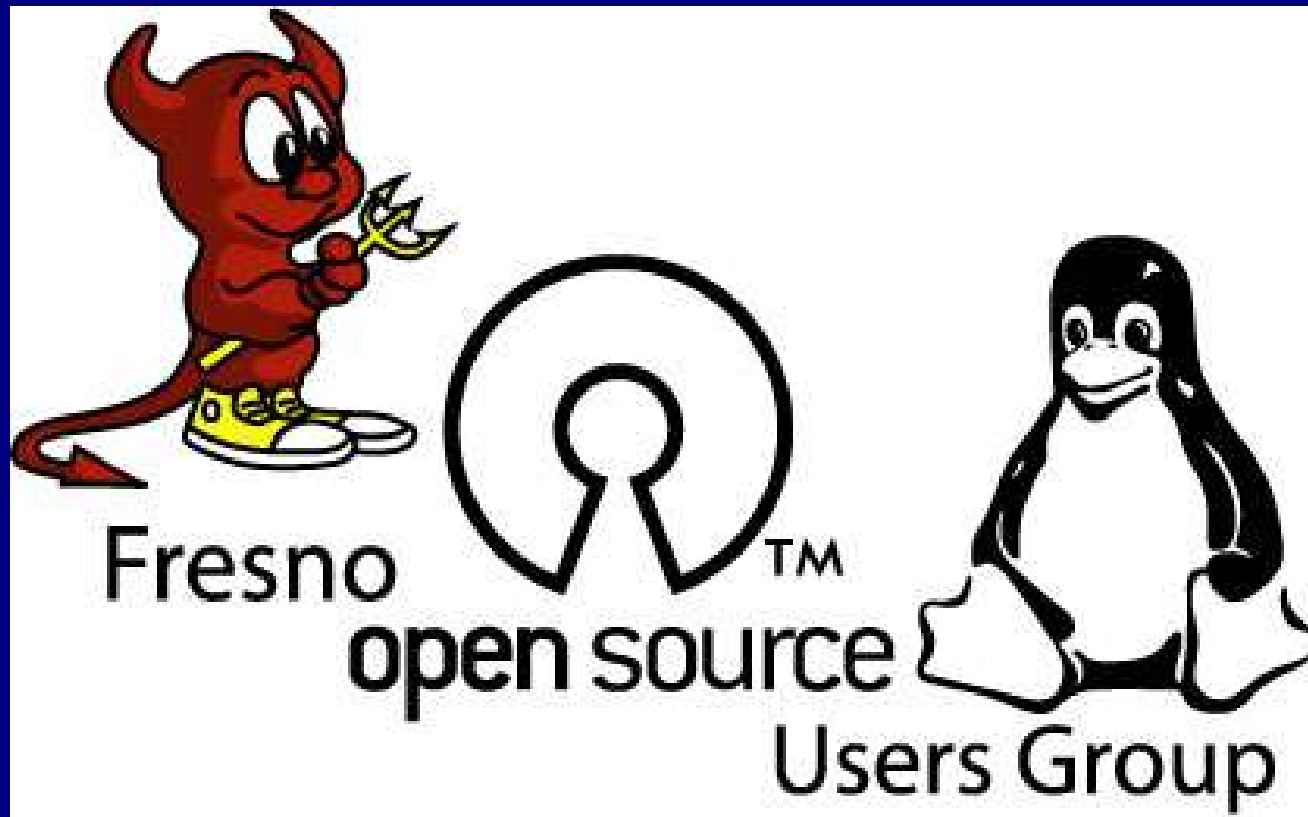
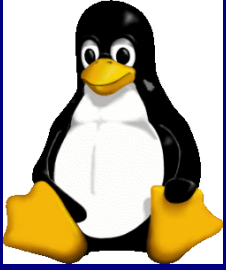


Welcome Fresno Open Source Users Group



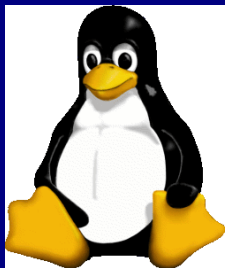
About Me

- **My name is Steven Hollingsworth**
 - ★ I am a Network Administrator at Howe Electric Inc.
 - ★ My Email is stevodestructo@gmail.com
 - Feel free to email me and ask me questions anytime
 - ★ I've been using Linux for about 3 years
 - I'm still a n00b [i.e. next to Robert Nickle]
 - ★ FOSUG has been a great place to meet wonderful friends. [lemon chicken... mmmm]



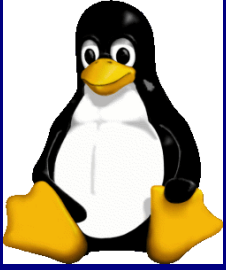
Helps if you know...

