Teldat SiemSensor

Quick Guide

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1 About This Guide

This Quick Guide focuses on the SiemSensor application for the Teldat Atlas i6x.

1.1 Supported Devices

The information contained in this guide only applies to the Atlas i6x equipped with an internal storage device and the SiemSensor application.

1.2 Who should read this manual?

This manual should be read by users who need to configure SiemSensor in an Atlas i6x.

1.3 When should I read this manual?

Read this guide as soon as you are ready to configure your SiemSensor application. This manual includes examples of different scenarios, where the SiemSensor application is useful, and explains how to configure the different parameters.

1.4 What is in this manual?

This Quick Guide contains the following information:

- A brief comment on the purpose and operation of the SiemSensor application.
- How to configure the application using the internal web of the Atlas i6x.
- A description of some general scenarios where the SiemSensor could be used.
- Troubleshooting.

1.5 What is not in this manual?

This quick guide does not contain information on Atlas i6x hardware. Also, its purpose is not to describe all management operations available in the Management Platform, the Atlas i6x Application Host software and configuration, or any other application different from SiemSensor. It does not contain information on how to setup the device to connect to the Internet. For information on how to configure the device, please see the relevant manuals for the different protocols, which can be found on the following web site: *www.teldat.com*.

1.6 How is the information organized?

Chapter 1 explains how to use this guide and describes its contents. Chapter 2 introduces the SiemSensor application and offers a brief explanation regarding its components and the intrusion detection rules. Chapter 3 focuses on the different configuration methods for this application. Chapter 4 presents several scenarios where the SiemSensor could be used.

In addition, this document includes an Appendix that provides additional information related to the configuration of certain aspects of the SiemSensor application and a Troubleshooting section at the end.

1.7 Technical Support

Note

The manufacturer reserves the right to make changes and improvements in the appropriate features in either software or hardware of this product, modifying the specifications of this manual without prior notice. The screen captures provided throughout the guide are provided as information guidelines only. Some small modifications may exist in the current software.

1.8 About OpenSource Software

Some software components of this product contain copyrighted software that is licensed under the GPL, GFDL, LGPL and other open source licenses. Within three years of our last product shipment, you may download for free the complete corresponding source code from *http://irouter.integra-te.com/zeus_id/* . If you want the full corresponding source code to be stored in a physical medium (such as a CD-ROM), we may charge a cost for having to physically perform source distribution. This offer is valid to anyone in receipt of this information.

For more information about the licenses of the software installed in the Application Host of an Atlas i6x, please refer to the *About section* of the device's web configurator.

Chapter 2 What is the SiemSensor application?

2.1 Teldat SiemSensor application

Teldat's SiemSensor application is a SIEM (Security Information and Event Management) probe that runs an IDS (Intrusion Detection System) software to analyze traffic received by the Atlas i6x and generate alert events when a packet triggers an alarm (based on a set of rules provided). These alerts are sent to a server, which also receives alerts coming from other Atlas i6x SiemSensor applications and agents. After analyzing all the information received, the server takes action.

