SNORT NIDS

Tutorial

Working with Snort

What is SNORT?

Snort is a free open source network traffic analysis tool, written by Martin Roesch at Sourcefire (Cisco acquired for $2.7 billion in July 2013.)

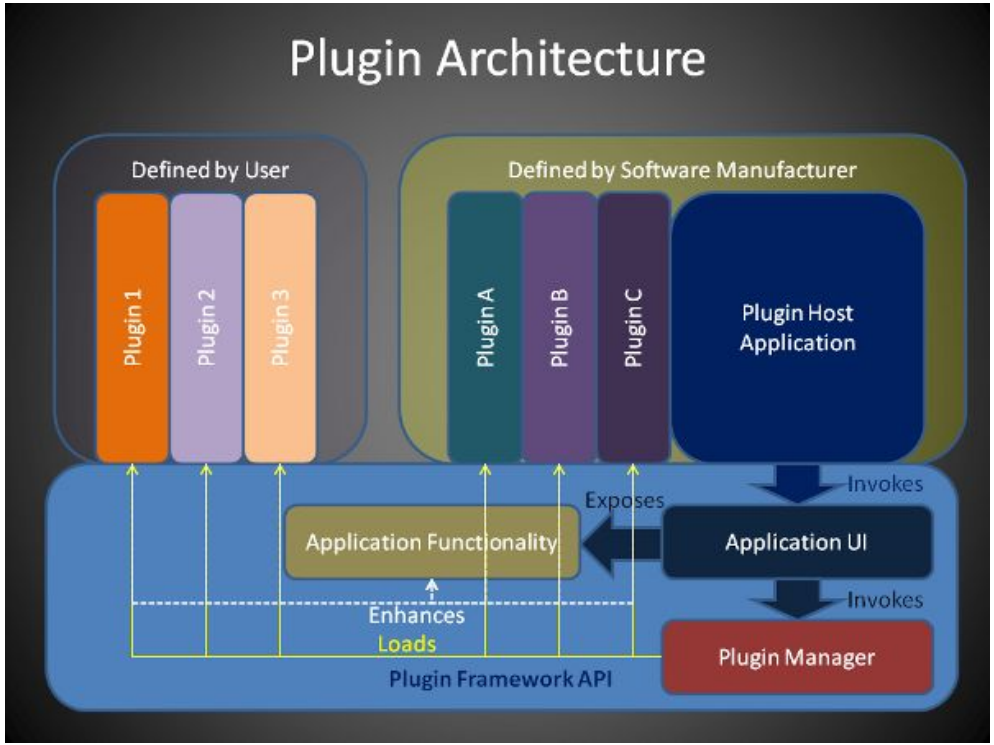
Snort support the following network analysis functionalities:

1. Sniffer
2. Packet Logger
3. Intrusion Detection
4. Intrusion Prevention

Snort could be used for real-time network traffic analysis or for after the fact network forensics analysis.

SNORT Design & Architecture

SNORT has a lightweight design based on a **plug-in architecture**. This plug-in architecture give SNORT endless flexibility to extend and customize SNORT features and capabilities.



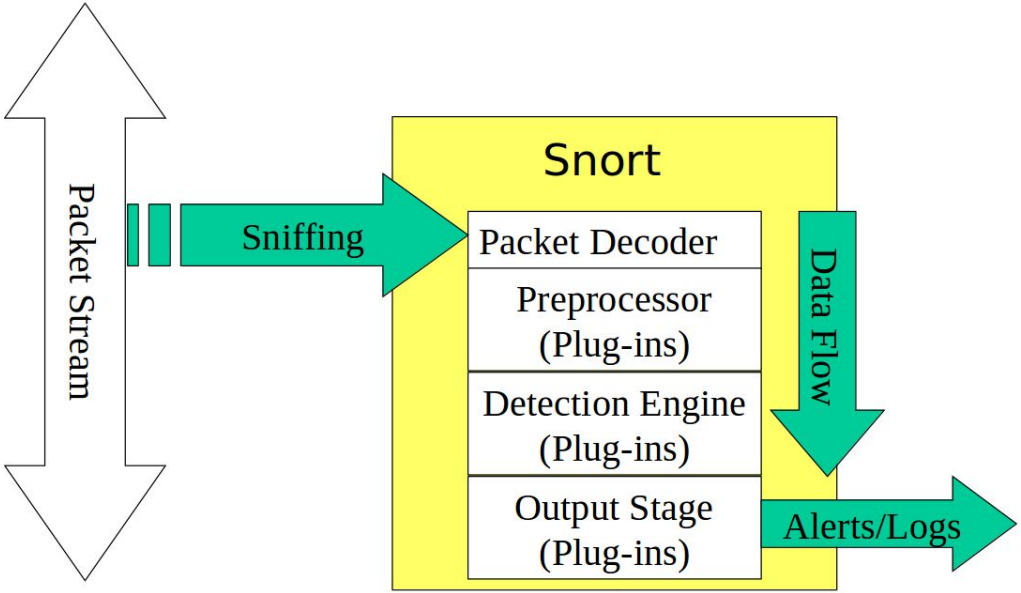
SNORT Plugins Types

You can extend snort by implementing your own components and integrating them with snort, rather than (or in addition to) using the default components.