- **These aren't necessary but might help a bit.**
 - ★ Basic understanding of Unix type operating systems.
 - ★ Some Linux Kernel Basics
 - i.e. How to compile a kernel with 'make menuconfig'
 - Drew covered this in the last meeting
 - ★ Basic Networking Concepts
 - What an IP address is [192.168.0.1]
 - What a subnet mask is [i.e. 255.255.255.0 or /24]
 - What a port is [80/www , 22/ssh, etc..] and what services are “usually” hooked up to them.
 - ★ How to execute commands, and basic bash scripting
 - For the init script and iptables setup



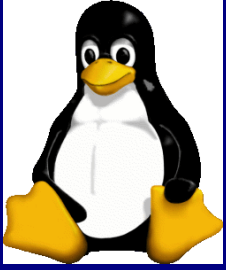
A Few Notes

- **I will be using examples from Gentoo exclusively.**
 - ★ Any examples of installing packages will be from gentoo's emerge unless otherwise noted
 - <http://packages.gentoo.org>
 - ★ why? Because it's the Linux Distro I use exclusively.
 - (But it really doesn't matter because they are all pretty much the same besides package management)
 - ★ <http://www.gentoo.org>



Firewall Basics

The Big Bad Internet... Is just a “hop” away



Types of Firewalls

- **Packet Filter**

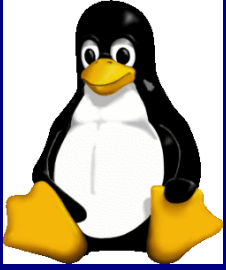
- ★ Only allows or blocks traffic by IP Address or Port Numbers

- ★ Example: ipfwadm(Linux <2.0) ipchains (Linux <2.4)

- **Stateful Packet Filters**

- ★ Has Packet Filter ability with the ability to keep track of all packet states and act accordingly

- ★ Example OpenBSD pf, Netfilter (Linux >2.4)



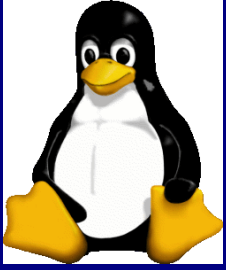
Types of Firewalls (cont.)

🌐 Proxies

- ★ Separate from Firewall, specializes in a specific protocol and protects against specialized attacks. Looks inside data of packet and makes decisions based on that information.
- ★ Examples: Squid, Postfix, ftp-proxy

🌐 Hybrid Firewalls

- ★ Combines Stateful Packet Filters with Proxy like behaviour for maximum control and protection to internal network.
- ★ Examples: Watchguard Firebox, Cisco Pix
- ★ Iptables + Snort make this jump



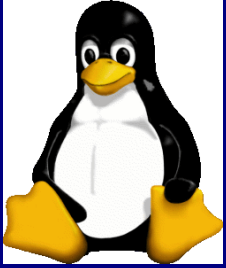
Physical Firewall Types

- **Local**

- ★ Only protects local machine and services on local machine
- ★ Only need one Interface

- **Network Address Translation (NAT)**

- ★ Passes traffic from one (public) interface to a (private) internal interface. Allowing one to have multiple hosts behind a firewall seem as one
- ★ Need two interfaces (public/private)

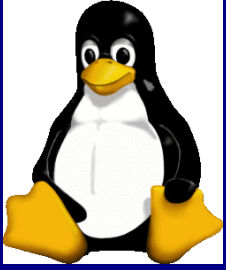


Physical Firewall Types

• Bridge

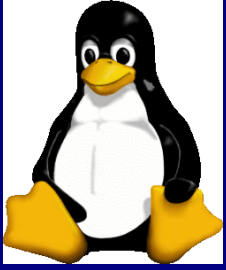
- ★ Transparently passes traffic between two interfaces
- ★ Does not directly effect Network Topology
- ★ Needs three Interfaces (managing interface<ipaddr>, internal bridge interface, external bridge interface)
- ★ Most often used as a parameter firewall protecting DMZ hosts.

• **I will draw these on the whiteboard if anybody wants me to.**



Iptables

Otherwise known as NetFilter



Previous Versions of the Linux Firewall

- **Linux Kernel ≤ 2.0**

- ★ IPFWADM

- Strictly a packet filter, with very basic structure

- **Kernel 2.2**

- ★ IPCHAINS

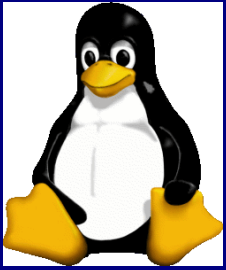
- Still at basic packet filter, but with much improved interface and structure

- **Kernel 2.4 – 2.6**

- ★ IPTABLES

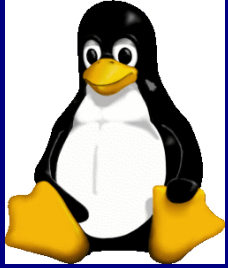
- This is what this part of the presentation will be on. A stateful packet filter, with a solid structure and many many capabilities.

