	Short description of the pattern written in French. It appears in **Message** column of the **Applications and protections** module.	Customized text surrounded by quotation marks. **Example** "Accès au site perdu.org"
ldescr_fr	Additional information about the signature, expressed in French. It is displayed in a tool tip, when scrolling the mouse over the description of the alarm (**Message** column in the **Applications and protection** module).	Customized text surrounded by quotation marks. **Example** "Cette signature personnalisée est capable de détecter lorsqu'un poste tente d'accéder au site perdu.org"
reference=	For custom pattens, this field is for reference only. In some cases, it completes the description of the pattern in the *CustomPatterns.in* file.	url,http://www.xxx.yz **Example** url,http://documentation.stormshield.eu
2= **3=** **4=** **etc.**	Additional regular expressions (*variants*). When several variants have been defined, their IDs must be consecutive. **Example of an invalid list of variants:** 1="blue" 3="red" 4="green" 6="yellow"	Regular expression surrounded by quotation marks

Fromasqversion=	Lowest version of SNS firmware needed in order to manage the pattern.	Version number. **Example** 1.0.0
Uptoasqversion=	Highest version of SNS firmware needed in order to manage the pattern.	Version number. **Example** 8.0.0

Defining custom patterns

Since the aim of this example is to detect the connection to the website http://*perdu.org* in a TCP or UDP request, the chosen context will be `tcpudp:hostname`. The value chosen for the pattern ID is 4101.

Defining a pattern that uses only mandatory fields

On the development workstation:

1. Create a file named *CustomPatterns.in,*
2. Edit this file and insert the section "[tcpudp:hostname.global]" containing the revision number of the pattern, followed by the section "[tcpudp:hostname.4101]" including the mandatory parameters:

```
[tcpudp:hostname.global]
Revision=1
[tcpudp:hostname.4101]
type=asq
classification=1
action_fw=pass,pass,block,block
level_fw=minor,minor,major,major
description="Access to perdu.org site"
ldescr="This custom signature is able to detect when a computer tries to
connect to the website perdu.org"
1="^(.+\.)?(?i)perdu\.org(?-i)$"
```

3. Insert as many "[identifier.context]" sections as the number of custom patterns you wish to define in the context in question (maximum 2048 patterns per context).

Meanings of the various fields in this example

Field **Revision**

The revision number of the custom patterns in the *tcpudp:hostname* context is 1.

Field **type**

The pattern must be an *asq* pattern: it is supposed to activate a security policy and raise an alarm.

Field **classification**

The pattern belongs to the *Applications* category.

Field **action_fw**

Whenever a connection to the website perdu.org is detected, the action associated with the activated alarm will be:

- *Pass* for "Internet" and "Low" predefined security templates,
- *Block* for "Medium" and "High" predefined security templates,

Field **level_fw**

This alarm's level is:

- *Minor* for "Internet" and "Low" predefined security templates,
- *Major* for "Medium" and "High" predefined security templates,

Field **description**

The message associated with the pattern and which appears in the web administration interface is "Access to perdu.org site".

Field **ldescr**

The tooltip that appears when you scroll over the message indicates: "This custom signature is able to detect when a computer tries to connect to the website perdu.org".

Field **1**

The regular expression used for detecting connections to perdu.org is:

```
^(.+\.)?(?i)perdu\.org(?-i)$
```

Examples of additional fields that can be added to this definition

```
severity=2
resource=perdu
reference=url,http://perdu.org
description_fr="Accès au site perdu.org"
ldescr_fr="Cette signature personnalisée détecte la tentative de connexion
d'une machine au site Web perdu.org"
Fromasqversion=1.0.0
Uptoasqversion=8.0.0
```

Implementing custom patterns on the acceptance testing firewall

Four steps are required to add custom signatures on a firewall:

Transferring the custom pattern file to the acceptance testing firewall

In SCP (in command line or using a WinSCP utility), copy the CustomPatterns.in file into the /usr/Firewall/ConfigFiles folder of the acceptance testing firewall.

Verifying the validity of the custom pattern definition file

Run the command:

```
enpattern -t /usr/Firewall/ConfigFiles/CustomPatterns.in
```

If the signature definition file is invalid, one or several messages will appear indicating the types of errors detected.

Compiling custom patterns

After having fixed any anomalies detected in the custom pattern definition file, run the command:

```
enpattern -fav
```

This command will launch the compilation of all patterns (options -f and -a). The option -v enables the command's verbose mode.

The folder /usr/Firewall/Data/CustomPatterns/Download will then contain one file per context, containing all patterns specific to this context (e.g.: *tcpudp_hostname*).

Enabling custom patterns in the intrusion prevention engine

On the acceptance testing firewall, run the command:

```
enasq
```

This command forces the intrusion prevention engine to take into account the custom patterns compiled earlier.