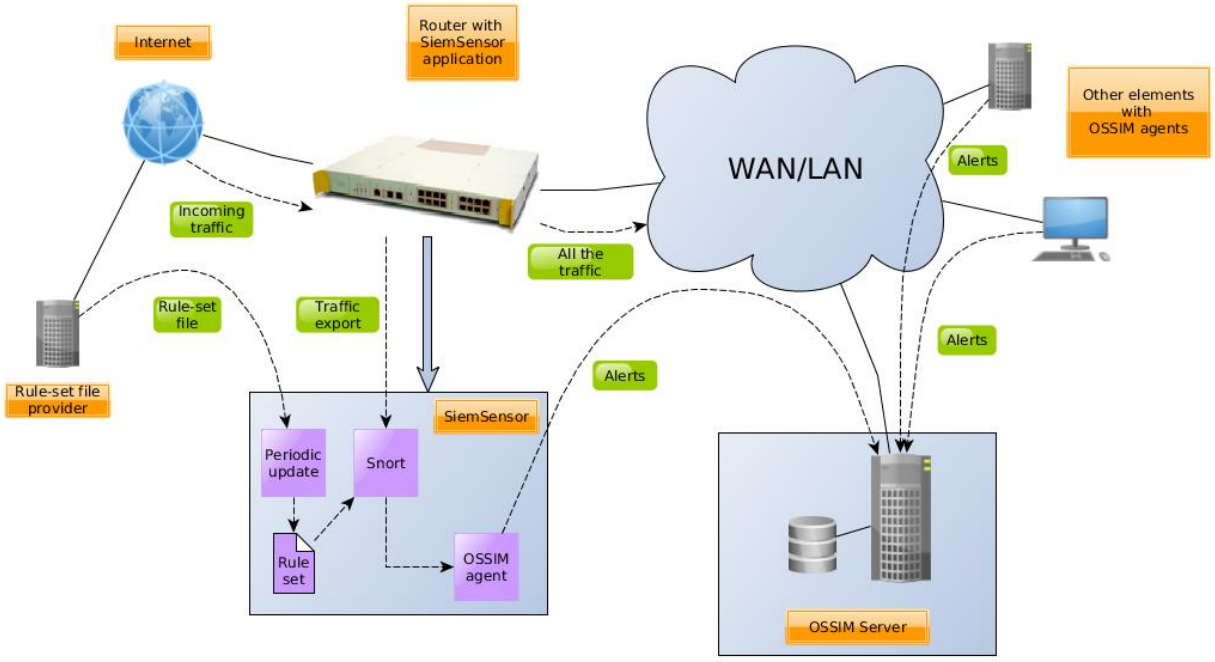
SIEM is a term used to describe a software that combines security information management and security event management. A SIEM system usually manages log security data, provides analysis of security alerts and generates reports.

Relevant security data is not only collected in one place, but in multiple locations. Looking at all the data from a single point of view helps to analyze the collected information and discover atypical situations. SIEM systems work by deploying multiple hierarchically-structured agents to gather security events from all kinds of devices (end-user devices, servers, network and security equipment, etc.).

2.2 Components

The SiemSensor application comprises an OSSIM agent and Snort software that manages the IDS (Intrusion Detection Service) functions. All the packets received by the device are analyzed by the Snort software. To ensure the Snort software has access to the traffic, the Atlas i6x CIT must be configured to export traffic (see Appendix *CIT Configuration* on page 15). Traffic is copied to the application's core so the Snort can analyze it, while the original traffic is routed by the CIT. Snort software inspects the packet and looks for any matches to the rules provided. If there is a match, then an event is generated and the OSSIM agent sends it to an OSSIM server. The OSSIM server evaluates the threat of the received event (possibly taking into account its relationship with other received events) and takes appropriate action. The following image shows a diagram of the SiemSensor operation and components. The network design presented in the diagram is just an example and may change. It is also possible that the OSSIM server is not on the WAN/LAN network but on the Internet. Also, the rule-set provider can be the OSSIM server or a machine on the WAN/LAN network. *OSSIM server infrastructure is not provided by Teldat* .

Fig. 2.1. SiemSensor operation diagram



OSSIM (Open Source Security Information Management) is an open source SIEM. By combining log management, asset management and discovery with information from dedicated information security controls and detection systems, OSSIM provides a view of all the security-related aspect of the system.

The OSSIM agent needs an OSSIM server to collect the information sent by the SiemSensor. Further information on OSSIM can be found in the following link:

http://www.alienvault.com/open-threat-exchange .

Snort is an open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS finger-printing attempts, etc. The SiemSensor application allows several parameters related to the IDS function to be configured.Further information about Snort can be found in the following link: *https://www.snort.org/* .

