How To Configure Templates for fw monitor

Technical Reference Guide

29 August 2011
Important Information

Latest Software
We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

Latest Documentation
The latest version of this document is at:
http://supportcontent.checkpoint.com/documentation_download?ID=12312
For additional technical information, visit the Check Point Support Center (http://supportcenter.checkpoint.com).

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>29 August 2011</td>
<td>First release of this document</td>
</tr>
</tbody>
</table>

Feedback
Check Point is engaged in a continuous effort to improve its documentation.

Please help us by sending your comments (mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on How To Configure Templates for fw monitor Technical Reference Guide).
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Important Information</td>
<td>3</td>
</tr>
<tr>
<td>How to Configure Templates FWMonitor</td>
<td>5</td>
</tr>
<tr>
<td>Objective</td>
<td>5</td>
</tr>
<tr>
<td>Supported Versions</td>
<td>5</td>
</tr>
<tr>
<td>Supported OS</td>
<td>5</td>
</tr>
<tr>
<td>Supported Appliances</td>
<td>5</td>
</tr>
<tr>
<td>Before You Start</td>
<td>5</td>
</tr>
<tr>
<td>Related Documentation</td>
<td>5</td>
</tr>
<tr>
<td>Assumed Knowledge</td>
<td>6</td>
</tr>
<tr>
<td>Impact Environment and Warnings</td>
<td>6</td>
</tr>
<tr>
<td>How fw monitor works</td>
<td>6</td>
</tr>
<tr>
<td>Using fw monitor</td>
<td>7</td>
</tr>
<tr>
<td>Command Line fw monitor Switches</td>
<td>8</td>
</tr>
<tr>
<td>fw monitor Capture Masks</td>
<td>9</td>
</tr>
<tr>
<td>fw monitor Filters</td>
<td>10</td>
</tr>
<tr>
<td>Data Types</td>
<td>10</td>
</tr>
<tr>
<td>Logical and Relational Operators</td>
<td>10</td>
</tr>
<tr>
<td>Macros</td>
<td>11</td>
</tr>
<tr>
<td>Using fw monitor with the fw ctl Chain</td>
<td>11</td>
</tr>
<tr>
<td>Index</td>
<td>13</td>
</tr>
</tbody>
</table>
How to Configure Templates FWMonitor

Objective
Inspecting network traffic through a Firewall is an essential part of deployment and troubleshooting tasks. *fw monitor* is a powerful built-in tool by Check Point, used to simplify this task. *fw monitor* captures network packets at multiple capture points within the Firewall through all interfaces simultaneously. This document briefly describes how to use *fw monitor* and its features to simplify traffic capture through a firewall.

Supported Versions
Supported on all versions.

Supported OS
Supported on all OS platforms.

Supported Appliances
Supported on all appliances and open servers.

Before You Start

Related Documentation

sk41045 - fw monitor command
(https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_dogoviewsolutiondetails=&solutionid=sk41045&js_peid=p-114a7ba5fd7-10001&partition=general&product=vsx, -
http://supportcontent.checkpoint.com/solutions?id=sk41045)

sk30583 - what is fw monitor
(https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_dogoviewsolutiondetails=&solutionid=sk30583&js_peid=p-114a7ba5fd7-10001&partition=general&product=security -
http://supportcontent.checkpoint.com/solutions?id=sk30583)

sk41059 - How to interpret fw monitor output files in Wireshark
(https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_dogoviewsolutiondetails=&solutionid=sk41059&js_peid=p-114a7ba5fd7-10001&partition=general&product=security -
http://supportcontent.checkpoint.com/solutions?id=sk41059)

sk33358 - Useful FW Monitor commands
(https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_dogoviewsolutiondetails=&solutionid=sk33358&js_peid=p-114a7bc3b09-10006&partition=expert&product=security -
http://supportcontent.checkpoint.com/solutions?id=sk33358)
Assumed Knowledge

- Working knowledge of network technology
- General knowledge of TCP / IP.
- General knowledge of packet flow through Check Point Gateway.
- General usage of packet protocol analyzers like snoop, tcpdump, Wireshark or Ethereal.
- General knowledge about Firewall chain modules + INSPECT filter.