- **Preprocessor**
  - Packets are examined/manipulated before being handed to the detection engine
- **Detection**
  - Perform single or multiple tests on one or more aspect/field of the packet
- **Output**
  - Report results from the other plug-ins

SNORT Architecture

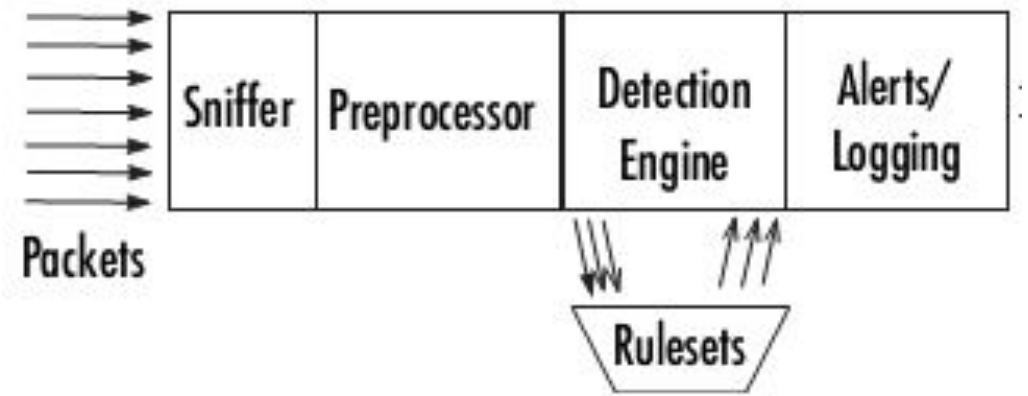
Snort has also a **layer architecture** where each layer receive an input form the top layer, perform some actions and pass the output to the bottom layer



SNORT Architecture and Components

The key **components** in SNORT are:

1. Network Sniffer
2. Preprocessor
3. Detection Engine
4. Knowledge-base
5. Notifier (Logging and Alerts)



SNORT Components: Network Sniffer

This a **basic packet sniffer** has similar features other network traffic sniffer (e.g. tcpdump, wireshark) and it is based on libpcap

The network sniffer also known as **packets decoder** takes the packets from different types of network interfaces

Send the packets to the **preprocessor** if the packets require preprocessing or send the packets to the **detection engine** if preprocessing is not required.

SNORT Components: Preprocessor

The preprocessing allowing users and programmers to drop modular plugins into Snort

The preprocessing code **runs before the detection engine** is called, but after the packet has been decoded.

The preprocessors can **modify and edit the packets data** to prepare them for the detection engine if require.

A preprocess could **rearrange packets contents** that have been crafted by the hacker to avoid detection during deep packet inspection. Or **reassemble packets fragments** and send the whole packet to the detection engine for signature testing.

SNORT Components: Detection Engine

The detection engine uses a set of rules to catch any intrusion activity exists in a packet.

It can dissert a packet and apply rules on different parts of the packet. This includes the:

1. The IP header of the packet
2. The Transport layer header: e.g. TCP, UDP.
3. The application layer level header: e.g. SSH, FTP, HTTP, SNMP, SMTP, IMAP, etc
4. Packet payload: you can create a rule to find a string inside the data.

SNORT Components: Knowledge-base

Snort support a large **rule-based** that is updated regularly by the snort community and CISCO/Sourcefire.

Snort users can **design and write** their own **custom rules**. The rules have if-then-else structure. Rules are usually grouped or categorized by protocols or attacks

- Backdoors
- Chat
- DDoS
- Finger
- FTP
- ICMP
- IMAP
- Multimedia
- MySQL
- NETBIOS
- NNTP
- Oracle
- P2P
- POP
- RPC
- Scan
- Shellcode
- SMTP
- SNMP
- SQL
- Telnet
- TFTP
- Virus
- Web…
- X11

SNORT Components: Notifier|Logging and Output

The captured packet may be used to log the activity or generate an alert.

Logs are kept in simple text files, tcpdump-style files, or some other formats

Log files are stored under **/var/log/snort** folder by default on Debian like OS

Snort support many output plugins such as: **text** output, **syslog** server, **XML**, **IDMEF** (Intrusion Detection Message Exchange Format), **MySQL**, **Oracle**, **SPLANK**, **SMB**, etc

Installing and Configuring SNORT

Snort can run on **Windows**, **Linux**, and **Mac OS**. However, the recommend platform for deploying and running SNORT is Linux-like OS

