

A Novice's Guide to Using Multiple Layers of Snort to Defend the Home Network

by James McQuaid

At my home, I utilize a SANS-style layered defense consisting of four perimeter firewalls located within three conceptual rings. Each firewall performs a specific function that the others cannot. I chose to implement a Bleeding home topology due to the fact that **security vendors are losing the arms race on the Internet**. They are simply being outspent by the malware industry. The Internet is a wonderful technology that enriches our lives, but never before in history have sexual predators, organized crime, and hostile governments been able to reach into the homes of ordinary people (this is but one reason why the work of open source security volunteers is so very important).

In the outer ring, a firewall appliance is employed to filter out some inbound attacks and provide an initial layer of stateful packet inspection. I'm using a Netgear FVS-338, which is a Linux firewall (the source can be downloaded at Netgear's site). Outbound blocking is achieved by defining custom services which are set to Always Deny. To protect the FVS-338, you should configure it block all ICMP on the inbound interface as well as, any service ports that you do not use. The FVS-338 will block unrequested traffic on the inbound interface by default, but if a client machine becomes infected, malicious traffic will be solicited. Limit the Netgear's LAN IP range to the single IP address utilized by Smoothwall, and the two IP addresses assigned to the Honeybots. We previously used the FVS-318 in the same configuration, and found that it should be set to reboot itself daily. The FVS-338 provides six to nine times the bandwidth, and allows you to configure a second LAN address. You may be able to use the CISCO PIX on the outer ring depending upon RAM and installed licensing.

In the center ring of the perimeter defense, we've deployed two Snort Inline machines in bridging mode (bridging NICs do not have IP addresses making these firewalls difficult to attack). The host machines should have a considerable amount of RAM and processing power: 2 GB of RAM is best. The Honeywall ISO (<http://www.honeynet.org>) is an excellent distro, but if you have good UNIX or Linux skills, you will be rewarded with additional flexibility in your preprocessor configurations by compiling your own Snort Inline. The machine deployed closest to the FVS-338 is the most recent Honeywall distribution: roo-1.2.hw-1.iso which can be downloaded at <http://www.honeynet.org/tools/cdrom/roo/iso/current/>. This Snort Inline uses Source Fire's rules and configuration with the exception that we've added our own disallowed-ports.rules file, and http inspection is set at 100%. Roo 1.2 makes decisions for you regarding drop and replace options. The configuration menu now includes a "Generate IPS Rules" option, which you should use to enable the Honeywall's intrusion prevention capabilities. The /etc/blacklist.txt file is used to block hostile IP addresses. We have over 4,000 IP ranges blocked, having accumulated addresses from the **SANS ISC Top 10 Sources** (<http://isc.sans.org/top10.html>) list and Handler's Diaries (<http://isc.sans.org/>), **shadowserver.org** (<http://www.shadowserver.org/wiki/>), Symantec's Threat Explorer (http://www.symantec.com/smb/security_response/threatexplorer/threats.jsp) page, Know Your Enemy: Malicious Web Servers (<http://www.honeynet.org/papers/mws/index.html>), SRI's Multiperspective Malware Infection Analysis Page (

[analysis/public/](#)), observations at work, home logs, etc. Blocked ranges are expressed in CIDR notation. With this many ranges dropped, you will need to sort your blacklist file; the dual processor Poweredge requires approximately 25 minutes to boot up (the majority of this time is spent on the IRQ Handler setup). In the whitelist.txt file, you will need to list your ISP's DNS servers and your gateway.

Between the two Honeywall's you will need to deploy a switch with an MTU of 1500. We are using gigabit devices, Cat-6 and Cat-5e cabling across most of the network. This switch's power cord is plugged into an inexpensive, analog timer (commonly used for electric lamps). The timer turns off the switch during the night, **effectively reducing our attack window by 30%**, and enforces our house rules which prohibit teens from surfing online all night.

