Snort Webinar Series
Using MultiConfig

John Gay – jgay@sourcefire.com
Snort Multiconfig

- Allows Snort to have multiple configuration files
  - Separate configuration files based on Subnet or Vlan ID
  - Allows for different configurations without running multiple instances of Snort
  - Creates Unique Configuration Instances
Snort Multiconfig

• Default Configuration
  ▶ Standard snort.conf called with the –c option
    ● Used to call the non-default files
    ● Used for traffic not specified in the non-default configurations

  ▶ Multiple non-default config files
    ● Can specify by Vlan or Subnet
Snort Multiconfig

● To specify the non-default config files
  ▶ To Bind to Vlan
    ● Config binding: <path to non-default.conf> vlan <vlanIDList>
      – Vlan IDs can be comma separated list of Vlan IDs or ranges
        (ranges separated by “-”)

  ▶ To Bind to IP List
    ● Config binding: <path to non-default.conf> net <ipList>
      – Subnets can be IPv4 or IPv6 addresses
      – CIDR blocks or individual Ips
      – Maximum of 512 items
Snort Multiconfig

- Default Configuration
  - Any settings not defined in the non-default configuration uses the settings from the default
  - Preprocessors must be enabled in the default before they can be used in the non-default config
  - Memory options or number of instances for preprocessors is only read from the default config
  - Memory settings specified in the non-default configs are ignored
Snort Multiconfig

- Applying the Configuration - Vlan
  - Every incoming packet will be assigned to a configuration
    - If VlanID is present than the inner most VlanID is used
    - If the assigned configuration is the default then Snort looks for Subnet configs
    - The most specific subnet config based on the destination ID is used
    - If no non-default subnet config is found then Snort looks for the subnet config based on source IP
    - If nothing is found then the default config is used
Snort Multiconfig

- Applying the Configuration - Subnet
  - Every incoming packet will be assigned to a configuration
    - The most specific subnet config based on the destination ID is used
    - If no non-default subnet config is found then Snort looks for the subnet config based on source IP
    - If nothing is found then the default config is used
  - Subnet Configuration Conflicts
    - If there is an IP configuration conflict (source and destination IP’s match a configuration) The first matched non-default configuration will be applied.
Snort Multiconfig

• Configuration Specific Options
  ▶ The Config Options that are specific to each configuration are as follows:
    • config policy_id
      – 16-bit number used in unified2 output to identify alerts
    • config policy_mode
      – tap
      – inline
      – inline_test
    • config policy_version
      – Allows for versioning information to configuration files.
**Snort Multiconfig**

- **Configuration Specific Options**
  - If not defined in the specific configuration the following options will use their default configuration (not the same as what is in the default config file)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum_drop</td>
<td>disable_tcpopt_experimental_alerts</td>
</tr>
<tr>
<td>disable_decode_alerts</td>
<td>disable_tcpopt_experimental_drops</td>
</tr>
<tr>
<td>disable_decode_drops</td>
<td>disable_tcpoptObsolete_alerts</td>
</tr>
<tr>
<td>disable_ipopt_alerts</td>
<td>disable_tcpoptObsolete_drops</td>
</tr>
<tr>
<td>disable_ipopt_drops</td>
<td>disable_ttcp_alerts</td>
</tr>
<tr>
<td>disable_tcpopt_alerts</td>
<td>disable_tcpopt_ttcp_alerts</td>
</tr>
<tr>
<td>disable_tcpopt_drops</td>
<td>disable_ttcp_drops</td>
</tr>
<tr>
<td>disable_tcpopt_alerts</td>
<td></td>
</tr>
<tr>
<td>disable_tcpopt_ttcp_alerts</td>
<td></td>
</tr>
<tr>
<td>disable_tcpopt_drops</td>
<td></td>
</tr>
</tbody>
</table>
Snort Multiconfig

- Rules Configuration
  - Rules must be defined in each configuration.
  - If a Rule is not defined for a specific configuration then traffic for that configuration will not be parsed against that rule.
  - Rules share rule options with the default configuration
  - Rules with a higher revision will override versions with lesser revisions used in other configurations
  - Rules can be customized in each configuration but only for the following:
    - Source IP and Port
    - Destination IP and Port
    - Action
Snort Multiconfig

● Variable Configuration
  ▶ Variables must be defined in each configuration
  ▶ If defined rules make use of variables then those variables must be defined in those sections
Snort Multiconfig

- Rule Configuration
  - Rules must be defined in each configuration.
  - If a Rule is not defined for a specific configuration then traffic for that configuration will not be parsed against that rule.
  - Rules share rule options with the default configuration
  - Rules can be customized in each configuration but only for the following:
    - Source IP and Port
    - Destination IP and Port
    - Action
Snort Multiconfig