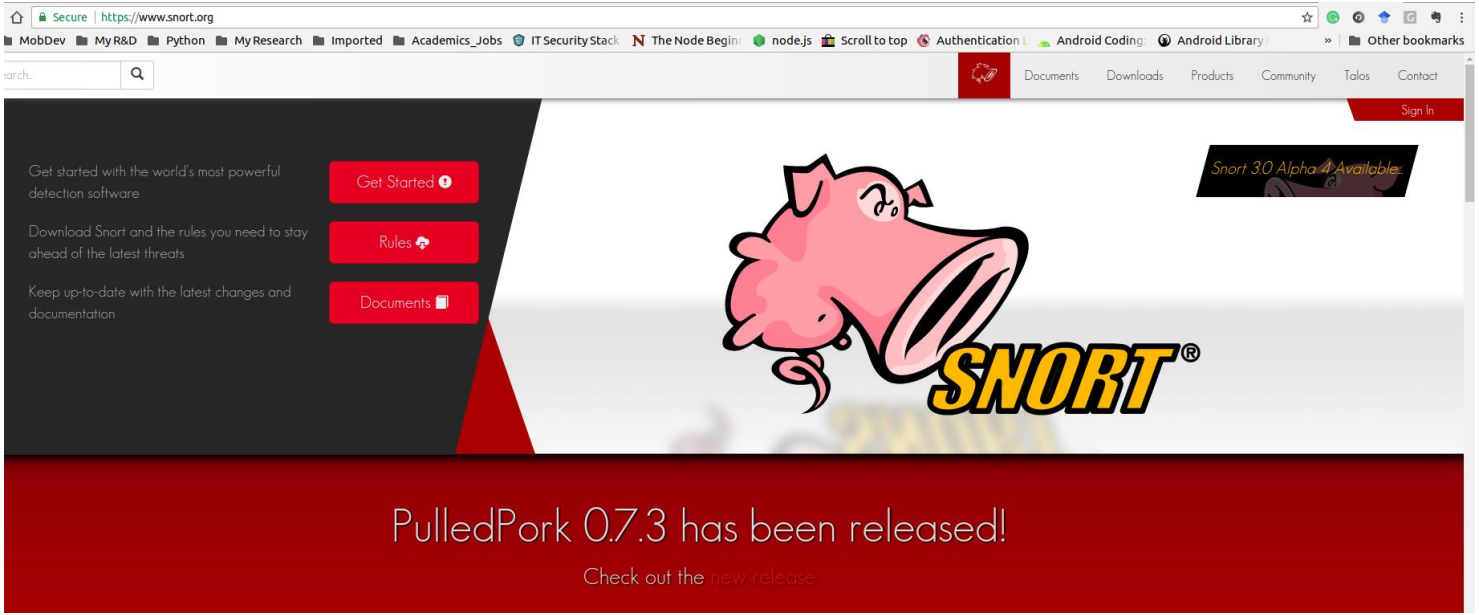
Snort essentially runs via command line interface. There are some third parties GUIs for snort, although the most common use of snort is through a command line.

The basic Snort configuration on different platforms is almost the same with some minor changes.

Snort comes with a **default configuration** that can be modified to execute specific functionality.

Installing SNORT

You can download snort from www.snort.org



Installing SNORT

There are different options to download and install SNORT, depend on you need and setting.

1. You can download SNORT binaries for your platform (e.g. ubuntu, mac, windows, redhat, etc)
2. You can download and install snort using your Linux distro software and package management tool.
3. You can download the source code and build and compile snort from the source code.

Installing SNORT on Ubuntu

To install SNORT on Ubuntu using the software and package manager, use the following commands:

```
sudo apt-get install snort
```

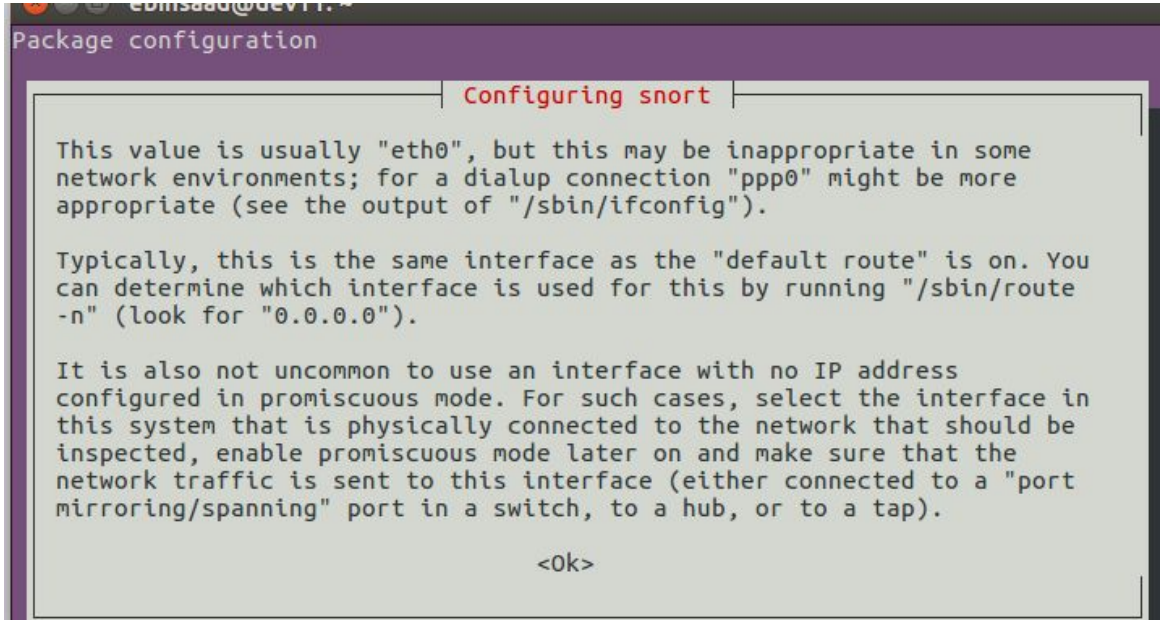
You will need to configure the network interface and the home network during installation. You can edit this information at anytime