Our next line of defense consists of **Bleeding Snort Inline** (<http://www.bleedingthreats.net>) deployed on Honeywall Roo 1.1 (roo-1.1.hw-1.iso) available at <http://www.honeynet.org/tools/cdrom/roo/iso/archives/hw-1.2/>. **The Bleeding rules provide your home network with a substantive defensive capability against unknown threats.** You can use WinRAR (<http://www.rarlabs.com/>) to unpack the tar.gz files in Windows. You can use Notepad++ (<http://sourceforge.net/projects/notepad-plus/>) to edit the Bleeding rules in Windows to drop packets in inline mode, or you can edit them within Honeywall using WinSCP (<http://sourceforge.net/projects/winscp/>). When you connect to either Honeywall with WinSCP, you will need to set the Server response timeout setting above the default 15 seconds. On this dual processor machine we are employing the following Bleeding rulesets (modified to drop):

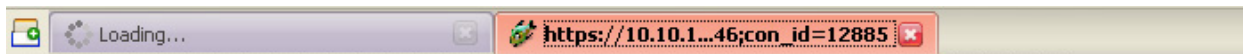
```
attack-responses.rules
bleeding-attack_response.rules
disallowed-ports.rules
bleeding.rules
finger.rules
bleeding-exploit.rules
community-exploit.rules
exploit.rules
bleeding-compromised.rules
bleeding-dshield.rules
bleeding-botcc.rules
bleeding-storm.rules
bleeding-drop.rules
bleeding-scan.rules
bleeding-virus.rules
bleeding-malware.rules
backdoor.rules
bleeding-web.rules
bleeding-p2p.rules
bleeding-voip.rules
bleeding-dos.rules
shellcode.rules1
mysql.rules
virus.rules
tftp.rules
dns.rules
icmp.rules
```

community-virus.rules
community-web-client.rules

¹ You will need to comment out one or more of the shellcode rules depending upon your router, DNS and NetBios configurations.

This bleeding configuration consumes 694 MB of RAM initially, which will rise with outbound LAN traffic. The Snort Inline process will utilize 6 times the amount of RAM that the Snort process does. If you apply the Bleeding rules to Snort, the firewall will utilize 802 MB of RAM initially. In both Honeywalls, Snort rules are treated separately from Snort Inline rules. Snort Inline rules drop packets while Snort rules flag and log packets. Operating two separate Snorts and two separate Snort Inlines provides the opportunity to pursue differing (and perhaps complimentary) detection and intrusion prevention strategies.