Impact Environment and Warnings

The fw monitor command can cripple a Firewall that is already under heavy load. It is always best to test packet captures during off peak times. If you are testing kernel drops, make sure to run them at the same time, so you can reference packets in the drop file to the packet capture.

It is recommended to run the fw monitor command from a directory with plenty of space so that you do not fill up the hard drive, such as /var or c:\temp.

How fw monitor works

In contrast to other capturing tools like snoop or tcpdump, fw monitor does not use the promiscuous mode on network interface cards. Based on the fact that the Firewall already receives all packets (due to the kernel module between the NIC driver and IP stack) fw monitor uses its own kernel module to capture packets (compared to filtering/encrypting them).

Unlike snoop or tcpdump, fw monitor has the ability to capture packets at different positions in the FireWall-1 kernel module chain; snoop and tcpdump are capturing packets when they enter or leave the computer. Especially when NAT is involved fw monitor offers the possibility to capture packets at multiple locations. This can help you to see how the packets are translated by the firewall and on which IP address the routing decision is made.

fw monitor is able to capture packets at four different positions in the Firewall: There are four inspection points as a packet passes through the virtual machine

- on the inbound interface before the Virtual Machine (pre-inbound)
Using fw monitor

The easiest way to use *fw monitor* is to invoke it without any parameter. This will output every packet from every interface that passes (or at least reaches) the enforcement module. Please note that the same packet is appearing several times (two times in the example below). This is caused by *fw monitor* capturing the packets at different capture points.

```
[cpmodule]# fw monitor
  monitor: getting filter (from command line)
  monitor: compiling
  monitorfilter:
  Compiled OK.
  monitor: loading
  monitor: monitoring (control-C to stop)
eth0:i[285]: 172.16.1.133 -> 172.16.1.2 (TCP) len=285 id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
eth0:i[285]: 172.16.1.133 -> 172.16.1.2 (TCP) len=285 id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
eth0:o[197]: 172.16.1.12 -> 172.16.1.133 (TCP) len=197 id=44599
TCP: 18190 -> 1050 ...PA. seq=941b05bc ack=bf8bca83
eth0:o[197]: 172.16.1.12 -> 172.16.1.133 (TCP) len=197 id=44599
TCP: 18190 -> 1050 ...PA. seq=941b05bc ack=bf8bca83
eth0:o[1500]: 172.16.1.12 -> 172.16.1.133 (TCP) len=1500 id=44600
TCP: 18190 -> 1050 ...A. seq=941b0659 ack=bf8bca83
^C
monitor: caught sig 2
monitor: unloading
```

**Break Sequence**

Use `^C` (that is Control + C) to stop *fw monitor* from capturing packets.

```
eth0:i[285]: 172.16.1.133 -> 172.16.1.2 (TCP) len=285 id=1075
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
```

The above packet was captured on the first network interface (eth0) in inbound direction before the virtual machine (lowercase i)

```
TCP: 1050 -> 18190 ...PA. seq=bf8bc98e ack=941b05bc
```

The second line tells us that this is an TCP payload inside the IP packet which was sent from port 1050 to port 18190. The following element displays the TCP flags set (in this case PUSH and ACK). The last two elements are showing the sequence number (seq=bf8bc98e) of the TCP packet and the acknowledged sequence number (ack=941b05bc). You will see similar information for UDP packets.