You can check snort version after the installation complete using the command

```
snort --version
```

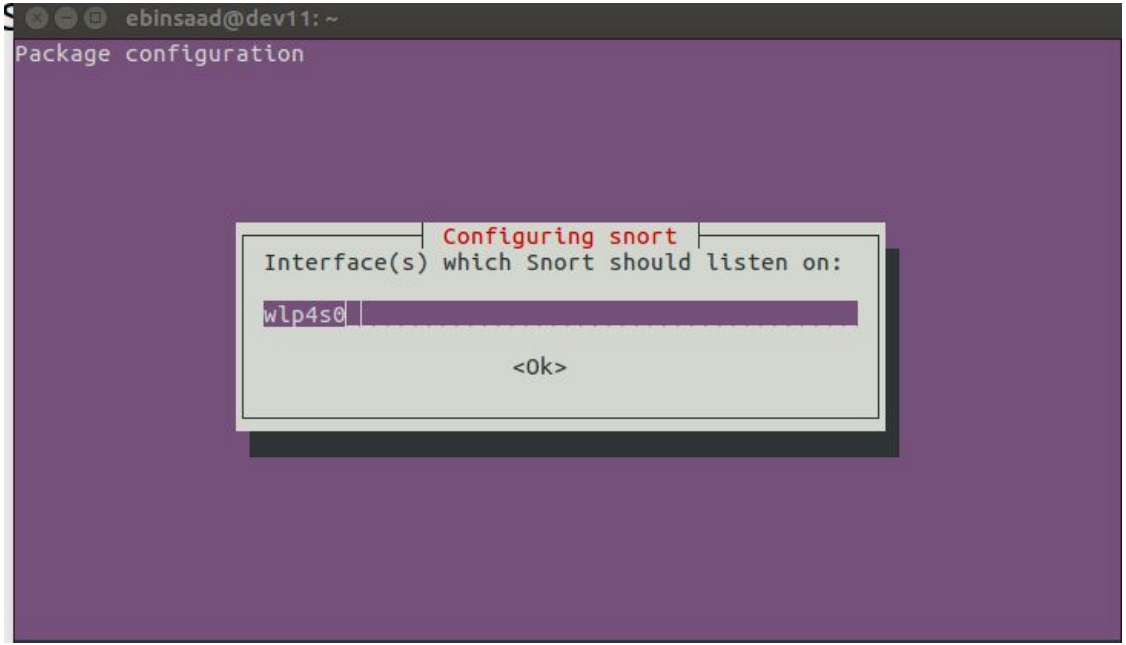
Installing SNORT on Ubuntu

SNORT sniffer configuration



Installing SNORT on Ubuntu

SNORT sniffer configuration and home network



Installing SNORT on Ubuntu

Check snort installation and version by typing:

```
snort  --version
```

```
ebinsaad@dev11: ~
ebinsaad@dev11:~$ snort --version

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8
```

Configuring SNORT

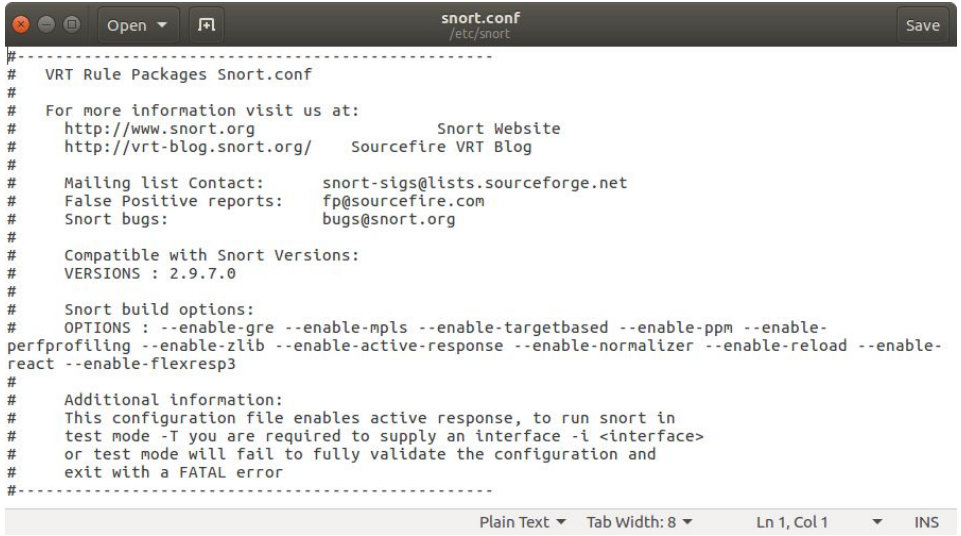
Snort comes with a default configuration file that you can use as a template to configure your snort deployment or to create different configuration files for different analysis tasks. To open and edit the configuration file use the following command

```
sudo gedit /etc/snort/snort.conf
```