- Demonstration
  - Snort.conf

config binding: vlan1090.conf vlan 1090
config binding: vlan1099.conf vlan 1099

alert tcp 192.168.133.50 any -> 192.168.111.99 80 \ 
  (msg:"Syn from 133.50 to 111.99 default config"; \ 
  sid:1000000;flags:s;)

alert tcp 192.168.133.50 any -> 192.168.111.90 80 \ 
  (msg:"Syn from 133.50 to 111.90 default config"; \ 
  sid:1000001;flags:s;)

alert tcp 192.168.133.50 any -> 192.168.10.99 80 \ 
  (msg:"Syn from 133.50 to 111.99 default config"; \ 
  sid:1000099;)

alert tcp 192.168.133.50 any -> 192.168.10.90 80 \ 
  (msg:"Syn from 133.50 to 111.90 default config"; \ 
  sid:1000090;)
Snort Multiconfig

- Demonstration
  - vlan1090.conf

```plaintext
alert tcp 192.168.133.50 any -> 192.168.10.90 80 \n(msg:"Syn from 133.50 to 10.90 vlan 1090 config"; \nsid:1001090;flags:s;)
```
Snort Multiconfig

- Demonstration
  - vlan1099.conf

```plaintext
alert tcp 192.168.133.50 any -> 192.168.10.99 80 \
  (msg:"Syn from 133.50 to 10.99 vlan 1099 config"; \
  sid:1001099;flags:s;)
```
Snort Multiconfig

04/21-22:01:52.451181  [**] [1:1001099:0] Syn from 133.50 to 10.99 vlan 1099 config [**]  
[Classification ID: (null)]  [Priority ID: 0]  {TCP}  192.168.133.50:58235 -> 192.168.10.99:80

04/21-22:02:09.159597  [**] [1:1001090:0] Syn from 133.50 to 10.90 vlan 1090 config [**]  
[Classification ID: (null)]  [Priority ID: 0]  {TCP}  192.168.133.50:47560 -> 192.168.10.90:80

04/21-22:02:32.068960  [**] [1:1000001:0] Syn from 133.50 to 111.90 original config [**]  
[Classification ID: (null)]  [Priority ID: 0]  {TCP}  192.168.133.50:44580 -> 192.168.111.90:80

04/21-22:02:35.353598  [**] [1:1000001:0] Syn from 133.50 to 111.90 original config [**]  
[Classification ID: (null)]  [Priority ID: 0]  {TCP}  192.168.133.50:44580 -> 192.168.111.90:80

04/21-22:02:58.011130  [**] [1:1000000:0] Syn from 133.50 to 111.99 original config [**]  
[Classification ID: (null)]  [Priority ID: 0]  {TCP}  192.168.133.50:43133 -> 192.168.111.99:80

04/21-22:03:01.543702  [**] [1:1000000:0] Syn from 133.50 to 111.99 original config [**]  
[Classification ID: (null)]  [Priority ID: 0]  {TCP}  192.168.133.50:43133 -> 192.168.111.99:80
Tcpdump –tnxxr tcpdump.log.1303439829

IP 192.168.133.50.58235 > 192.168.10.99.http: S 189723166:189723166(0) win 5840
<mss 1460,sackOK,timestamp 103766155 0,nop,wscale 2>
0x0000:  000c 295d fe94 000c 29ea 2030 8100 044b
0x0010:  0800 4500 003c aa95 4000 4006 7f40 c0a8
0x0020:  8532 c0a8 0a63 e37b 0050 0b4e f21e 0000
0x0030:  0000 a002 16d0 e05b 0000 0204 05b4 0402
0x0040:  080a 062f 588b 0000 0000 0103 0302

IP 192.168.133.50.47560 > 192.168.10.90.http: S 199790626:199790626(0) win 5840
<mss 1460,sackOK,timestamp 103780623 0,nop,wscale 2>
0x0000:  000c 295d fe94 000c 29ea 2030 0800 4500
0x0010:  003c 0a37 4000 4006 baa7 c0a8 8532 c0a8
0x0020:  6f5a ae24 0050 0cde eb1d 0000 0000 a002
0x0030:  16d0 2d8d 0000 0204 05b4 0402 080a 062f
0x0040:  e12a 0000 0000 0103 0302

IP 192.168.133.50.44580 > 192.168.111.90.http: S 215935773:215935773(0) win 5840
<mss 1460,sackOK,timestamp 103801130 0,nop,wscale 2>
0x0000:  000c 295d fe94 000c 29ea 2030 0800 4500
0x0010:  003c 0a37 4000 4006 baa7 c0a8 8532 c0a8
0x0020:  6f63 a87d 0050 0fc2 c539 0000 0000 a002
0x0030:  16d0 32f6 0000 0204 05b4 0402 080a 062f
0x0040:  e12a 0000 0000 0103 0302

IP 192.168.133.50.43133 > 192.168.111.99.http: S 264422713:264422713(0) win 5840
<mss 1460,sackOK,timestamp 103823978 0,nop,wscale 2>
0x0000:  000c 295d fe94 000c 29ea 2030 0800 4500
0x0010:  003c ff53 4000 4006 baa7 c0a8 8532 c0a8
0x0020:  6f63 a87d 0050 0fc2 c539 0000 0000 a002
0x0030:  16d0 fcea 0000 0204 05b4 0402 080a 062f
0x0040:  3a6a 0000 0000 0103 0302