You will only see a second line if the transport protocol used is known to *fw monitor*. Known protocols are for example TCP, UDP and ICMP. If the transport protocol is unknown or cannot be analyzed because it is encrypted (e.g. ESP or encapsulated (e.g. GRE) the second line will be missing.
Command Line fw monitor Switches

The syntax for fw monitor is:

```
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u</td>
<td>s</td>
</tr>
<tr>
<td>-i</td>
<td><strong>Flushing the standard output:</strong> Use to make sure that captured data for each packet is at once written to standard output. This is especially useful if you want to kill a running fw monitor process and want to be sure that all data is written to a file.</td>
</tr>
<tr>
<td>[-d] [-D]</td>
<td><strong>Debugging fw monitor:</strong> The -d option is used to start fw monitor in debug mode. This will give you an insight into fw monitor's inner workings. This option is only rarely used outside Check Point. It is also possible to use –D to create an even more verbose output.</td>
</tr>
<tr>
<td>&lt;{ e expr}+</td>
<td>{f &lt;filter-file}&gt; ] &gt;&gt;</td>
</tr>
<tr>
<td>-l len</td>
<td><strong>Limiting the packet length:</strong> fw monitor allow you to limit the packet data which will be read from the kernel with -l. This is especially useful if you have to debug high sensitive communication. It allows you to capture only the headers of a packet (e.g. IP and TCP header) while omitting the actual payload. Therefore you can debug the communication without seeing the actual data transmitted. Another possibility is to keep the amount of data low. If you don't need the actual payload for debugging you can decrease the file size by omitting the payload. It's also very useful to reduce packet loss on high-loaded machines. fw monitor uses a buffer to transfer the packets from kernel to user space. If you reduce the size of a single packet this buffer won't fill up so fast.</td>
</tr>
<tr>
<td>m mask</td>
<td><strong>Setting capture masks:</strong> By default fw monitor captures packets before and after the virtual machine in both directions. These positions can be changed. This option allows you to specify in which of the four positions you are interested.</td>
</tr>
<tr>
<td>-x offset[len]</td>
<td><strong>Printing packet/payload data:</strong> In addition to the IP and Transport header fw monitor can also print the packets’ raw data using the –x option. Optionally it is also possible to send all data that is written only to the screen the data written.</td>
</tr>
<tr>
<td>-o &lt;file&gt;</td>
<td><strong>Write output to file:</strong> Save the raw packet data to a file in a standard (RFC 1761) format. The file can be examined using by tools like snoop, tcpdump or Ethereal. Note - The snoop file format is normally used to store Layer 2 frames. For &quot;normal&quot; capture files this means that the frame includes data like a source and a destination MAC address. fw monitor operates in the firewall kernel and therefore has no access to Layer 2 information like MAC addresses. Instead of writing random MAC addresses, fw monitor includes information like interface name, direction and chain position as &quot;MAC addresses&quot;.</td>
</tr>
<tr>
<td>-T</td>
<td><strong>Print time stamp in microseconds. -T is needed only when -o is not used. When -o is used the exact time is written to the snoop file by default as of Corsica.</strong></td>
</tr>
</tbody>
</table>
Insert fw monitor chain module at a specific position: In addition to capture masks (which give the ability to look at packets in a specific position) fw monitor has the ability to define where exactly in the firewall chain the packets should be captured. This can be defined using these options.

Use absolute chain positions: If you use fw monitor to output the capture into a file (option –o), one of the fields written down to the capture file is the chain position of the fw monitor chain module. Together with a simultaneous execution of fw ctl chain you can determine where the packet was captured. Especially when using –p all you will find the same packet captured multiples times at different chain positions. The option –a changes the chain id from an relative value (which only makes sense with the matching fw ctl chain output) to an absolute value. These absolute values are known to CPEthereal and can be displayed by it.

Capture a specific number of packets: fw monitor enables you to limit the number of packets being captured. This is especially useful in situations where the firewall is filtering high amounts of traffic. In such situations fw monitor may bind so many resources (for writing to the console or to a file) that recognizing the break sequence (Control-C) might take very long.

Capture on a specific Virtual Router or Virtual Machine: VPN-1 Power VSX enables you to run multiple Virtual Routers and Firewalls on one physical machine. Using the option –vs you can specify on which virtual component the packets should be captured. This option is only available on a VPN-1 Power VSX module. Please refer to fw monitor on FireWall-1 VSX for more information.