- Created by Rusty Russell



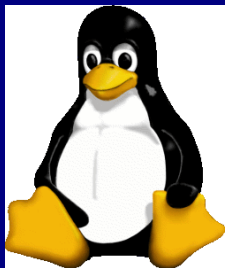
Iptables and the Linux Kernel

- **There are a lot of cool modules in the iptables that are not in the standard kernel you can patch and compile into your kernel.**
 - ★ See the netfilter patch-o-matic
 - <http://www.netfilter.org/downloads.html>
 - There are some really cool patches that do some wild things... like tarpit , string matching, and port scan detection built right into the kernel
 - ★ You can Also build Qos (Quality of Service into the Kernel)
 - This allows bandwidth throttling, allowing some ports, ip's and such more bandwidth than others)



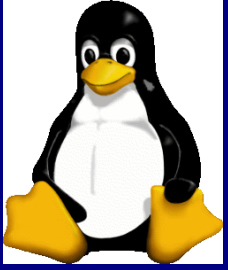
2.4 Kernel

- **If you want special Iptables Kernel Modules be sure to patch the kernel with netfilters Patch-O-Matic.**
- **Using Make Menuconfig**
 - ★ Networking Options
 - check Network packet filtering
 - Check 802.1d if you are going to be running as a bridge
 - IP Netfilter Configuration
 - Check all appropriate modules for firewall configuration (standalone, NAT, Bridge)



2.6 Kernel

- **Patch-O-Matic (if desired)**
- **Using Make Menuconfig**
 - ★ Device Drivers
 - ➔ Networking Support
 - Networking Options
 - ★ Network Packet Filtering
 - ➔ IP Netfilter Configuration
 - ➔ Bridge Network Configuration
 - 802.1d Bridge configuration



The Iptables Userspace (Binary || Program)

• **emerge iptables**

★ this will give you /sbin/iptables

★ Iptables userspace binary help

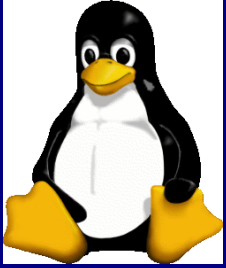
→ use the command 'iptables -help' or

→ see these man pages

• iptables(8)

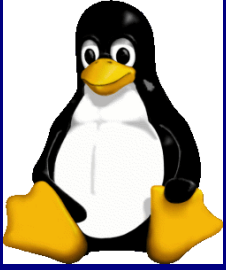
• iptables-save(8)

• iptables-restore(8)



The tables of Iptables

- **On the command line these are specifically selected with iptables -t <table>**
 - ★ If the -t option is left off it assumes the tables being altered is the filter table
 - ★ **FILTER**
 - **INPUT** chain
 - When the packet first hits an interface
 - **OUTPUT** chain
 - Whether or not it lets the packet leave
 - **FORWARD** chain
 - Where and if it passes the packet through the machine to another interface



The tables of Iptables (cont.)

- **NAT (Network Address Translation)**

- ★ PREROUTING

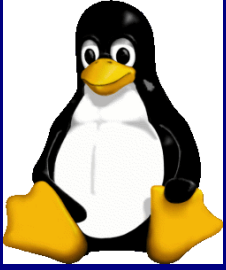
- for altering packets as soon as they come in

- ★ OUTPUT

- for altering locally-generated packets before routing

- ★ POSTROUTING

- for altering packets as they are about to go out



The tables of Iptables (cont.)

- **MANGLE (specialized packet manipulation)**

- ★ PREROUTING

- for altering incoming packets before routing

- ★ OUTPUT

- for altering locally-generated packets before routing).

- ★ INPUT

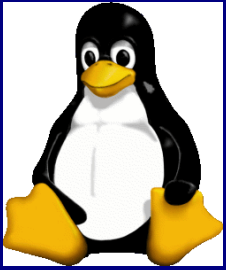
- for packets coming into the box itself