Configuring SNORT

To open and edit the configuration file use the following command

```
sudo gedit /etc/snort/snort.conf
```



Configuration Options

```
#################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
#################################################
```

Configuring Home Network

You can define your home network or use the default

```
..
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal
# IP addresses:
#ipvar EXTERNAL_NET !$HOME_NET
```

Configuring Home Network

To setup the network addresses you want to protect, you need to update the $HOME_NET accordingly. The default setting in the snort configuration snort.conf is as follows:

```
ipvar HOME_NET any
```

For instance, to protect subnet 142.104.64.0/24, change the above line in “snort.conf” to the following:

```
ipvar HOME_NET 142.104.64.0/24
```

Configuring Network Services

By default, snort will monitor all the servers running on your network. The following default line indicates that snort will monitor all the web servers on your network:

```
ipvar HTTP_SERVERS $HOME_NET
```

To monitor specific web servers on your network, for instance, running at 142.104.64.199 and 142.104.64.201, change the above line in “snort.conf” to the following:

```
ipvar HTTP_SERVERS [142.104.64.199, 142.104.64.201]
```

Configuring Network Services

You can also configure snort to monitor specific ports. For instance, by updating the HTTP_PORTS variable, you can monitor specific ports running on the web server. For example, the default setting for HTTP is defined in the snort.conf file as follows:

```
ipvar HTTP_PORTS 80
```

You can add additional ports by changing the above setting. For instance, the following line will allow monitoring ports 80, 81, and 8080 on the web server:

```
ipvar HTTP_PORTS [80,81, 8080, 443]
```

Configuring Output Module

This consists of selecting the output plugins and format for Snort. The output plugins entry specifies how snort alerts messages will be logged.

```
##################################################
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
##################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# metadata reference data.  do not modify these lines
include classification.config
```

Configuring Output Module

There are other options to configure the output module. For example, the following line instructs snort to use the CSV format to log the alerts:

```
output alert_csv: alert.csv default
```

You can also log to a database like MySQL, in this case we need first to create and add SNORT database to our MySQL engine. Snort come with a file called **create_mysql**, which has the schema for the database.

```
output database: log, mysql, user=snort
password=snortpass dbname=snort host=mysql.host
```

Configure SNORT rules

Using the snort configuration file we can enable or disable any rules-set. Every group of rules-set is usually stored in one rules file

```
##################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
##################################################

# Note to Debian users: The rules preinstalled in the system
# can be *very* out of date. For more information please read
# the /usr/share/doc/snort-rules-default/README.Debian file

#
# If you install the official VRT Sourcefire rules please review this
# configuration file and re-enable (remove the comment in the first line) those
# rules files that are available in your system (in the /etc/snort/rules
# directory)

# site specific rules
include $RULE_PATH/local.rules
```

Configure SNORT rules

Using the snort configuration file we can enable or disable any rules-set. Every group of rules-set is usually stored in one rules file

```
#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
```

Working with SNORT Rules

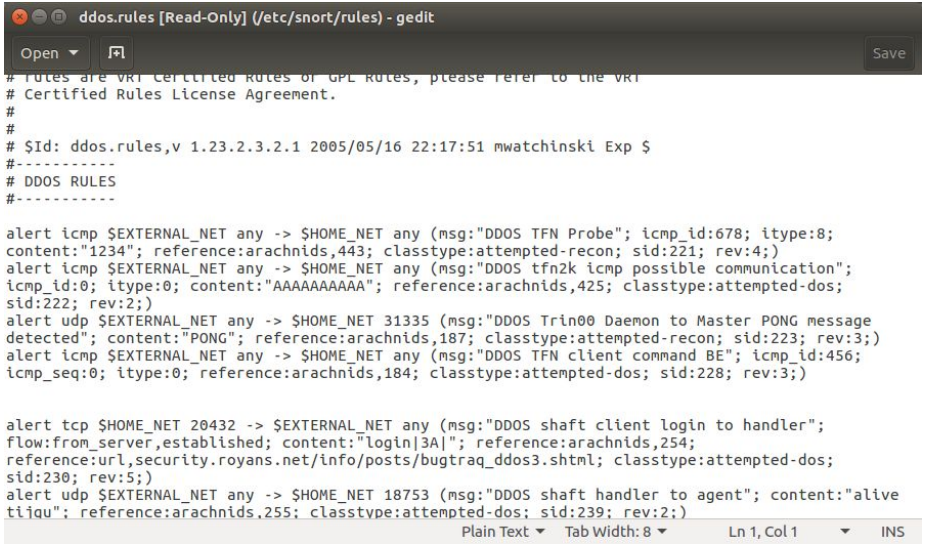
When installing snort using your Linux distro it usually come with many rules set installed on your machine. For **Debian-like OS** the rules are under **/etc/snort/rules/**

```
ebinsaad@dev11: ~
ebinsaad@dev11:~$ ls /etc/snort/rules/
attack-responses.rules          community-web-dos.rules    policy.rules
backdoor.rules                  community-web-iis.rules    pop2.rules
bad-traffic.rules               community-web-misc.rules   pop3.rules
chat.rules                      community-web-php.rules    porn.rules
community-bot.rules             ddos.rules                 rpc.rules
community-deleted.rules         deleted.rules              rservices.rules
community-dos.rules             dns.rules                  scan.rules
community-exploit.rules         dos.rules                  shellcode.rules
community-ftp.rules             experimental.rules         smtp.rules
community-game.rules            exploit.rules              snmp.rules
community-icmp.rules            finger.rules               sql.rules
community-imap.rules            ftp.rules                  telnet.rules
community-inappropriate.rules   icmp-info.rules            tftp.rules
community-mail-client.rules     icmp.rules                 virus.rules
community-misc.rules            imap.rules                 web-attacks.rules
community-nntp.rules            info.rules                 web-cgi.rules
community-oracle.rules          local.rules                web-client.rules
community-policy.rules          misc.rules                 web-coldfusion.rules
community-sip.rules             multimedia.rules           web-frontpage.rules
community-smtp.rules            mysql.rules                web-iis.rules
community-sql-injection.rules   netbios.rules              web-misc.rules
community-virus.rules           nntp.rules                 web-php.rules
community-web-attacks.rules     oracle.rules               x11.rules
```