Displays the usage.

fw monitor Capture Masks

By default fw monitor captures packets at all four positions. With -m it is possible to capture packets at specific positions. fw monitor uses single letters as indicators for the position:

<table>
<thead>
<tr>
<th>Capture position</th>
<th>fw monitor mask value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pre-inbound i</td>
<td>i (lowercase i)</td>
</tr>
<tr>
<td>post-inbound I</td>
<td>I (uppercase i)</td>
</tr>
<tr>
<td>pre-outbound o</td>
<td>o (lowercase o)</td>
</tr>
<tr>
<td>post-outbound O</td>
<td>O (uppercase o)</td>
</tr>
</tbody>
</table>

Using fw monitor masks it is easily possible to capture only packets before they are inspected by the firewall in inbound direction and after they have been inspected by the firewall in outbound direction.

In the example below we are capturing traffic between a client (10.2.4.12) and a web server (172.16.1.1). The client address is translated to 172.16.1.3 and the server address is translated to 10.2.253.2. You can easily see how the non-translated packet enters the firewall and how the translated packet (source and destination) is leaving the firewall:
Using the right combination of capture masks it’s very easy to find out when the firewall applies which NAT rules (Hide NAT, Static Destination NAT or Static Source NAT). This is especially useful when you need to know which packets the routing of the operating system is using to do the routing decision.

fw monitor Filters

fw monitor filters use a subset of INSPECT to specify the packets to be captured. The general syntax is the accept expression:

"accept" in fw monitor filters does not mean that packets are actually accepted by the firewall. fw monitor captures all packets which are accepted by the filter and discards the rest. A filter like accept; (capturing all packets) will in no way change the behavior of the Firewall and its rule base.

The complexity of an expression can vary from a simple test (checking for a specific value at a specific offset) to a complex expression using different checks and logical operators.

Data Types

INSPECT knows several native data types. Just some of them are useful for fw monitor:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexadecimal Integers</td>
<td>A number beginning with 0x</td>
<td>e.g. 0x5ab4</td>
</tr>
<tr>
<td>Octal Integers</td>
<td>A Number beginning with 0</td>
<td>e.g. 0777</td>
</tr>
<tr>
<td>Decimal Integers</td>
<td>Any other number</td>
<td>e.g. 23</td>
</tr>
<tr>
<td>IP Address</td>
<td>Four decimal integers separated by three periods</td>
<td>e.g. 172.45.2.4</td>
</tr>
</tbody>
</table>

Logical and Relational Operators

In addition to the single expressions testing for equality, you can combine different expressions using several logical and relational operators.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>= or is</td>
<td>Equal</td>
</tr>
<tr>
<td>!= or is not</td>
<td>Not equal</td>
</tr>
<tr>
<td>,</td>
<td>Logical AND</td>
</tr>
<tr>
<td>or</td>
<td>Logical Or</td>
</tr>
<tr>
<td>xor</td>
<td>Logical XOR</td>
</tr>
<tr>
<td>not</td>
<td>Logical NOT</td>
</tr>
</tbody>
</table>

### Macros

`fw monitor` offers an more intuitive way of specifying the desired field:

<table>
<thead>
<tr>
<th>Field</th>
<th>Macro</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>source address</td>
<td>src</td>
<td>[12:4,b]</td>
</tr>
<tr>
<td>destination address</td>
<td>dst</td>
<td>[16:4,b]</td>
</tr>
<tr>
<td>source port</td>
<td>sport</td>
<td>[20:2,b]</td>
</tr>
<tr>
<td>destination port</td>
<td>dport</td>
<td>[22:2,b]</td>
</tr>
</tbody>
</table>

Using these macros it very easy to define filters. Here are some examples:

- Captures everything except http traffic.  
  ```
  #fw "accept not ( sport=80 or dport=80);"
  ```

- All TCP packets sent between host 10.2.4.12 and 172.16.1.2  
  ```
  #fw monitor "accept [9:1]=9 , ((src=10.2.4.12 ,
  dst=172.16.1.2) or (src=172.16.1.2 , dst=10.2.4.12));"
  ```

- Captures all traffic from and to the host 172.29.109.1  
  ```
  #fw monitor –e "accept src=172.29.109.1 or
dst=172.29.109.1;"
  ```

- Captures all http traffic on port 80 only  
  ```
  #fw monitor –e "accept dport==80;"
  ```

- 3rd filter will capture only inbound direction before and after the virtual machine (I and I), and redirects the output to a file.  
  ```
  #fw monitor –m il –e "accept;" –o monitor.out
  ```

### Using `fw monitor` with the `fw ctl Chain`

Check Point uses a "kernel module chain" for different kernel modules which are working with the packets.
You can see the actual chain using the `fw ctl chain` command. This shows you the chain modules actually loaded on your machine and their order. `fw monitor` can be inserted in any position in the chain. Note that there are more kernel modules in the chain which are not visible by `fw ctl chain` and which cannot be used for `fw monitor` kernel module positioning.