- ★ FORWARD

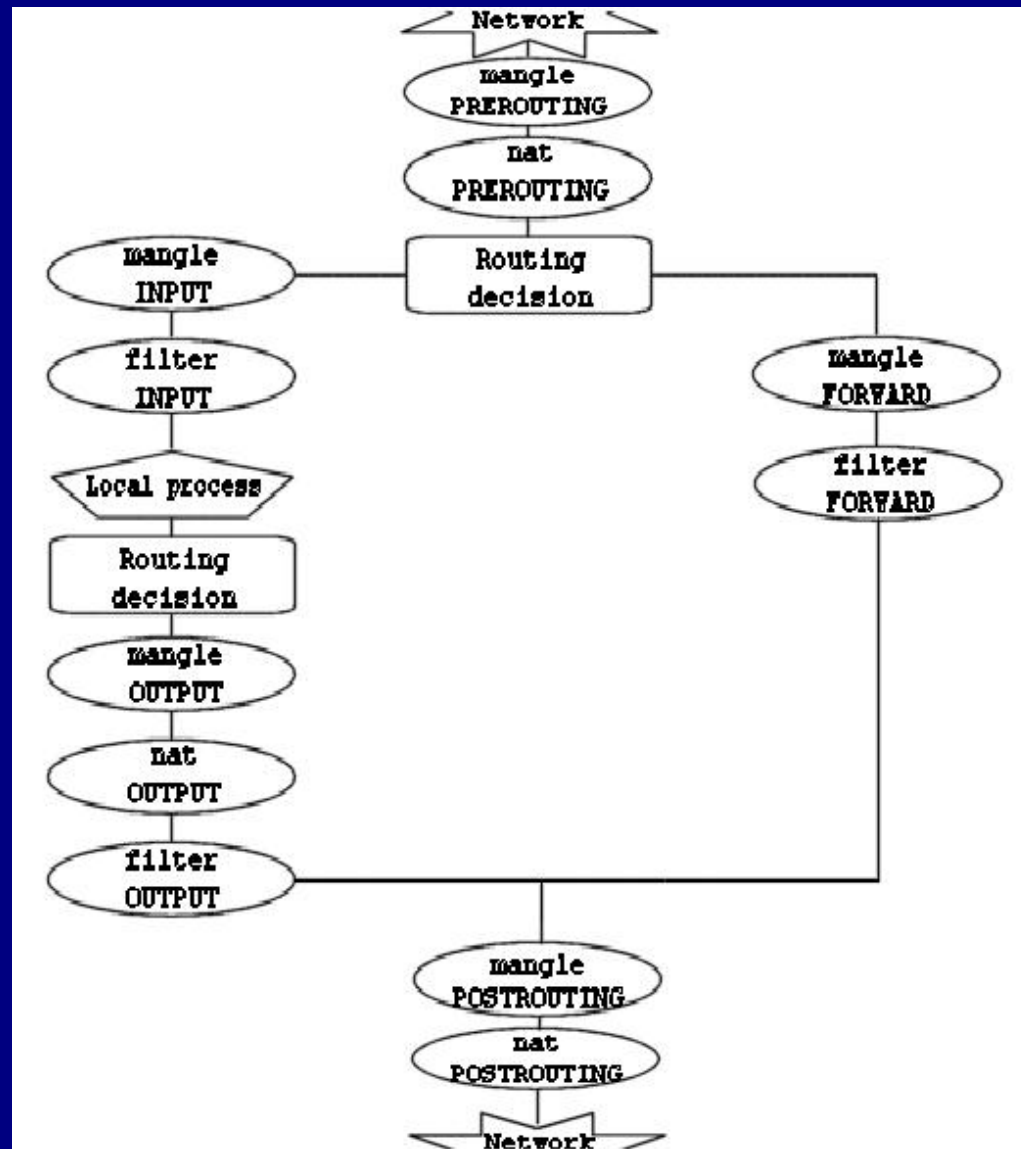
- for altering packets being routed through the box

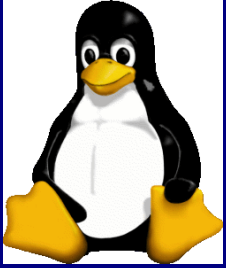
- ★ POSTROUTING

- for altering packets as they are about to go out



Graphic





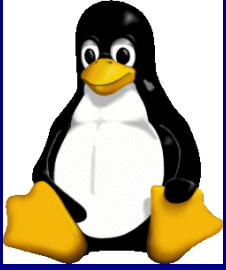
The flow

• **Setting Default Policies**

- ★ iptables -P INPUT DROP
- ★ iptables -P OUTPUT DROP
- ★ iptables -P FORWARD DROP

• **Creating Chains (in the filter table)**

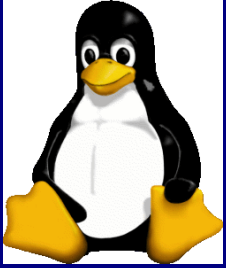
- ★ This is where the real magic happens and gives you the power to structure the way you handle iptables by creating custom chains in the filter table.
 - ➔ iptables -N tcp_bad_packets
 - ➔ iptables -N allowed_trusted
 - etc... etc... whatever way you think you need to group, structure or organize sections of how your firewall is layed out.



The Flow (cont.)

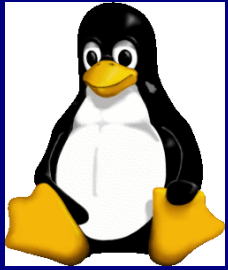
🔴 The iptables command put together

- ★ iptables
- ★ -t [nat, mangle] if not filter
- ★ -A [chain target]
 - ➔ input, output, forward, or custom chain
- ★ -p [tcp,udp,ICMP]
- ★ -i interface [eth0, eth1] etc...
- ★ -s [source address/subnet]
- ★ -d [dest. address/subnet]
- ★ -sport [source port]
- ★ -dport [dest. port]
- ★ -j [target]
 - ➔ DROP, REJECT, ACCEPT, QUEUE, RETURN, Custom Chain
- ★ There are many many more options, these are just the basics.



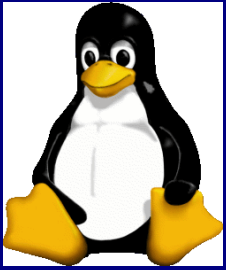
Examples

- **iptables -A INPUT -i eth0 -s 192.168.1.5 -j REJECT**
- **iptables -A FORWARD -i eth0 -d 10.10.20.2 -j ACCEPT**
- **iptables -A INPUT -i eth0 -d 192.168.1.10 -j trusted_hosts**
- **iptables -A trusted_hosts -s 192.168.2.20 -j ACCEPT**



The INIT Script

- **Make a file like `/etc/init.d/myiptables` for the init script.**
 - ★ After the file is edited / copied issue the command to set it to boot up.
 - `rc-update add myiptables default`
 - ★ You can add other targets to suit you needs to the script... see next.
 - ★ If you have an EXTREMELY Large or complex ruleset see `iptables-save(8)` and `iptables-restore(8)`