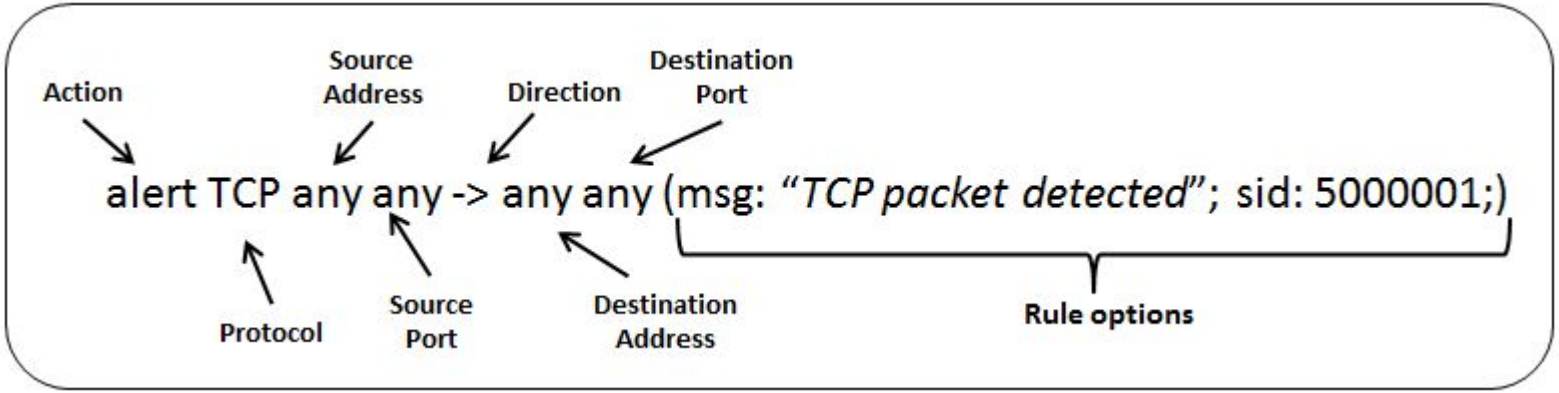
Working with SNORT Rules

You may use any text editor to open, view, and edit the rules files. For example we can open the ddos.rules file using the following command

```
gedit /etc/snort/rules/ddos.rules
```



Understanding SNORT Rules



Action: informs Snort what kind of action to be performed when it detects a packet that matches the rule description. The default action is **alert**, the other actions are: **log, pass, drop, reject and sdrop**.

SNORT Question

Why snort uses “alert” rule only?

Ask Question

3

Among community rules and registered rules, all are "alert" type rules only. Since there are more rule types like log,pass,activate,dynamic,drop,sdrop available , snort official rule sets use only alert type. Why other type of rules are not included in the snort official rule sets ?

