APPENDIX B

Configuring Packet Analyzer Security

The Cisco Packet Analyzer software provides a number of security features with user-customizable parameters. This appendix provides an overview of the security features, and describes the parameters that can be customized.

This section consists of the following security features:

- Idle Timeout, page B-1
- SSL/TLS Security, page B-1
- SSH Security, page B-4

Idle Timeout

To prevent unauthorized access to the Packet Analyzer GUI or CLI, an idle/inactivity timeout is supported.

On the CLI, the idle timeout is disabled by default. An idle timeout can be configured using the following command:

```
cli idle-timeout <timeout-in-seconds>
```

The CLI idle timeout can be disabled using the `no cli idle-timeout` command.

SSL/TLS Security

The Packet Analyzer GUI supports HTTPS for secured connections. The HTTPS server can be enabled using the following command:

```
ip http secure server enable
```

Configuring a Self-Signed Certificate

Packet Analyzer is configured with a built-in self-signed certificate, by default. If you intend to continue using a self-signed certificate, we recommend that you generate a unique self-signed certificate using the following command:

```
ip http secure generate self-signed-certificate [lifetime-in-days]
```

The lifetime defaults to 730 days (2 years), but you can specify a different lifetime, if desired. This command will prompt you for organizational details that are customarily included in SSL/TLS certificates to help identify the server. You must ensure that the "Common Name" field matches whatever name you use to access your Packet Analyzer, as this is the field a web browser uses to verify that an SSL/TLS certificate properly matches the host that is presenting it.

- If you access the Packet Analyzer through an IP address (For example, https://10.0.0.10/, enter just the IP address "10.0.0.10").
- If you access the Packet Analyzer through a hostname (For example, https://secpa.cisco.com/, enter just the hostname "secpa.cisco.com").

The first time you connect to the Packet Analyzer using a particular web browser, it must show a warning that the certificate of Packet Analyzer is untrusted (because it is self-signed, rather than signed by a trusted Certificate Authority). You need to click the warning each time your browser is restarted, unless you save it to your local certificate store. The procedure for saving the certificate depends on your browser and/or operating system.

Note Ensure that the Common Name of your certificate is set correctly, before saving the certificate.

- Some browsers (such as Mozilla Firefox) maintain their own certificate store, and adding a new certificate is as simple as selecting "Add Exception", and then making sure that the "Permanently store this exception" option is checked.
- Other browsers (such as Microsoft Internet Explorer and Google Chrome) use the operating system's certificate store. On Windows, one possible procedure is:
  - Run the show certificate command from the Packet Analyzer CLI and copy the entire certificate text (including the "BEGIN CERTIFICATE" and "END CERTIFICATE" markers) into a .cer file (for example, "secpa-cert.cer").
  - Run the "certmgr.msc" program (you may have to enter the Win+R key sequence to access the Run menu). Right-click the "Trusted Root Certification Authorities" item, select All Tasks > Import to start the Certificate Import wizard, and then import the certificate file (for example, "secpa-cert.cer").
  - Your browser should now show the Packet Analyzer as a trusted host. In some cases, you may need to restart the browser to recognize the new certificate.
- For other browsers or operating systems, consult your local documentation.

Configuring a CA-Signed Certificate

For optimal security, it is recommended that the Packet Analyzer must be configured with a certificate signed by a trusted Certificate Authority (CA). This configuration will avoid the browser warning messages without the hassle of installing a self-signed certificate on each individual machine used to access the Packet Analyzer. Many larger enterprises have an in-house CA that can sign certificates for internal use. There are also external CAs that can sign a certificate, typically for a fee.

To configure the Packet Analyzer to use a CA-signed certificate, first issue the following CLI command to generate a certificate request, which will be output to the screen:

```
ip http secure generate certificate-request
```

Copy and paste the text of the certificate request and submit it to the CA for signing. Once the CA-signed certificate is received, use the following CLI command and paste the signed certificate text into the terminal window:

```
ip http secure install certificate
```

The Packet Analyzer will read the text and install the certificate.