Honeywall's web interface (Walleye) includes many useful features. Those with physical access to the host machine are able to access Honeywall's web management interface through the use of a 3rd NIC. p0f fingerprints the attacker's operating system. Argus monitors flow, and Sebek captures process information. The Sebek honeypot affords some granularity in its configuration; you can enable "Roach Motel" mode, which prevents any outbound traffic from the Honeypot. To avoid potential issues with your ISP, I'd suggest that you set the honeypots to use the FVS-338's DNS proxy rather than those of the ISP's DNS servers. Walleye allows you to view packet captures or you may download them and perform analysis with Wireshark (<http://www.wireshark.org>) or Ethereal (<http://www.ethereal.com>).



```
05/04-06:57:03.047290 XX:XX:XX:XX . -> 0:14:6C:CB:2E:1C type:0x800 len:0x2C4
192.168.X.XXX:38513 -> 64.233.167.99:80 TCP TTL:64 TOS:0x0 ID:7281 IpLen:20 DgmLen:694
***AP*** Seq: 0x89B7DF51 Ack: 0x10E652F Win: 0x16D0 TcpLen: 20
47 45 54 20 2F 69 6D 61 67 65 73 2F 66 69 72 65 GET /images/fire
66 6F 78 2F 74 69 74 6C 65 2E 67 69 66 20 48 54 fox/title.gif HT
54 50 2F 31 2E 30 0D 0A 41 63 63 65 70 74 2D 45 TP/1.0..Accept-E
6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70 2C 69 64 ncoding: gzip,id
65 6E 74 69 74 79 0D 0A 48 6F 73 74 3A 20 77 77 entity..Host: ww
77 2E 67 6F 6F 67 6C 65 2E 63 6F 6D 0D 0A 55 73 w.google.com..Us
65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C er-Agent: Mozill
61 2F 35 2E 30 20 28 57 69 6E 64 6F 77 73 3B 20 a/5.0 (Windows;
55 3B 20 57 69 6E 64 6F 77 73 20 4E 54 20 35 2E U; Windows NT 5.
31 3B 20 65 6E 2D 55 53 3B 20 72 76 3A 31 2E 38 1; en-US; rv:1.8
2E 31 2E 33 29 20 47 65 63 6B 6F 2F 32 30 30 37 .1.3) Gecko/2007
30 33 30 39 20 46 69 72 65 66 6F 78 2F 32 2E 30 0309 Firefox/2.0
2E 30 2E 33 0D 0A 41 63 63 65 70 74 3A 20 69 6D .0.3..Accept: im
61 67 65 2F 70 6E 67 2C 2A 2F 2A 3B 71 3D 30 2E age/png,*/*;q=0.
35 0D 0A 41 63 63 65 70 74 2D 4C 61 6E 67 75 61 5..Accept-Langua
67 65 3A 20 65 6E 2D 75 73 2C 65 6E 3B 71 3D 30 ge: en-us,en;q=0
2E 35 0D 0A 41 63 63 65 70 74 2D 43 68 61 72 73 .5..Accept-Chars
65 74 3A 20 49 53 4F 2D 38 38 35 39 2D 31 2C 75 et: ISO-8859-1,u
74 66 2D 38 3B 71 3D 30 2E 37 2C 2A 3B 71 3D 30 tf-8;q=0.7,*,q=0
2E 37 2C 55 43 53 2D 32 3B 71 3D 30 2C 20 55 43 .7,UCS-2;q=0, UC
53 2D 34 3B 71 3D 30 2C 20 55 54 46 2D 31 3B 71 S-4;q=0, UTF-1;q
3D 30 0D 0A 50 72 61 67 6D 61 3A 20 6E 6F 2D 63 =0..Pragma: no-c
61 63 68 65 0D 0A 43 6F 6F 6B 69 65 3A 20 50 52 ache..Cookie: PR
45 46 3D 49 44 3D 30 61 39 35 34 38 34 36 34 36 EF=ID=0a95484646
61 35 61 63 66 35 3A 54 4D 3D 31 31 37 38 32 37 a5ac5:TM=117827
36 31 35 38 3A 4C 4D 3D 31 31 37 38 32 37 36 31 6158:LM=11782761
35 38 3A 53 3D 4D 48 44 6E 4A 76 6E 73 7A 6B 75 58:S=MHDnJvnszku
53 51 33 2D 30 0D 0A 52 65 66 65 72 65 72 3A 20 SQ3-0..Referer:
68 74 74 70 3A 2F 2F 77 77 2E 67 6F 6F 67 6C http://www.googl
65 2E 63 6F 6D 2F 66 69 72 65 66 6F 78 3F 63 6C e.com/firefox?cl
69 65 6E 74 3D 66 69 72 65 66 6F 78 2D 61 26 72 ient=firefox-a&r
6C 73 3D 6F 72 67 2E 6D 6F 7A 69 6C 6C 61 25 33 ls=org.mozilla%3
```

In Honeywall's web interface, Snort flagged packet captures are available for analysis

The Honeywall sample Snort Inline config file can be downloaded at docs.bleedingthreats.net. You will note that we are inspecting 100% of the packets (i.e. http_inspect_server's flow_depth is set to 0) and the inspection ports include all of the allowed instant messaging client ports. Instant messaging clients have become mini-browsers, and we treat them as such. In Smoothwall, we had difficulty with 100% inspection, and are only inspecting the first 1460 bytes in each packet.

Instant messaging clients have become mini-browsers, and we treat them as such.