The skeleton of an init script

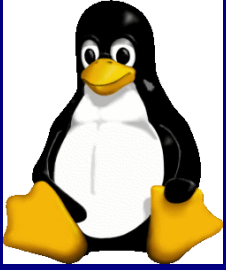
```
#!/sbin/runscript

opts="start stop"

depend() {
    need net logger
    before sshd
}

start() {
    echo 1 > /proc/sys/net/ipv4/ip_forward #for NAT
    ebegin "Starting Firewall"
        /etc/myiptables.sh or /etc/myiptables.pl
    eend $?
}

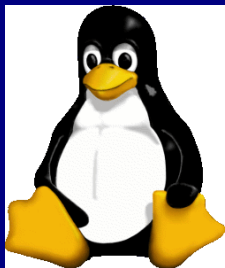
stop() {
    <shell commands>
    ebegin "Stopping firewall"
        $IPTABLES -F
        $IPTABLES -t nat -F
        $IPTABLES -t mangle -F
        $IPTABLES -X
        $IPTABLES -t nat -X
        $IPTABLES -t mangle -X
    eend $?
}
```



The iptables script

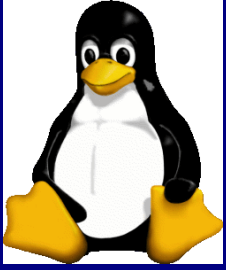
• The script Structure

- ★ Setting variables for hosts, interfaces and networks
 - `IFACE="eth0"`
- ★ Probing the kernel for modules that may be used in the script
 - `/sbin/depmod -a`
 - `/sbin/modprobe ip_conntrack`
- ★ Set options in `proc` as needed for custom stuff
 - `echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`
 - disabled broadcasts
- ★ Set Default Policies (i.e. `iptables -P INPUT DROP`)
- ★ Continue with the rest of script



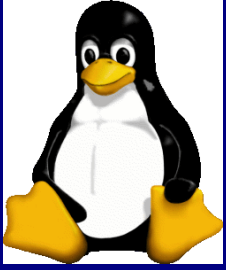
Wrapping it up

- **You can see a sample home firewall script at the downloads section of the fosug site from this meeting.**
 - ★ <http://www.fosug.org>
- **There will also be one posted at the meeting TBA.**
- **Lots of good tutorials and howtos here**
 - ★ <http://www.netfilter.org/documentation/>
 - ★



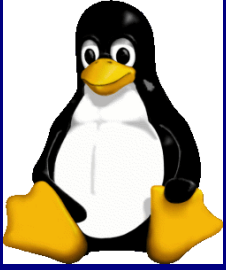
SNORT

It sucks more...



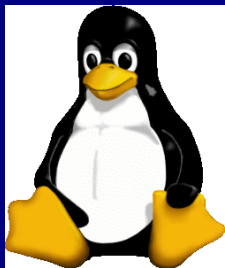
History of Snort

- **Has been around since 1998**
- **Is used on a a lot of difference OS's including Windows and Mac OS X**
- **It's called "The PIG" for obvious reasons**
- **Created and maintained by Marty Roesch**



What it is

- **It is basically a permissive packet capturing utility that has the ability to dump packets to screen or use a set of rules to analyse the contents of the packet and log it if it matches a rule.**
 - ★ No it doesn't sleep around
 - ★ Basic command line options:
 - If you wanted to dump all traffic passing too and from eth0
 - `snort -vdCi eth0`
 - `snort -vdCi eth0 src or dst 192.168.1.2 and src port 80`
 - `snort -vdCi bridge0 src 192.168.1.2 and dst port 21`
 - there are many more command line options type '`snort -help`' for the userspace switches and what they do.



Packet Capture (No Hex)

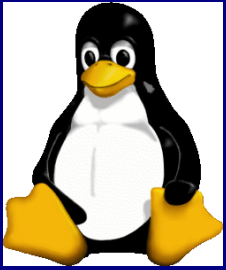
=====
=====

```
09/18-08:51:14.769350 67.182.27.192:54728 -> 63.202.57.170:80
TCP TTL:44 TOS:0x0 ID:48423 IpLen:20 DgmLen:52 DF
***A*** Seq: 0x81775DCD Ack: 0xD1C6FD50 Win: 0xB500 TcpLen: 32
TCP Options (3) => NOP NOP TS: 54060008 499904639
```

=====
=====

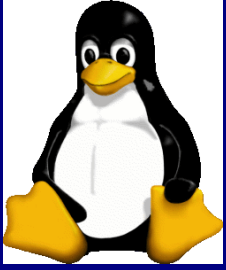
```
09/18-08:51:14.777298 67.182.27.192:54728 -> 63.202.57.170:80
TCP TTL:44 TOS:0x0 ID:48424 IpLen:20 DgmLen:459 DF
***AP*** Seq: 0x81775DCD Ack: 0xD1C6FD50 Win: 0xB500 TcpLen: 32
TCP Options (3) => NOP NOP TS: 54060008 499904639
GET /nav/home-1.gif HTTP/1.1..Host: www.howe-electric.com..User-
Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/200
40520 Firefox/0.8..Accept: image/png,image/jpeg,image/gif;q=0.2,
*/*;q=0.1..Accept-Language: en-us,en;q=0.5..Accept-Encoding: gzi
p,deflate..Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7..Keep-
Alive: 300..Connection: keep-alive..Referer: http://www.howe-ele
ctric.com/home.html....
```