2.3 Rule set selection

As it has been previously said, the SiemSensor application uses rules to detect anomalous packets. These are included in a rule set file (snort.rules) that must be downloaded from an available URL. The file must be periodically downloaded to keep the rule set updated.

Selecting the applicable rules is subjective and must be done by the client. It depends on multiple factors and *each client should design his own set to suit his business and security requirements*.

Snort (*https://www.snort.org/*) provides rules classified by their functionality. Subscribers receive Snort Suscriber Rule Set updates immediately on release. Alternatively, users can register in the Snort web and receive exactly the same rule set for free, thirty days after said release. A community rule set can also be downloaded without registering. Snort community rules are found in the following link:

https://www.snort.org/downloads/#rule-downloads .

The rules distributed by Snort usually come in a tarball with a no snort.rules file. Here, there are different configuration files (not necessary for updating the rules) and a rules directory. The rules are distributed in several files that contain rules associated to a certain issue, i.e. sql.rules, telnet.rules, etc. You have to create your own combination of snort.rules, using the ones present in the different rules files. This customized snort.rules file is the one that must be published in a URL the SiemSensor application can access.

We recommend keeping a record of the rules files used and to compare the old and new files to look for new rules or the latest version of the old ones. Subsequently, replace the old snort.rules file with the new version and publish it so it can be downloaded by the SiemSensor application.

Some companies, like AlienVault (OSSIM developer), offer sets of rules updated with the latest detected threats.

2.4 Snort rules design

Snort uses a simple, lightweight language, which is both flexible and powerful, to describe rules. Snort rules are divided into two logical sections: the rule header and the rule options. The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, as well as information on source and destination ports. The rule option section contains alert messages and information regarding the parts of the packet that should be inspected to assess if further action is needed. Here is an example of a rule:

```
alert tcp any any -> any 80 (msg: "HTTP Traffic";
 content: "MassiveDynamics"; nocase; sid:4000151; rev:1;)
```

The text up to the first parenthesis is the rule header and the section enclosed in parenthesis contains the rule options. The words before the colons in the rule options are known as option keywords.

The basic rule header is made up of the following information: Action Protocol SourceIP SourcePort Direction DestinationIP DestinationPort (rule options).

The sample rule uses the alert action when it detects TCP traffic from any source IP on any port to any destination IP on port 80 containing the case insensitive string MassiveDynamics. The message HTTP Traffic is written to our logs when the rule is triggered. The sid option is an identifier number for the rule that must not be repeated (choose a number upper 4000000 to avoid the commonly used rule numbers) and the rev option indicates the revision of the rule.

Snort uses PCRE (Perl Compatible Regular Expressions) to help build signature logic and help fine tune rule implementation. In the following link you can find an extensive description of Perl PCRE expressions.

http://perldoc.perl.org/perlre.html

All elements that make up a rule must be true for the indicated rule action to be taken. When taken together, the elements can be considered to form a logical AND statement. At the same time, the various rules in a Snort rules library file can be considered to form a large logical OR statement.

The SiemSensor application simplifies the task of adjusting the rules you have defined in the rules file, allowing you to modify (via web or text configuration) a set of parameters that are usually mutable depending on the circumstances. These are IPs and ports lists that are mapped in variables in the rules file. The equivalence between variables and configuration can be found in the following list:

- *HOME_NET*: Local networks
- *EXTERNAL_NET*: External networks
- *DNS_SERVERS*: DNS servers
- *HTTP_SERVERS*: HTTP servers
- *SMTP_SERVERS* : SMTP servers
- *TELNET_SERVERS*: Telnet servers
- *FTP_SERVERS*: FTP servers
- *SQL_SERVERS*: SQL servers
- *SNMP_SERVERS*: SNMP servers
- *HTTP_PORTS*: HTTP server ports
- *SHELLCODE_PORTS*: Shellcode attack ports
- *ORACLE_PORTS*: Oracle server ports
- *FTP_PORTS*: FTP server ports
- *FILE_DATA_PORTS* : File data ports

More information on the design of the rules can be found in this link: *http://manual.snort.org/node27.html*

Chapter 3 Configuring the application

This application, like other applications installed on the Atlas i6x device, is configurable in two ways: by using the the Atlas i6x internal web, or by using a text configuration or a configuration template inside the management platform to configure one or more devices simultaneously.