To further reduce attack surface, you can write custom Snort rules to drop inbound traffic to specific destination port ranges. For example, this rule from our disallowed ports rules prevents remote desktop connections into your network, but will still allow you to use remote desktop out to a server beyond your firewalls:

```
drop tcp $EXTERNAL_NET any -> $HOME_NET 3375:3395 (msg:"TCP RDP PORTS 3375-3395 DISALLOWED";sid:100000765; )
```

There is a good firewall checklist available at SANS which will help you enhance your defenses by writing custom rules at:

<http://www.sans.org/score/checklists/FirewallChecklist.doc?portal=4f123a9b9aee81fd492e3fe888ab0531>

During the initial configuration, Honeywall asks if you wish to limit allowed ports out. The correct answer is yes. The ports allowed out are determined by inner LAN client software requirements (these ports should match those unblocked in the outer ring).

The screenshot shows the Honeywall web interface. At the top, there are browser tabs for https://10.10.1...46;con_id=12885 and https://10.10.10...0146;con_id=12885. Below the tabs, a section titled "Details for this flow" shows a flow from 192.168.x.x to 64.233.167.99 on May 4th 06:57:03. The flow is identified as TCP FIN with 38513 Linux packets. The destination is identified as http. Below this, an "IDS details" table shows a single entry with a priority of 3 and classification of "Misc activity". The name of the event is "INFO web bug 0x0 gif attempt". At the bottom, there is a "Flow Examination" section with links for "Packet Decode" and "Rule Evaluation".

Timestamp	Priority	Classification	Type	Name	Revision
May 4th 06:06:53	3	Misc activity		INFO web bug 0x0 gif attempt	2

Individual data flows flagged by Snort can be examined in Honeywall's web interface

One disadvantage of using Snort Inline in the center ring is that you do not get as many packet captures as when you run it on the outer ring. This may degrade your ability to detect initial reconnaissance attempts against your network. Honeywall has excellent packet capture logging, making it a nice tool for those desiring to write Bleeding signatures for new threats. If you deploy Honeywall on the outer ring, in order to maintain the rich experience on the client machines (i.e. Gmail and Yahoo mail), you will need to either 1) open up many more allowed ports out, (undesirable) or 2) manually configure port translation (which can require some time).

In between the Bleeding Snort machine and the Smoothwall you will need to deploy a switch with an MTU of 1500.