Configuring SSL/TLS Parameters

The Packet Analyzer is configured for a balance between security and usability in the enterprise, by default. However, some users have particular requirements for the SSL/TLS ciphersuites and protocol versions that must be allowed by HTTPS servers on their networks, so the Packet Analyzer also offers the ability to customize the parameters.

Configuring SSL/TLS Ciphersuites

To configure the ciphersuites that the Packet Analyzer HTTPS server will accept, use this command:

```
ssl-tls ciphersuites set <ciphersuite-specification>
```

This command sets the SSLCipherSuite directive of the Apache web server built into the Packet Analyzer software. For details about the format of the ***ciphersuite-specification*** argument, refer to the following links:

http://httpd.apache.org/docs/2.4/mod/mod_ssl.html#sslciphersuite

http://www.openssl.org/docs/apps/ciphers.html

To evaluate a ***ciphersuite-specification*** argument using the Packet Analyzer's particular version of OpenSSL, use this command:

```
ssl-tls ciphersuites eval <ciphersuite-specification>
```

This command displays the list of ciphersuites that a given ***ciphersuite-specification*** represents.

To examine or verify the currently-configured ciphersuite specification, use this command:

```
show ssl-tls ciphersuites
```

Configuring SSL/TLS Protocols

To configure the SSL/TLS protocol versions that the Packet Analyzer HTTPS server will accept, use this command:

```
ssl-tls protocols set <protocol-directive>
```

This command sets the SSLProtocol directive of the Apache web server built into the Packet Analyzer software. For details about the format of the ***protocol-directive*** argument, refer to th following link:

http://httpd.apache.org/docs/2.4/mod/mod_ssl.html#sslprotocol

TLS v1.0, v.1.1, and v1.2 are enabled by default, while SSLv2 and SSLv3 are disabled. Note that you must not attempt to enable SSLv2 or SSLv3 protocols, as these versions of SSL have severe security flaws, and the industry as a whole is transitioning towards removing support for these protocols entirely.

For best security, enabling only TLS v1.2 is recommended (for example., **ssl-tls protocols set TLSv1.2**). However, some older browser versions do not enable support for the more recent versions of TLS, so you may have to visit an advanced settings dialog or similar to enable such TLS versions explicitly. Make sure that enabling the newer TLS versions may expose incompatibilities with other web servers (particularly older versions) that may be running in your environment.

SSH Security

The Packet Analyzer CLI supports SSH for secured connections. The SSH server can be enabled using the following command:

```
exsession on ssh
```

Configuring SSH Authorized Keys

In addition to logging into the CLI using password, the Packet Analyzer also supports logins using an SSH private key. To enable this functionality, a list of authorized keys must first be imported using the command:

```
ssh authorized-keys import <user> <key-file-url>
```

where ***<user>*** is "root" (for password-less CLI access) or a valid web username (for password-less access to capture or report files via SFTP). The format of this file is the standard OpenSSH authorized_keys file described in the "AUTHORIZED_KEYS FILE FORMAT" section in the following URL:

http://www.openbsd.org/cgi-bin/man.cgi?query=sshd&sektion=8

Note that SSH key options, if present, are removed when a key is imported.

The authorized keys for a given user can be displayed using the command:

```
show ssh authorized-keys <"checksums" | "file"> <user>
```

where **checksums** displays just the checksums (to facilitate comparison with your local copy of the file) and **file** displays the full contents.

Configuring SSH Ciphers and MACs

The Packet Analyzer is configured for a balance between security and usability in the enterprise, by default. However, some users have particular requirements for the ciphers and MACs that must be allowed by SSH servers on their networks, so the Packet Analyzer also offers the ability to customize the parameters.

To configure the ciphers that the Packet Analyzer SSH server will accept, use this command:

```
ssh ciphers set <ciphers-directive>
```

The ***ciphers-directive*** is simply a comma-separated list of the ciphers to be allowed, in order of preference (highest first).