=====
=====



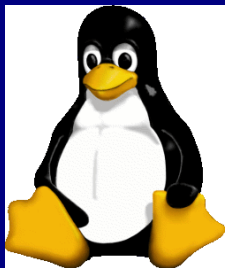
Packet Capture with Hex

```
=====  
09/18-08:58:18.503784 67.182.27.192:65426 -> 63.202.57.170:80  
TCP TTL:44 TOS:0x0 ID:24288 IpLen:20 DgmLen:459 DF  
***AP*** Seq: 0x9B6F059E Ack: 0xEB9453B5 Win: 0x16D0 TcpLen: 32  
TCP Options (3) => NOP NOP TS: 54483814 499947047  
47 45 54 20 2F 6E 61 76 2F 68 6F 6D 65 2D 31 2E GET /nav/home-1.  
67 69 66 20 48 54 54 50 2F 31 2E 31 0D 0A 48 6F gif HTTP/1.1..Ho  
73 74 3A 20 77 77 77 2E 68 6F 77 65 2D 65 6C 65 st: www.howe-ele  
63 74 72 69 63 2E 63 6F 6D 0D 0A 55 73 65 72 2D ctric.com..User-  
41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35 Agent: Mozilla/5  
2E 30 20 28 58 31 31 3B 20 55 3B 20 4C 69 6E 75 .0 (X11; U; Linu  
78 20 69 36 38 36 3B 20 65 6E 2D 55 53 3B 20 72 x i686; en-US; r  
76 3A 31 2E 36 29 20 47 65 63 6B 6F 2F 32 30 30 v:1.6) Gecko/200  
34 30 35 32 30 20 46 69 72 65 66 6F 78 2F 30 2E 40520 Firefox/0.  
38 0D 0A 41 63 63 65 70 74 3A 20 69 6D 61 67 65 8..Accept: image  
2F 70 6E 67 2C 69 6D 61 67 65 2F 6A 70 65 67 2C /png,image/jpeg,  
69 6D 61 67 65 2F 67 69 66 3B 71 3D 30 2E 32 2C image/gif;q=0.2,  
2A 2F 2A 3B 71 3D 30 2E 31 0D 0A 41 63 63 65 70 */*;q=0.1..Accep  
74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E 2D 75 t-Language: en-u  
73 2C 65 6E 3B 71 3D 30 2E 35 0D 0A 41 63 63 65 s,en;q=0.5..Acce  
70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A 69 pt-Encoding: gzi  
70 2C 64 65 66 6C 61 74 65 0D 0A 41 63 63 65 70 p,deflate..Accep  
74 2D 43 68 61 72 73 65 74 3A 20 49 53 4F 2D 38 t-Charset: ISO-8  
38 35 39 2D 31 2C 75 74 66 2D 38 3B 71 3D 30 2E 859-1,utf-8;q=0.  
37 2C 2A 3B 71 3D 30 2E 37 0D 0A 4B 65 65 70 2D 7,*;q=0.7..Keep-  
41 6C 69 76 65 3A 20 33 30 30 0D 0A 43 6F 6E 6E Alive: 300..Conn  
65 63 74 69 6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 ection: keep-ali  
76 65 0D 0A 52 65 66 65 72 65 72 3A 20 68 74 74 ve..Referer: htt  
70 3A 2F 2F 77 77 77 2E 68 6F 77 65 2D 65 6C 65 p://www.howe-ele  
63 74 72 69 63 2E 63 6F 6D 2F 68 6F 6D 65 2E 68 ctric.com/home.h  
74 6D 6C 0D 0A 0D 0A tml....
```

Daemon Mode Snort

- **Edit /etc/snort/snort.conf**
 - ★ the base install has a file called /etc/snort/snort.conf.distrib that is a sample file, this is usually a good place to start to see how things are layed out.
- **Logs to /var/snort (default)**
 - ★ can be changed in snort.conf
- **Snort init script /etc/init.d/snort**
- **Snort has pre defined rules that tell you when people are doing bad things on your network**
 - ★ You can make your own custom rules



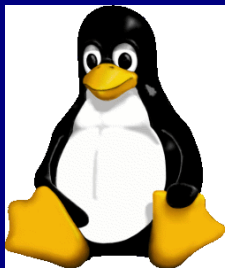
Snort and Snort Packages (Gentoo)

- * net-analyzer/snort
 - Latest version available: 2.1.3
 - Latest version installed: 2.1.3
 - Size of downloaded files: 2,352 kB
 - Homepage: <http://www.snort.org/>
 - Description: Libpcap-based packet sniffer/logger/lightweight IDS
 - License: GPL-2

- * net-analyzer/snortalog
 - Latest version available: 2.2.1
 - Latest version installed: [Not Installed]
 - Size of downloaded files: 413 kB
 - Homepage: <http://jeremy.chartier.free.fr/snortalog/>
 - Description: a powerful perl script that summarizes snort logs