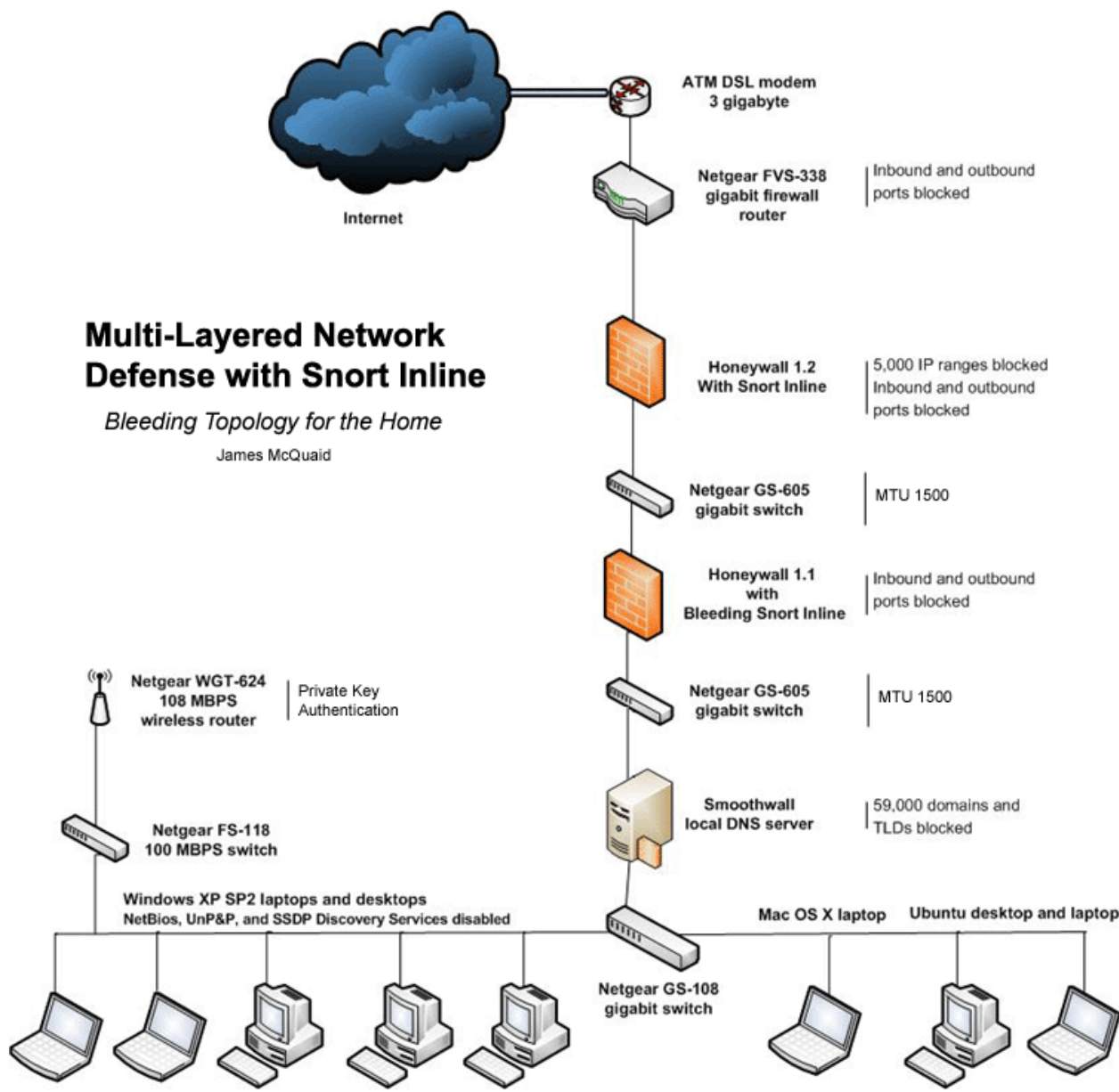
The inner ring of our layered home perimeter defense uses Smoothwall Express 2.0 with Fixes 1-9 (<http://www.smoothwall.org>) and several Smoothwall Homebrew Mods (<http://sourceforge.net/projects/smoothiemods/>) (including DNSMasq Update and DHCPD Update). The host hardware is a Pentium 2.4 GHz with 1 MB of RAM. Snort does not drop

packets here, it places alerts in a web accessible log file. Smoothwall will utilize all of the Bleeding Snort rulesets with the exception of the bleeding-botcc-BLOCK.rules and the bleeding-drop-BLOCK.rules, which require SnortSam. Smoothwall functions as an internal DNS server; significantly, you want to utilize **Bleeding's BlackHoleDNS project** (http://doc.bleedingthreats.net/bin/view/Main/AllProjects#BlackHoleDNS_for_Spyware) with it. We have a short list of TLDs blocked along with 63,000 hostile domains in blackhole.conf. I modified an open source perl script which removes duplicates and sorts the domains alphabetically. If you participate in the **Spyware Listening Post project** (http://doc.bleedingthreats.net/bin/view/Main/AllProjects#Spyware_Listening_Post), you can have your blackhole.conf file updated automatically. In Smoothwall's Squid ACL files, you should set safe ports equal to those configured in Honeywall's allowed ports out list. Limit Smoothwall's client-facing IP range to the number of client machines and wireless devices you have. ***The Bleeding rulesets provide early warning of an infected client machine*** (frequently before the anti-virus vendors have developed the relevant signatures). You can run Snort on Smoothwall in an ultra-sensitive configuration on the inner ring because the Netgear appliance and the Honeywalls will have already dropped the inbound bad packets. Although you will see Google's web bugs (1 pixel gif files) and other traffic in the IDS logs, Smoothwall's firewall logs should usually be empty. A word of caution, if either the outer or center ring fails, you will need to quickly reconfigure Snort, or Smoothwall will very rapidly exhaust its available memory and be subject to attack. Configure Smoothwall's client machine-facing NIC with a different subnet than that facing the Netgear. Block the DSL modem's IP range and the two Honeypot IP addresses in Smoothwall.