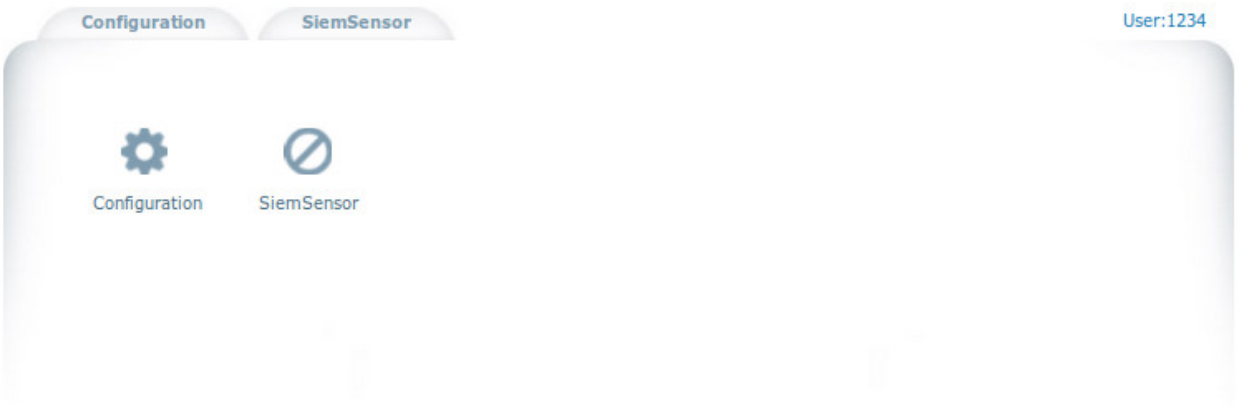
Note

Remember that traffic must be exported from CIT to Linux to make it available for the application. Otherwise, SiemSensor will not operate as expected. Please see Appendix *CIT Configuration* on page 15 to find out how to do it.

3.1 Web configuration

The application is represented, in the main window, as an icon over the SiemSensor name. To access the application's configuration menu, press the SiemSensor tab or click on the SiemSensor icon or name.