- * net-analyzer/snortsam
 - Latest version available: 2.24
 - Latest version installed: [Not Installed]
 - Size of downloaded files: 426 kB
 - Homepage: <http://www.snortsam.net/>
 - Description: Snort plugin that allows automated blocking of IP addresses on several firewalls
 - License: as-is

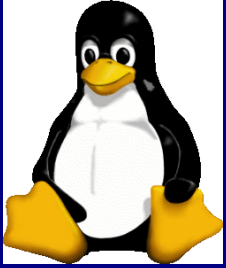
- * net-analyzer/snortsnarf
 - Latest version available: 021111.1-r1
 - Latest version installed: [Not Installed]
 - Size of downloaded files: 140 kB
 - Homepage: <http://www.silicondefense.com/software/snortsnarf/>
 - Description: Snort Snarf parses Snort log files, and converts them into easy-to-read HTML files.
 - License: GPL-2



Snort.conf

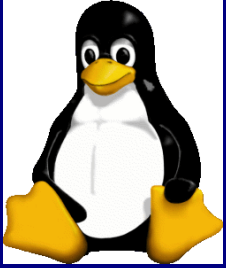
• Set Variables

- ★ var HOME_NET 192.168.1.0/24
- ★ var EXTERNAL_NET any
- ★ var DNS_SERVERS 192.168.1.77/32
- ★ var SMTP_SERVERS [192.168.1.55/32,192.168.1.33/32]
- ★ var HTTP_SERVERS 192.168.2.0/24
- ★ var SQL_SERVERS \$HOME_NET
- ★ var TELNET_SERVERS \$HOME_NET
- ★ var SNMP_SERVERS \$HOME_NET



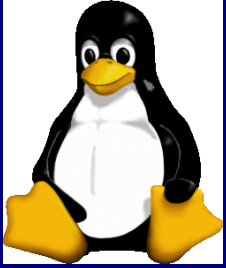
Snort.conf (cont.)

- **If you want output plugins such as snortsam to block IP's for certain events automatically... or be alerted by teneshi or syslog-ng**
 - ★ `output alert_syslog: LOG_AUTH LOG_ALERT`
 - can be used in conjunction with syslog alerting tools
 - ★ `output alert_fwsam: localhost:666/password`



Snort.conf (cont.)

- **Masking, unmasking certain Rulesets at end of conf file**
 - ★ include \$RULE_PATH/misc.rules
 - ★ include \$RULE_PATH/attack-responses.rules
 - ★ #include \$RULE_PATH/oracle.rules
 - ★ #include \$RULE_PATH/mysql.rules
 - ★ include \$RULE_PATH/snmp.rules
 - ★ include \$RULE_PATH/smtp.rules
 - ★ include \$RULE_PATH/imap.rules
 - ★ #include \$RULE_PATH/pop2.rules



Building your own Rules

• Looking at one already chat.rules

★ alert tcp \$HOME_NET any -> \$EXTERNAL_NET any (msg:"CHAT ICQ access"; flow:to_server,established; content:"User-Agent|3A|ICQ"; classtype:policy-violation; sid:541; rev:9;)

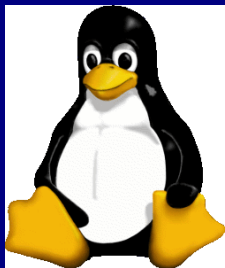
• Looking at a custom rule

★ alert tcp any any -> 192.168.1.0/24 111 (content:"|00 01 86 a5|"; msg:"mounted access");)

→ you can come up with any number of rules like this is a file called say /etc/snort/mycustom.rules

★ For more info on creating custom rules visit

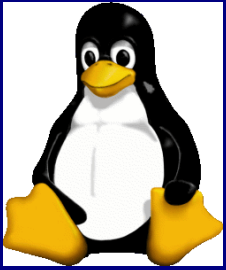
→ http://www.snort.org/docs/snort_manual/node15.html



The Logfile

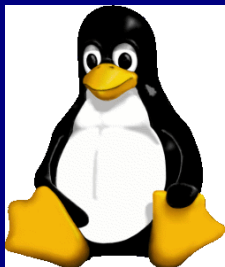
- **by default is /var/log/snort/alert**

```
[**] [1:366:7] ICMP PING *NIX [**]  
[Classification: Misc activity] [Priority: 3]  
09/17-15:13:40.700337 67.182.27.192 -> 63.202.57.162  
ICMP TTL:237 TOS:0x0 ID:9393 IpLen:20 DgmLen:84  
Type:8 Code:0 ID:46915 Seq:688 ECHO  
  
[**] [1:384:5] ICMP PING [**]  
[Classification: Misc activity] [Priority: 3]  
09/17-15:13:40.700337 67.182.27.192 -> 63.202.57.162  
ICMP TTL:237 TOS:0x0 ID:9393 IpLen:20 DgmLen:84  
Type:8 Code:0 ID:46915 Seq:688 ECHO  
  
[**] [119:4:1] (http_inspect) BARE BYTE UNICODE ENCODING [**]  
09/17-15:16:33.425751 63.202.57.162:7786 -> 216.155.193.166:80  
TCP TTL:64 TOS:0x0 ID:37312 IpLen:20 DgmLen:74 DF  
***AP*** Seq: 0xD3E692B7 Ack: 0x4ECF4389 Win: 0x1C84 TcpLen: 20  
  
[**] [1:2452:4] CHAT Yahoo IM ping [**]  
[Classification: Potential Corporate Privacy Violation] [Priority: 1]  
09/17-15:16:36.521867 63.202.57.162:32373 -> 216.155.193.163:5050  
TCP TTL:64 TOS:0x0 ID:37396 IpLen:20 DgmLen:60 DF  
***AP*** Seq: 0x563B4D7E Ack: 0xA2E541D5 Win: 0x1C84 TcpLen: 20
```