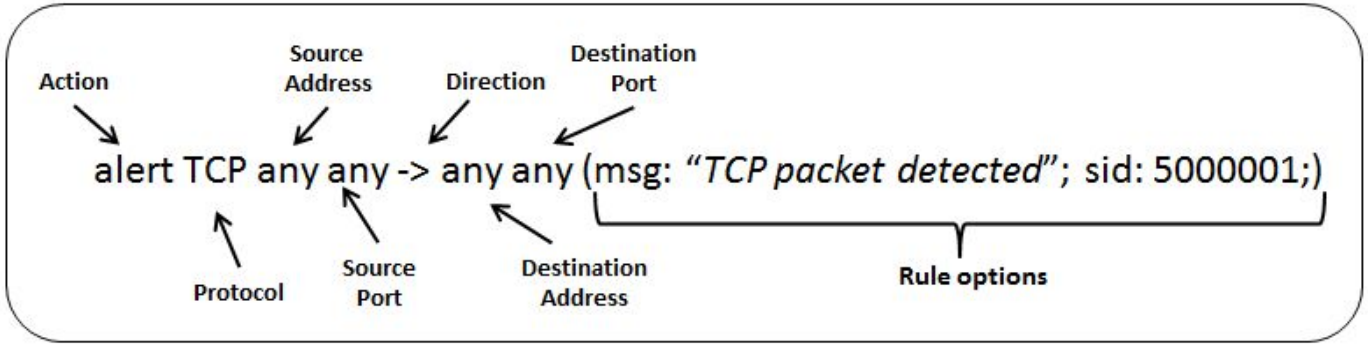
ids snort

asked 2 years, 5 months ago

viewed 688 times

active 2 years, 4 months ago

Understanding SNORT Rules

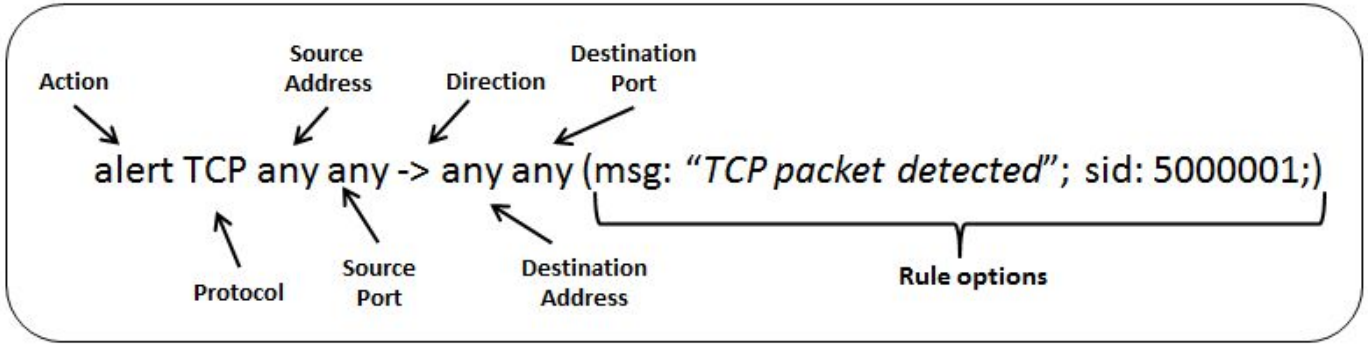


Protocol: this option tell snort to apply this rule for a specific protocol e.g (**ip, tcp, udp, icmp, any**)

Source IP: this option tell snort to apply this rule for a specific source IP address, subnet or any ip address.

Source Port: This part of header describes the source Port from which traffic is coming.

Understanding SNORT Rules



Direction operator ("->", "<>"): It denotes the direction of traffic flow between sender and receiver networks.

Destination IP: This part of header describes the destination network interface in which traffic is coming for establishing a connection.

Destination Port: This part of header describes the destination Port on which traffic is coming for establishing a connection.

Understanding SNORT Rules

Rule Options: The body for rule option is usually **written between circular brackets** "( )" that contains **keywords** with their argument and the keyword are separated by semicolon **";"**

There are general options and the keywords, and there are options and keywords that are protocol specific.

In the options we may specify or search for a unique pattern in the packet payload using the keyword **content**

You may use **regex** or regular expression to **match payload contents**

Writing Snort Rules

We can write a SNORT rule to generate an alert when we detect incoming or outgoing ICMP ping request

alert icmp any any < > any any (itype:8;msg: "ping detected"; sid:1000001;)

The above rule will generate an alert message on every ping packet detect by the IDS regardless the ping message direction.

The above rule is an example of a **valid** SNORT rule but a **bad** one.

Writing Snort Rules

Let us write a rule to detect a **DOS Jolt attack**.

Jolt attack is a denial of service (DoS) attack caused by a very large ICMP packet that is fragmented in such a way that the targeted machine is unable to reassemble it

```
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"DOS Jolt attack";
dsize:408; fragbits:M; reference:cve,1999-0345; classtype:attempted-dos;
sid:268; rev:4;)
```

Writing Snort Rules

Let us write a rule to detect potential SQL injection attacks that use SQL keywords like [and, or] or attacks that use special characters

alert tcp any any -> any 80 (msg: "AND SQL Injection"; content: "and" ; nocase; sid:100000008;)

alert tcp any any -> any 80 (msg: "OR SQL Injection"; content: "or" ; nocase; sid:100000009;)

alert tcp any any -> any 80 (msg: "Form Based SQL Injection"; content: "%27" ; sid:1000003;)

Writing Snort Rules

Let us write a rule to malicious attachments (e.g. virus, botnet, etc) in TCP traffic.

**alert tcp $HOME_NET any -> $EXTERNAL_NET 25 (msg:"VIRUS OUTBOUND bad file attachment"; flow:to_server,established; content: "*Content-Disposition|3A|";nocase;*

***pcre:"/filename\s*=\s*.*?\.(?=[abcdehijlmnoprsvwx])(a(d[ep]|s[dfx])|c([ho]m|li|md|pp)|d(iz|ll|ot)|e(m[fl]|xe)|h(lp|sq|ta)|jse?|m(d[abew]|s[ip])|p(p[st]|if|[lm]|ot)|r(eg|tf)|s(cr|[hy]s|wf)|v(b[es]?|cf|xd)|w(m[dfsz]|p[dmsz]|s[cfh])|xl[tw]|bat|ini|lnk|nws|ocx)[\x27\x22\n\r\s]/iR*"; classtype:suspicious-filename-detect; sid:721; rev:8;)**

SNORT Question

Log Attacks in Different Files using Snort

4

How can I log attacks separately using snort. I basically want to log attacks invoked from different files separately. Like if I have 2 files, `ddos.rules` and `log.rules`, then I want logs generated from `ddos.rules` in one file and logs generated from `log.rules` in another.