Fig. 3.1. Web configuration window



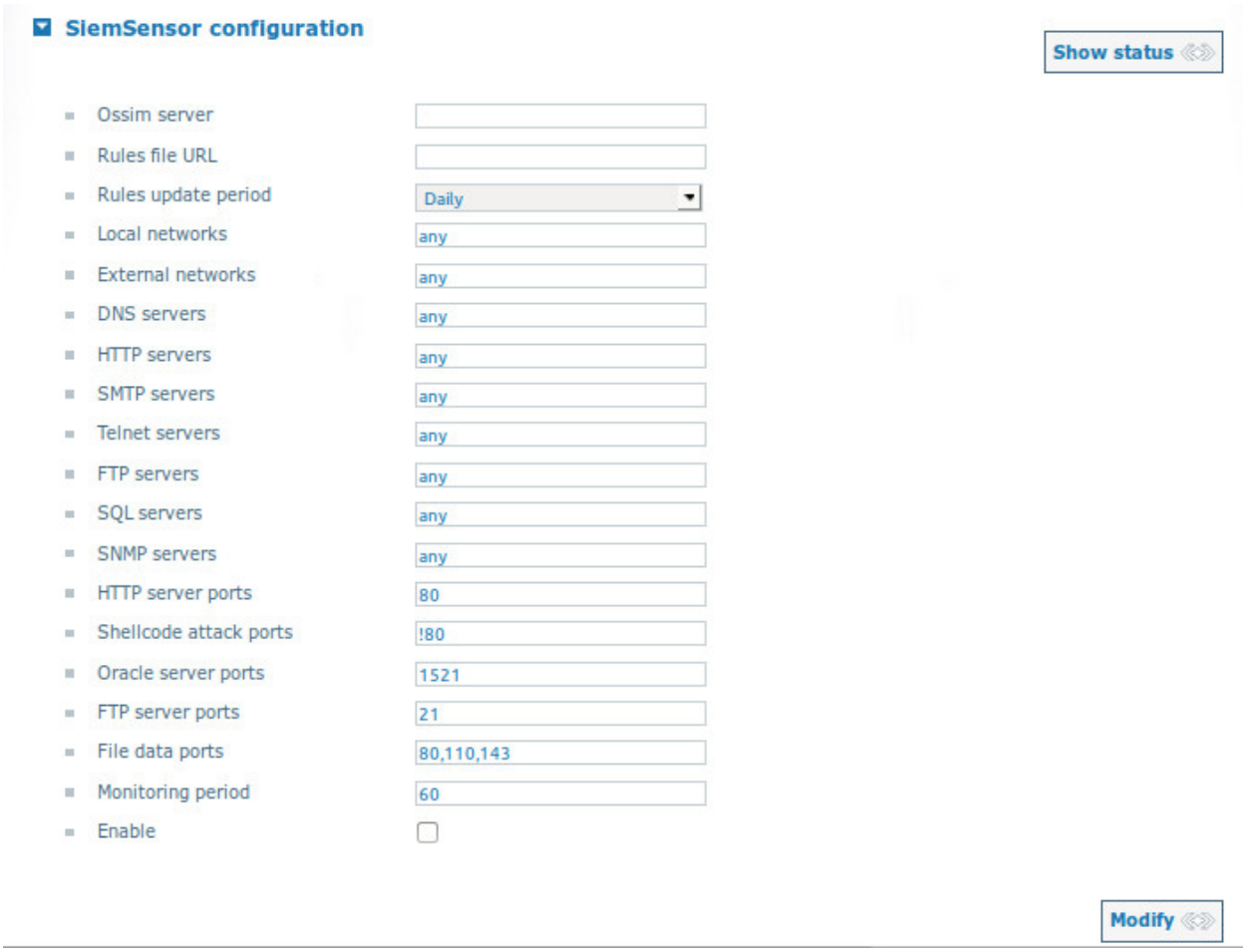
Clicking on one of them opens a new window. Click on SiemSensor (in the left-hand menu) to access the main configuration section.

Note

Summary of interface buttons:

- Modify : Use this button to modify the value of the current section parameters. You must click on this button before pressing any other (such as a table button). Otherwise, the whole page refreshes and you lose any modifications you have made.
- Show status : Use this button to show status parameters.
- Show conf : Use this button to return to the configuration section.

Fig. 3.2. Main configuration window



3.1.1 Application: General Parameters

The previous figure shows the application configuration parameters. The meaning of each parameter is explained in this section.

In this window you can also see two buttons: Show status and Modify. When the modify button is clicked, the changes made in the configuration parameters are saved and applied. When the status button is pressed, the status screen appears.

The status screen shows information on the current status of the application. More specifically, this information includes:

- The Ossim agent status: it can be stopped, connecting or connected.
- The number of captured events.
- The Snort status: stopped or running.
- The number of packets received.
- The number of packets analyzed.
- The number of packets dropped.
- And the number of alerts.

Fig. 3.3. Status window

Click on the Show conf button to return from the status screen to the parameter configuration window.

To save the changes made in the configuration parameters, press the button Modify . The configuration changes are applied immediately.

3.1.1.1 Ossim server

This parameter indicates the IP address for the OSSIM server, which must be valid. Lists or CIDR blocks are not allowed.

3.1.1.2 Rules File URL

This is the document URL. Said document contains the rules going to be downloaded. This URL must be periodically accessed to update the set of rules. The value has to be a well-formed URL.