The Packet Analyzer is configured to allow only SSH connections using AES CTR mode ciphers, by default. To examine or verify the currently-configured ciphers directive, or to see the list of available cipher options, use this command:

```
show ssh ciphers
```

To configure the MACs that the Packet Analyzer SSH server will accept, use the following analogous commands:. .

```
ssh macs set <macs-directive>
show ssh macs
```

The Packet Analyzer is configured to allow only SSH connections using HMAC-SHA1 and HMAC-RIPEMD160 MACs, by default.

Secure File Transfers

The Packet Analyzer has many commands that involve transferring files between the Packet Analyzer and external servers. A few examples of such commands are:

- Packet Analyzer software image upgrades: **upgrade** ***<image_url>***
- Packet Analyzer software patch installation: **patch** ***<patch_url>***
- Packet Analyzer configuration backup to network location: **config upload <url>**
- Packet Analyzer configuration restore from network location: **config network** ***<url>***

In all these cases, the URL provided is allowed to utilize an insecure plain text protocol like FTP or HTTP, if desired. However, for optimal security, using a secure protocol is recommended. The secure protocols supported by the Packet Analyzer are:

- SCP (Secure Copy) - relies on SSH for secure transport.
- SFTP (Secure File Transfer Protocol) - relies on SSH for secure transport.
- HTTPS (Hypertext Transfer Protocol Secure) - relies on SSL/TLS for secure transport.

Here are some examples of commands that perform file transfers:

- **upgrade https://files.cisco.com/upgrade_image.bin.gz**
- **patch scp://user:pass@ssh-server.cisco.com/patch.rpm**
- **config upload sftp:// user:pass@ssh-server.cisco.com/~/**

Protecting Against Man-in-the-Middle Attacks

A man-in-the-middle (MITM) attack is one in which a user unknowingly communicates with an impostor server, either because the impostor is positioned to intercept traffic en route to the legitimate server, or because the legitimate server is offline and the impostor has taken its place. To protect against such attacks, we recommend that the Packet Analyzer be configured such that it can verify that an external server it is communicating with is legitimate.

SSH Known Hosts

For SSH, the **known_hosts** file is a list of the public keys of SSH servers that the Packet Analyzer must consider "known" (legitimate). A known_hosts file can be imported using the command:

```
ssh known-hosts import  <known-hosts-file-url>
```

The format of this file is the standard OpenSSH known_hosts file described in the "SSH_KNOWN_HOSTS FILE FORMAT" section in the following URL:

http://www.openbsd.org/cgi-bin/man.cgi?query=sshd&sektion=8

After importing a known_hosts file, enable host key verification to ensure that only connections to known hosts are successful (connections to unknown hosts fail with an error message). Host key verification can be enabled using the command:

```
ssh host-key-verification
```

SSL/TLS CA Certificates

For SSL/TLS, the CA certificates file contains the list of Certificate Authority (CA) root certificates that the Packet Analyzer must trust. This file must contain PEM-formatted X.509 certificates (the format used by the cURL tool). An example of such a file can be found at the following URL:

http://curl.haxx.se/ca/cacert.pem

If you have HTTPS servers with self-signed certificates that the Packet Analyzer must consider trusted, simply include the self-signed certificate in this file.

A CA certificates file can be imported using the command:

```
ssl-tls ca-certs import  <ca-certs-file-url>
```

After importing CA certificates file, enable CA certificate verification to ensure that only connections to hosts with valid signed certificates are successful (connections to hosts with invalid certificates fail with an error message). The CA certificate verification can be enabled using the command:

```
ssl-tls cert-verification
```

Software Image Upgrades

For software image upgrades, the **upgrade** command displays MD5 and SHA-512 checksums for the downloaded image file prior to installation. The checksums are useful for ensuring that the image was not corrupted during download, but in the event that the image was downloaded to the Packet Analyzer without CA certificate or SSH host key verification enabled, the checksums can also be used to verify the authenticity of the image by comparing them to the checksums published on Cisco.com for the given software image.