Testing custom patterns

The following steps are required in order to test custom patterns:

Checking for the presence of custom patterns in the web administration interface

1. In **Configuration** > **Application protection** > **Applications and protections**, display the **Type** column**.**
2. Click once on the title of this column to show custom patterns first:

Testing custom patterns

1. On a workstation that communicates through the acceptance testing firewall, generate traffic corresponding to the custom pattern:
2. In the dashboard of the acceptance testing firewall (**Protections** component), the alarm raised by this pattern must be present:
3. This alarm can also be viewed in the **Monitoring > Audit logs > Alarms** module.

Deploying patterns on the Active Update server

On the validation firewall, generate the archive containing all the custom patterns using the command:

```
enpattern -favz
```

This command will launch the compilation of all patterns (options -f and -a) and the creation of the archive that groups these signatures (option -z) and is meant to be provisioned on the Active Update server. The option -v enables the command's verbose mode.

The folder /usr/Firewall/Data/CustomPatterns/Download contains the output of this command:

- The archive named *custom_patterns_active_update.tgz,*
- A file per context, containing all patterns specific to this context (e.g.: *tcpudp_hostname*).

Transfer the archive *custom_patterns_active_update.tgz* to the root of the website hosted on your Active Update server, then unzip it.

This archive contains the following:

- A CustomPatterns-vX.index file that includes the list of custom patterns and their revision numbers,
- A file CustomPatterns-vX.md5 that allows verifying the integrity of the index file,
- A tree grouping the custom patterns.

The custom pattern is now ready to be deployed on the pool of client firewalls.

Downloading the custom pattern on client firewalls

Each client firewall receives the custom pattern:

- During an automatic synchronization with its Active Update server (scheduled every 3 hours),
- Via the **Active Update** component in the dashboard, by clicking on the **Restart** menu located next to the entry "IPS: protection signatures".

Custom patterns can be updated on a firewall as follows:

1. The firewall downloads the file CustomPatterns-vX.md5 on the Active Update server from the Active Update sub-system for custom patterns.
2. Whenever this file differs from its local .md5 file located in the folder /usr/Firewall/Data/CustomPatterns/Download/, the firewall downloads the file CustomPatterns-vX.index from the server in order to compare the revisions of each context:
    - If the context does not exist on the firewall or if the revision number of the context found on the server is higher than its local file's revision number, the firewall will download the server's context file, which will then be compiled and added to the firewall's patterns.
    - If the context file exists on the firewall but is not or no longer found on the Active Update server, this context file will then be deleted from the firewall. Patterns attached to this context will also be deleted from the firewall.
    - If the revision number of the context on the firewall is equal to or higher than the file's revision number on the Active Update server, the file will not be modified and the local version of the context will be applied on the firewall.

Editing custom patterns

The following steps are required in order to edit custom patterns:

Editing the *CustomPatterns.in* file

On the development workstation:

1. In the *CustomPatterns.in* file:

- Edit the section that defines the pattern,
- Increment the **Revision** field of the corresponding context.

2. Transfer this file to the acceptance testing firewall to replace the existing *CustomPatterns.in* file.

- or -

On the acceptance testing firewall, run the *CustomPatterns.in* file:

- Edit the section that defines the pattern,
- Increment the **Revision** field of the corresponding context.

Validating and compiling custom patterns

On the acceptance testing firewall:

1. Validate the custom pattern file:

```
enpattern -t /usr/Firewall/ConfigFiles/CustomPatterns.in.
```

2. Compile the signatures:

```
enpattern -fav
```

3. Enable the patterns in the intrusion prevention engine:

```
enasq
```

4. In **Application protection** > **Applications and protections**, ensure that this pattern is present.
5. Test the changes made to your custom pattern by using appropriate network traffic. Ensure that the alarm is raised as expected (**Dashboard** > **Alarms** widget).

On the Active Update server

Deploy the new custom pattern archive on your Active Update server.

Deleting custom patterns

The following steps are required in order to delete custom patterns:

On the development workstation

1. In the file *CustomPatterns.in*:

- Delete the section that defines the pattern,
- Increment the **Revision** field of the corresponding context.

2. Transfer this file to the acceptance testing firewall.

On the acceptance testing firewall

1. Delete the individual file corresponding to this context in the folder */usr/Firewall/Data/CustomPatterns/Download* (e.g.: tcpudp_hostname).
2. Validate the custom pattern file.
3. Launch the compilation of patterns.
4. In **Application protection** > **Applications and protections**, check whether this pattern has been deleted.

On the Active Update server

Deploy the new custom pattern archive on your Active Update server.

Further reading

Additional information and responses to questions you may have are available in the **Stormshield knowledge base** (authentication required).

documentation@stormshield.eu

All images in this document are for representational purposes only, actual products may differ.