3.1.1.3 Rules update period

Period of time between rule updates. Three options can be selected: Never (no update will be done), Daily or Weekly.

Note

Unless otherwise specified in the parameter description, IPs may be specified individually, in a list, as a CIDR block or as a list of CIDR blocks.

IPs and CIDR blocks may be negated with "!". Lists carry out an OR operation with the non-negated elements, take the results and carry out an AND operation. Using this last result, they carry out another OR operation with the negated elements. For example:

1.1.1.1/32,2.2.2.0/24,!2.2.2.2/32,!2.2.2.3/32

Matches the IP 1.1.1.1 and IP from 2.2.2.0 to 2.2.2.255, with the exception of 2.2.2.2 and 2.2.2.3.

When more than one IP address is inserted in a field, the order of elements does not matter. The element “any” can be used to match all IPs, although “!any” is not allowed. Also, negated IP ranges that are more general than non-negated IP ranges are not allowed.

3.1.1.4 Local networks

Set up the network addresses you are protecting. There must be one or more CIDR blocks.

3.1.1.5 External networks

Set up the external network addresses. There must be one or more CIDR blocks.

3.1.1.6 DNS servers

This parameter indicates one or more Domain Name System IP addresses.

3.1.1.7 HTTP servers

List of web servers on your network.

3.1.1.8 SMTP servers

List of SMTP servers on your network.

3.1.1.9 Telnet servers

List of Telnet servers on your network.

3.1.1.10 FTP servers

List of FTP servers on your network.

3.1.1.11 SQL servers

List of SQL servers on your network.

3.1.1.12 SNMP servers

List of SNMP servers on your network.

Note

Ports fields support the declaration of lists and ranges of ports. Variables, ranges, or lists may all be negated with "!". Also, "any" specifies any ports, but "!any" is not allowed. Valid port range from 0 to 65535.

Port ranges may be specified with a ":", such as in:

80:90,888:900

3.1.1.13 HTTP server ports

List of ports you run web servers on.

3.1.1.14 Shellcode attack ports

List of ports you want to look for shellcode on.

3.1.1.15 Oracle server ports

List of ports you might see Oracle attacks on.

3.1.1.16 FTP server ports

List of ports you run FTP servers on.

3.1.1.17 File data ports

List of file data ports for file inspection.

3.1.1.18 Monitoring period

Seconds between SiemSensor status information updates. This must be a positive value.

3.1.1.19 Enable

Check this box to enable SiemSensor.

3.2 Text configuration commands

This section describes all the configuration directives allowed in the application's text configuration.

Note

The configuration directives should be sent in a single text file to the device, through the Atlas i6x management portal.

If a statement does not appear in the configuration text, the engine will use the default value.

3.2.1 SiemSensor configuration

```
siemsensor
```

Top level configuration directive

3.2.1.1 Ossim server

```
ossim-server <value>
```

Ossim server IP address.This must be a valid IP address. Lists or CIDR blocks are not allowed.

3.2.1.2 Rules file URL

```
rules-url <value>
```

This is the document's URL. Said document contains the rules that are going to be downloaded. This URL must be periodically accessed to update the set of rules. The value has to be a well-formed URL.

3.2.1.3 Rules update period

```
update-period <value>
```

Period of time between rule updates. Three options can be selected: Never (no update will be done), Daily or Weekly.

Default value: Daily

3.2.1.4 Local networks

```
localnet <value>
```

Network addresses you are protecting.

Default value: any

3.2.1.5 External networks

```
externalnet <value>
```

External network addresses.

Default value: any

3.2.1.6 DNS servers

```
dns-servers <value>
```

One or more Domain Name System IP addresses.

Default value: any

3.2.1.7 HTTP servers

```
http-servers <value>
```

List of web servers on your network.

Default value: any

3.2.1.8 SMTP servers

```
smtp-servers <value>
```

List of SMTP servers on your network.

Default value: any

3.2.1.9 Telnet servers

```
telnet-servers <value>
```

List of Telnet servers on your network.