```
[Expert@cpmodule]# fw ctl chain
in chain (9):
  0: -7f800000 (ca8d9698) IP Options Strip (ipopt_strip)
  1:  2000000 (cblc1c64) vpn decrypt (vpn)
  2:  1fffff6 (ca8da0f8) Stateless verifications (asm)
  3:  1fffff0 (cblc17f0) vpn decrypt verify (vpn_ver)
  4:  1000000 (ca8eb688) SecureXL connection syn (secxl_sync)
  5:   0 (ca8aa0c0) fw VM inbound (fw)
  6:  2000000 (cblc2aa0) vpn policy inbound (vpn_pol)
  7:  1000000 (ca8eb728) SecureXL inbound (secxl)
  8:  7f800000 (ca8d98e4) IP Options Restore (ipopt_res)
out chain (8):
  0: -7f800000 (ca8d9698) IP Options Strip (ipopt_strip)
  1:  1fffff2 (cblc16fc) vpn nat outbound (vpn_nat)
  2:  1f00000 (ca8da0f8) Stateless verifications (asm)
  3:   0 (ca8aa0c0) fw VM outbound (fw)
  4:  2000000 (cblc26e0) vpn policy outbound (vpn_pol)
  5:  1000000 (ca8eb728) SecureXL outbound (secxl)
  6:  2000000 (cblc2164) vpn encrypt (vpn)
  7:  7f800000 (ca8d98e4) IP Options Restore (ipopt_res)
```

The output of `fw ctl chain` is platform, version and product dependent. There is no reason to worry if your `fw ctl chain` output looks different to the above. The number and kind of modules displayed here may vary based on the platform used and products installed.

`fw monitor` inserts its own modules in this module chain and captures packets. By default this is not the first and last position in the chain. Therefore the original meaning of before and after needs to be redefined.
Index

A
Assumed Knowledge • 6

B
Before You Start • 5

C
command Line fw monitor Switches • 8

D
Data Types • 10

F
fw monitor Capture Masks • 9
fw monitor Filters • 10

H
How fw monitor works • 6
How to Configure Templates FWMonitor • 5

I
Impact Environment and Warnings • 6
Important Information • 3

L
Logical and Relational Operators • 10

M
Macros • 11

O
Objective • 5

R
Related Documentation • 5

S
Supported Appliances • 5
Supported OS • 5
Supported Versions • 5

U
Using fw monitor • 7
Using fw monitor with the fw ctl Chain • 11