In between the Smoothwall and the user desktops machines, we use a Netgear GS-108 switch. This switch's power cord is plugged into an inexpensive, analog timer (commonly used for electric lamps). The timer turns off the switch during the night, isolating the client machines from one another during their scheduled anti-virus and anti-spyware scans.

Behind the inner ring, the demographics of the machine population vary, but are normally half Windows and half non-Windows (OS X, CentOS, Ubuntu and SE Fedora). All of them are limited to resolving DNS to the Smoothwall DNS server. The XP machines have multiple security apps installed to compensate for the OS. These include Avira (<http://www.avira.com/en/pages/index.php>) Antivir PersonalEdition Classic, for protection against zero day viruses, and avast! Professional antivirus both are installed on each XP. Most avast users don't properly use the software's web scanning feature: this requires that browser proxies be set to 127.0.0.1 on port 12080. F-Prot is another anti-virus client in use. We have not been able to run Antivir and F-Prot on the same box. Sunbelt's Counterspy and Lavasoft's Ad-Aware Plus provide the XP machines a measure of realtime protection, and allow scheduled anti-spyware scanning. ZoneLabs' IMSecure Pro allows somewhat safer instant messaging configurations; it allows content type filtering and prevents address, telephone, and other personal data outbound via IM (parents should consider installing it). Although it has been said that "**all desktop firewalls are made of straw**", they remain a vital component in stopping the spread of a virus once a machine on your LAN has been infected. Comodo's (<http://www.comodo.com/>) Firewall Pro has been tested and provides the best information leak protection possible. Sunbelt's Kerio Personal Firewall (driver 4.3.142) provides Snort on the XP

desktop. Kerio Personal Firewall will accept the Bleeding rulesets (they are located in the rlk files in C:\Program Files\Sunbelt Software\Personal Firewall\Config\IDSRules). Kerio drops offending packets when a Bleeding rule is tripped. Web pages will render even as packets are dropped. Use Kerio's packet filtering to prevent client machines from direct communication with the IP address of the firewall/router device in the outer ring. You will need client machines with 64-bit processors and a minimum of 1 GB of RAM to run all of this software on Windows XP. Each security software package is scheduled to run a deep scan at 24 hour intervals. These scans can take hours, must not overlap, or occur when the machines are in use. The Linux desktops require less RAM and perform quite well with 32-bit processors.



At our home, we've been able to substantially reduce malware infections and intrusions using this multiple layers of snort topology. My older teens can more safely instant message, watch streaming video, play games online and so forth; in contrast, Microsoft's ISA Server does not permit this much end user functionality. During the past year, since adopting a multi-layered topology, we've had only one infection (I always reload an infected machine due to the risk of a hidden payload). A few years ago, I was using a double NIC Microsoft Small Business Server 2003 (with RRAS) in tandem with a perimeter firewall appliance, it was impossible to keep worms from traversing SBS's multitude of shares. SBS also had compatibility problems with both Sunbelt's Kerio Personal Firewall and Agnitum's Outpost firewall. Despite having fully patched Windows XP machines with up-to-date anti-virus, I was reloading an XP machine per month. I had read about **Bleeding's BlackHoleDNS project**, and when I couldn't get it to function in SBS, I reloaded that box with Smoothwall.

Your ability to use Smoothwall in an ultra-sensitized configuration will be affected by your network topology as well as, bandwidth usage. If you are configuring snort for use in a home network or small office, you can operate with higher sensitivity than you could in a production environment or on the perimeter of a large organization. In a larger organization, you will benefit from extensive segmentation of your network using snort inline, and by employing Bleeding's BlackHoleDNS project. Depending upon your domain's requirements, you may be able to use Smoothwall as well. Whether in a large or small environment, there are several snort.conf preprocessor settings that you will need to fine tune. Because its primary purpose is early warning of a problem on the LAN, you want Snort on Smoothwall (in the inner ring) to use as much of the available RAM as possible without risking memory exhaustion, performance problems or instability. *This is not a set and forget network, you should regularly review the logs from the firewalls as well as, your client machines, and then take appropriate action.*

Snort Pre-Processor Configurations:

Below elements of the default configuration are contrasted with that we're using in Smoothwall. The Honeywall sample Snort Inline config file can be downloaded at docs.bleedingthreats.net.