Default value: any

3.2.1.10 FTP servers

```
ftp-servers <value>
```

List of FTP servers on your network.

Default value: any

3.2.1.11 SQL servers

```
sql-servers <value>
```

List of SQL servers on your network.

Default value: any

3.2.1.12 SNMP servers

```
snmp-servers <value>
```

List of SMTP servers on your network.

Default value: any

3.2.1.13 HTTP server ports

```
http-ports <value>
```

List of web servers on your network.

Default value: 80

3.2.1.14 Shellcode attack ports

```
shellcode-ports <value>
```

List of ports you want to look for shellcode on.

Default value: !80

3.2.1.15 Oracle server ports

```
oracle-ports <value>
```

List of ports you might see Oracle attacks on.

Default value: 1521

3.2.1.16 FTP server ports

```
ftp-ports <value>
```

List of ports you run FTP servers on.

Default value: 21

3.2.1.17 File data ports

```
filedata-ports <value>
```

List of file data ports for file inspection.

Default value: 80,110,143

3.2.1.18 Monitoring period

```
monitoring-update-period <value>
```

Seconds between SiemSensor status information updates.This must be a positive value.

Default value: 60

3.2.1.19 Enable SiemSensor

```
enable
```

Enable SiemSensor application.

Default value: Disabled. Insert *enable*in the configuration text to activate the engine.

Chapter 4 Use cases

In this section, several use cases are presented to show the SiemSensor application capabilities.

4.1 Checking for incorrect login on Telnet sessions

To check incorrect login attempts on the Telnet server port, the following rule must be defined in the rules file:

```
alert tcp $TELNET_SERVERS $TELNET_PORTS -> $EXTERNAL_NET any (msg:"TELNET login
 incorrect"; content:"Login incorrect"; flow:from_server,established;
 classtype:bad-unknown; sid:718; rev:6;)
```

The rule generates an alert that applies to TCP packets. The variables $TELNET_SERVERS, $TELNET_PORTS, $EXTERNAL_NET are the data introduced in the Telnet servers, Telnet server ports and External network fields on the web configuration page. For example, they could be: "10.0.0.3, 10.0.0.4", "23" and "!10.0.0.0/24". The rule only checks the response from Telnet servers (10.0.0.3, 10.0.0.4), not the requests, and applies to the Telnet sessions coming from outside the private network (!10.0.0.0/24). If someone from the internal network starts a Telnet session, the rule will not detect said traffic.

4.2 PCI DSS compliance in retail

The PCI DSS (Payment Card Industry Data Security Standard) is an information security standard for organizations that handle branded credit cards from major providers (including Visa, MasterCard, American Express, JCB and Discover). The standard was created to increase controls on card holder data, using the information to reduce credit card fraud. Financial institutions and credit card companies require businesses to comply with this standard. Although there are twelve security requirements altogether, we are presently going to focus on one: the tracking and monitoring of every access to network resources and card holder data.

An OSSIM server can perform this task if OSSIM agents, located in different parts of the network, provide data. The compilation of logs and threat detection are some of its main functions. With the SiemSensor application, you can monitor the traffic coming from the Atlas i6x, compare the traffic with a set of pre-established rules and send alerts to the OSSIM server in the event of anomalies or suspicious traffic. To meet PCI DSS requirements, the pre-set rules must include Payment Card security guidelines (such as the detection of numbers or other credit card data present in packets and network resources accesses).

Appendix A CIT Configuration

The SiemSensor application needs to be specifically configured on the CIT to operate correctly. The traffic we want the SiemSensor to analyze has to be exported from CIT to Linux (so it's available for the application). The exported traffic is copied (not diverted) so the CIT can carry on managing the original traffic as per usual.