The Directory Structure

```
drwx----- 2 snort snort 80 Sep 18 03:28 80.67.74.119
drwx----- 2 snort snort 80 Sep 18 03:47 80.67.74.96
drwx----- 2 snort snort 80 Sep 18 03:44 80.67.74.97
drwx----- 2 snort snort 80 Sep 18 09:32 81.152.160.90
drwx----- 2 snort snort 80 Sep 18 01:52 81.164.189.124
drwx----- 2 snort snort 80 Sep 18 03:46 81.18.71.152
drwx----- 2 snort snort 80 Sep 18 04:15 81.185.80.215
drwx----- 2 snort snort 80 Sep 18 08:55 81.40.170.150
drwx----- 2 snort snort 80 Sep 18 08:35 82.154.116.43
drwx----- 2 snort snort 80 Sep 18 03:05 82.154.225.206
drwx----- 2 snort snort 80 Sep 18 03:07 82.155.41.20
drwx----- 2 snort snort 80 Sep 18 08:50 82.160.25.2
drwx----- 2 snort snort 80 Sep 18 04:04 82.175.228.27
drwx----- 2 snort snort 80 Sep 18 03:26 82.35.132.123
drwx----- 2 snort snort 80 Sep 17 16:55 82.39.75.103
drwx----- 2 snort snort 80 Sep 18 06:45 82.80.153.121
drwx----- 2 snort snort 80 Sep 18 07:26 83.24.10.81
drwx----- 2 snort snort 80 Sep 18 03:32 83.25.12.165
drwx----- 2 snort snort 80 Sep 18 09:02 83.27.160.219
drwx----- 2 snort snort 80 Sep 18 00:10 83.30.84.133
drwx----- 2 snort snort 80 Sep 18 03:47 84.97.138.8
-rw----- 1 snort snort 880K Sep 18 10:00 alert
```

Snortsnarf

SnortSnarf summary page



Top 21 destination IPs

SnortSnarf v021111.1

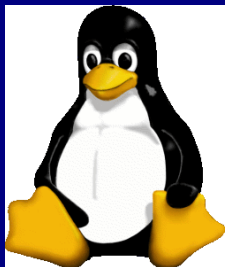
[Signature section \(679\)](#) [Top 20 source IPs](#) [Top 20 dest IPs](#)

This page provides summary information about alerts acquired using input module SnortFileInput, with sources:

- /var/log/snort/alert

The most active destination IPs are shown. Rank is determined by the number of alerts with that IP as the destination. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Destination IP	# Signatures triggered	Originating sources
rank #1	192 alerts	63.202.57.162	4 signatures	(4 source IPs)
rank #2	145 alerts	216.155.193.166	1 signatures	63.202.57.162
rank #3	61 alerts	206.190.38.131	1 signatures	63.202.57.162
rank #4	25 alerts	63.202.57.170	6 signatures	(4 source IPs)
rank #5	24 alerts	63.202.57.163	3 signatures	(3 source IPs)
rank #6	22 alerts	63.202.57.165	3 signatures	(12 source IPs)



Snortalog

Legend :

- RED :** Dangerous connections (potentially bad, further investigation needed!)
- GREEN :** Warning connections (strange, may need further investigation!)
- BLACK :** Not dangerous alert

General Statistics

- The distribution of event by hour
- Popularity of one source host
- Popularity of one destination host
- The distribution of event by destination port
- The distribution of event by protocols
- The distribution of event type of log

Specific Statistics

- Events from one host to any with same method
- Events to one host from any with same method
- Events from a host to a destination
- Events to one destination port grouped by attack
- Distribution of attack methods
- Distribution of classification method
- The distribution of event by severity
- Events by hour

The distribution of event by protocols

Graph

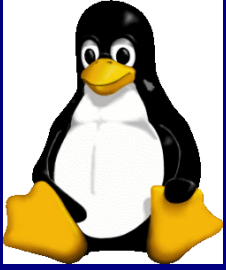
%	No	Protocols
52.58	500	icmp
33.75	321	tcp
13.67	130	udp



The distribution of severity

Graph





Intrusion Prevention System (IPS)

🌐 Snortsam

- ★ allow you to use the snort alerts to block hosts that activate alerts
- ★ must be compiled into the snort binary
- ★ snort.conf
 - ➔ have to add output alert_fwsm: host:port/password
- ★ /etc/snortsam.conf
 - ➔ iptables eth0 log
 - ➔ defaultkey password
 - ➔ accept 127.0.0.1
 - ➔ dontblock 63.202.57.160/27
 - ➔ logfile /var/log/snortsam.log
 - ➔ daemon
 - ➔ bindip 127.0.0.1
- ★ <http://www.snortsam.net/>