Preprocessor: flow

Purpose: the Flow tracking module is meant to start unifying the state keeping mechanisms of Snort.

Our Smoothwall config: preprocessor flow: memcap 100663296, rows 8198, stats_interval 0 hash 2

Default config: preprocessor flow: stats_interval 0 hash 2

memcap: the number of bytes to allocate

Smoothwall Caveats: the number of rows in the hash table can be increased to enhance performance; increases will require additional RAM.

Preprocessor: Frag 3

Purpose: the frag3 preprocessor is a target-based IP defragmentation module.

Frag 3 Global Configuration:

Our Smoothwall config: preprocessor frag3_global: memcap 67108864, max_frags 131072

Default config: preprocessor frag3_global: prealloc_nodes 8192

Frag 3 Engine Configuration:

Our Smoothwall config: preprocessor frag3_engine: policy linux detect_anomalies bind_to (outer perimeter CIDR)

Our Smoothwall config: preprocessor frag3_engine: policy first detect_anomalies bind_to (inner perimeter CIDR)

Our Smoothwall config: preprocessor frag3_engine: policy last detect_anomalies

Default config: preprocessor frag3_engine: policy first detect_anomalies

memcap: memory cap for self-preservation (default is 4 MB).

max_frags: maximum simultaneous fragments to track (default is 8192).

Smoothwall caveats: Be certain to check /var/smoothwall/messages/ to confirm the actual configuration after restarting Snort. Smoothwall will ignore your snort.conf file in certain circumstances.

Preprocessor: Stream 4

Purpose: Stream4 provides TCP stream reassembly and stateful analysis capabilities.

Our Smoothwall config: preprocessor stream4: detect_scans, detect_state_problems,
disable_evasion_alerts, state_protection, memcap 33608864

Default config: preprocessor stream4: detect_scans, disable_evasion_alerts, memcap 8388608

Stream4 Options:

preprocessor stream4 default settings:

session timeout (timeout) 30 seconds

session memory cap (memcap) 8388608 bytes

stateful inspection (noinspect) active (noinspect disabled)

stream stats (keepstats) inactive

state problem alerts (detect_state_problems) inactive (detect_state_problems disabled)

evasion alerts (disable_evasion_alerts) inactive (disable_evasion_alerts enabled)

asynchronous link (asynchronous_link) inactive

log flushed streams (log_flushed_streams) inactive

max TCP sessions (max_sessions) 8192

session cache purge (cache_clean_sessions) 5

self preservation threshold (self_preservation_threshold) 50 sessions/sec

self preservation period (self_preservation_period) 90 seconds

suspend threshold (suspend_threshold) 200 sessions/sec

suspend period (suspend_period) 30 seconds

state protection (state_protection) inactive

server inspect limit (server_inspect_limit) -1 (inactive)

UDP session tracking (enable_udp_sessions) inactive

max UDP sessions (max_udp_sessions) 8192

stream4_reassemble Configuration:

Our Smoothwall config: preprocessor stream4_reassemble: both, favor_new, ports: all,
emergency_ports 21 23 25 42 53 80 110 111 135 136 137 139 143 222 445 513 1433 1521
3306

Default config: preprocessor stream4_reassemble: client

stream4_reassemble Options:

preprocessor stream4_reassemble default settings:

reassemble client (clientonly) active

reassemble server (serveronly) inactive

reassemble both (both) inactive

reassemble ports (ports) 21 23 25 42 53 80 110 111 135 136 137 139 143 445 513 1433
1521 3306

emergency reassemble ports (ports) 21 23 25 42 53 80 110 111 135 136 137 139 143 445 513
1433 1521 3306

reassemble alerts (noalerts) active (noalerts disabled)

favor old packet (favor_old) active

favor new packet (favor_new) inactive

flush on alert (flush_on_alert) inactive

overlap limit (overlap_limit) -1 (inactive)

large packet performance (large_packet_performance) inactive

Smoothwall caveats: If you configure stream 4 to be overly sensitive, you can seriously flood your Snort logs. If you under configure it, you won't see events that you might want to be aware of. Over time, you will want to experiment with most of the configuration options.