To export the traffic, you have to manually modify the CIT configuration. The following steps show you how to do it:

(a) Telnet to the CIT IP address.

```
telnet <CIT-address>
```

(b) Enter the dynamic configuration mode.

```
p 5
```

(c) Configure an access list (you may modify an existing access list or create a new one). This depends on whether the configuration already has an access list for traffic export, defined or not. First, access the access lists menu.

```
feature access-lists
```

- *Create a new access list.* To export all the traffic, create the following access list:

```
      access-list 102
         description vli_traffic_export
;
         entry 1 default
         entry 1 permit
;
      exit
```

  This access list allows all traffic to be exported. You may create an access list that only allows certain kinds of traffic to be exported (by filtering per protocol, ports and other options, which are out of the scope of this quick guide).

  Copy the access list to the CIT configuration here, in the access-list menu.

- *Modify an existing access list.* If an access list is being used for traffic export, you have to add entries for the kind of traffic that you want to export to the existing configuration (or allow all traffic to be exported, as described in a previous section on how to create an access list). You can copy the example in the access lists menu as if it were a new access list.

(d) Exit the access list menu.

```
exit
```

(e) Enter the VLI menu.

```
feature vli
```

(f) Enable traffic export if it is not already enabled. Insert the number of your access list.

```
application traffic-export access-list <access-list-number>
```

(g) Exit the VLI menu.

```
exit
```

(h) Exit the dynamic configuration.

```
ctrl + p
```

(i) Exit Telnet.

```
logout
```

Appendix B Troubleshooting

Below,there are some examples of several common scenarios and their outcome. They may help you deal with situations where the Atlas i6x SiemSensor application does not behave as expected.

We assume your SiemSensor application is installed in the Atlas i6x Application Host and the device is accessible over the network.

B.1 Symptom: Sending the same packet several times and only getting an alert for some

Sometimes, a packet that matches a certain rule may not trigger an alert. This false-negative situation can be the result of a work overload on the device.

The Snort has to analyze every packet received. If the amount of traffic is very high, or there are a lot of rules (every packet is checked against each rule), the performance of the SiemSensor will decrease. When this happens, the packets that cannot be managed by the overloaded Snort are dropped without being analyzed. This means that sending two identical packets can trigger an alarm in one case and receive no response in the other (because the packet has been dropped).

A packet is routed even if Snort drops it. Remember that the analyzed traffic is exported, i.e. copied from the communications core to the applications core. All traffic is routed normally.

To avoid packages from being systematically dropped, set a reasonable number of rules.

B.2 Symptom: Performance degradation

As stated in the previous section, *Symptom: Sending the same packet several times and only getting an alert for some* on page 17, the Snort checks each packet received against all the rules. If using the SiemSensor application makes the device run slower or perform less, the set of rules could be too large (making the process of checking every packet overly expensive). A heavy traffic load can also have the same effect.

We highly recommend you don't set a large number of rules. Instead, please pick the rules that best meet the security needs of your business and try to keep the number down.

B.3 Symptom: The system generates false alerts

IDS false positives signal an attack when there is none, triggering a false alert. To detect an intrusion, a simple pattern matching of signatures is often insufficient. However, that's what we do with the Snort and our set of rules. If the signature is not carefully designed, there will be lots of matches. Despite the fact that 99.99% of the analyzed data is made up of non-anomalous traffic, all attack tests need to be performed on each packet. As a result, accuracy is often traded for speed. Resorting to a set of rules that is neither 100% precise nor unique may trigger some false alerts, but is a reasonable trade-off for better overall performance.

B.4 Symptom: SiemSensor is enabled but Snort and OSSIM agent status are "stopped"

Snort starts the processing before OSSIM agents do. There may be different reasons for Snort not starting. The most common one is that some of the rules in the rules file contain a syntax error. When this happens, Snort does not start. Please check that there are no mistakes in your rules.

B.5 Symptom: SiemSensor is enabled, services are running, but apparently no alerts are triggered

It's possible that no traffic is being analyzed. Please check you are exporting traffic from CIT to the Linux, as described in Appendix *CIT Configuration* on page 15.