Preprocessor: sfportscan

Purpose: The sfPortscan module, developed by Sourcefire, is designed to detect the first phase in a network attack

```
Smoothwall config:  preprocessor sfportscan: proto { all } \  
                   scan_type { all } \  
                   memcap { 67108864 } \  
                   sense_level { high } watch_ip { xxx.xxx.x.x, xxx.xxx.x.x }
```

```
Default config: preprocessor sfportscan: proto { all } \  
               scan_type { all } \  
               sense_level { low }
```

sense_level available options:

Low: Low alerts are only generated on error packets sent from the target host, and because of the nature of error responses, this setting should see very few false positives. However, this setting will never trigger a Filtered Scan alert because of a lack of error responses. This setting is based on a static time window of 60 seconds, after which this window is reset.

Medium: Medium alerts track connection counts, and so will generate filtered scan alerts. This setting may false positive on active hosts (NATs, proxies, DNS caches, etc), so the user may need to deploy the use of Ignore directives to properly tune this directive.

High: High alerts continuously track hosts on a network using a time window to evaluate portscan statistics for that host. A "High" setting will catch some slow scans because of the continuous monitoring, but is very sensitive to active hosts. This will require you to tune sfPortscan.

watch_ip:

Defines which IPs, networks, and specific ports on those hosts to watch. The list is a comma separated list of IP addresses and/or an IP address using CIDR notation. Optionally, ports are specified after the IP address/CIDR using a colon and can be either a single port or a range denoted by a dash. IPs or networks not falling into this range are ignored if this option is used.

Smoothwall caveats: Filtered port scan alert types are more likely than other alert types to be false positives. Yahoo and Google will generate false positives; take notice of other alerts. Generally, the alerts from sfportscan will require that you regularly do lookups of IP addresses which appear in your logs.

Preprocessor: preprocessor http inspect

Smoothwall default: preprocessor http_inspect: global iis_unicode_map
/var/smoothwall/snort/unicode.map 1252

Our Smoothwall: preprocessor http_inspect: global detect_anomalous_servers iis_unicode_map
/var/smoothwall/snort/unicode.map 1252

http_inspect_server Options:

Smoothwall default: preprocessor http_inspect_server: server default profile all ports { 80 }

Our Smoothwall: preprocessor http_inspect_server: server \$HOME_NET profile all ports { 80
1863 3128 5050 5061 5190 5191 5192 5193 5222 5223 6891 8080 8180 13324 13325 32771
56885 } oversize_dir_length 300 flow_depth 1460

Our Smoothwall: preprocessor http_inspect_server: server default profile all ports { 80 1863
3128 5050 5061 5190 5191 5192 5193 5222 5223 6891 8080 8180 13324 13325 32771 56885
} oversize_dir_length 300 flow_depth 1460

Option: oversize_dir_length

This option takes a non-zero positive integer as an argument. The argument specifies the max char directory length for URL directory. If a URL directory is larger than this integer, an alert is generated.

Option: ports

This is how the user configures which ports to decode on the HTTP server. Encrypted traffic (SSL) cannot be decoded, so adding port 443 will only yield encoding false positives.

Option: flow_depth

Purpose: this specifies the amount of server response payload to inspect. Specifying a `flow_depth` of 300 (the default) increases throughput speed because it ignores a large part of the network traffic (HTTP server response payloads). The HTTP headers of friendly packets are usually under 300 bytes in length.

The `flow_depth` value can be set from -1 to 1460. A value of -1 causes Snort to ignore all server side traffic for ports defined in ports. A value of 0 causes Snort to inspect all HTTP server payloads defined in ports (this will slow down throughput speed). Values above 0 tell Snort the number of bytes to inspect in the first packet of the server response.