Is it possible, and if so how can I do it?

Write and Deploy your own Snort Rules

As a new SNORT user it is recommend that you write your new snort rules into the file **local.rules**. Which is commonly used to store custom SNORT rules for a given site.

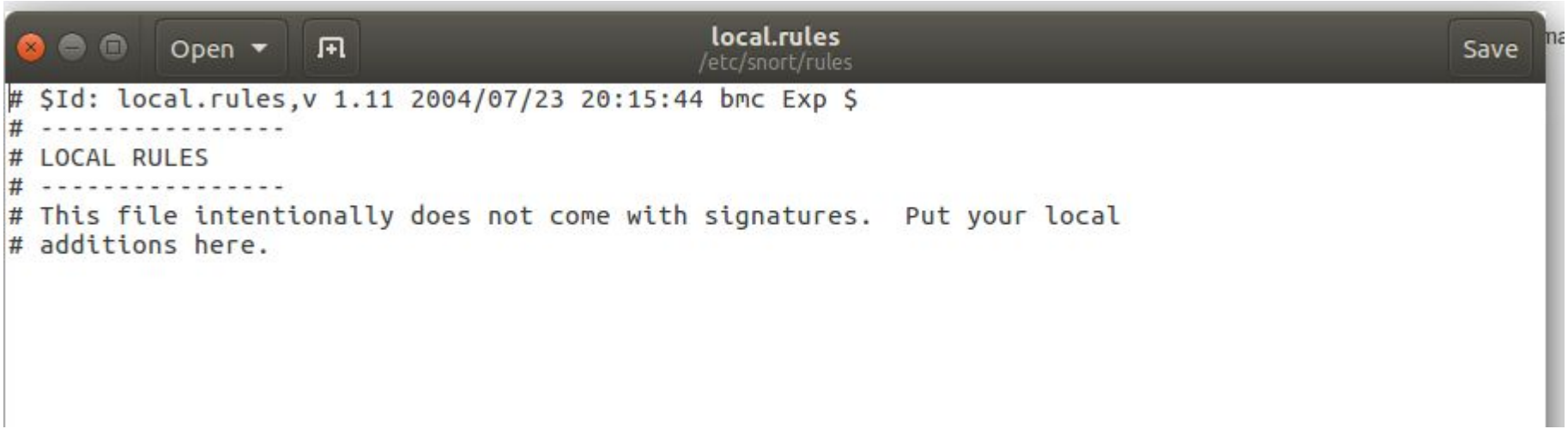
To open this file use any text editor using the following command

sudo gedit /etc/snort/rules/local.rules

```
ebinsaad@dev11:~$ sudo gedit /etc/snort/rules/local.rules
```

Write and Deploy your own Snort Rules

As you can see the **local.rules** is by default an empty file (no predefined rules)



Write and Deploy your own Snort Rules

Let us add the following rule and save the file:

alert tcp any any -> any 443 (msg: "detect HTTPs traffic"; sid:80000000001; rev:1;)

SNORT Log files

SNORT logs are by default stored **/var/log/snort/**

```
ebinsaad@dev11: ~
ebinsaad@dev11:~$ ls /var/log/snort/
snort_alert.csv  snort.log  snort.log.1520437486
ebinsaad@dev11:~$ cp /var/log/snort/snort_alert.csv /home/ebinsaad/
```

SNORT Alerts in CSV file

snort_alert.csv - LibreOffice Calc

Liberation Sans | 10

I7 | 2607:f8b0:400b:80d::200d

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
379	03/07-10:46:32.603961	1	2690588673	1	detect HTTPs traffic	TCP	2001:1970:5e1a:af00:f06c:eada:66b4:e4d4	56270	2607:f8b0:400b:80d::200e	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x56	***A****	0x83ED
380	03/07-10:46:32.674976	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x4A	******S*	0xA65A
381	03/07-10:46:32.704458	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
382	03/07-10:46:32.704642	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x10E	***AP***	0xA65A
383	03/07-10:46:32.714607	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x4A	******S*	0x7605
384	03/07-10:46:32.735226	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
385	03/07-10:46:32.735244	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
386	03/07-10:46:32.735252	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
387	03/07-10:46:32.736168	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x144	***AP***	0xA65A
388	03/07-10:46:32.740857	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x9F	***AP***	0xA65A
389	03/07-10:46:32.741275	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x531	***AP***	0xA65A
390	03/07-10:46:32.753537	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
391	03/07-10:46:32.753586	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x68	***AP***	0xA65A
392	03/07-10:46:32.768839	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
393	03/07-10:46:32.769248	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x110	***AP***	0x7605
394	03/07-10:46:32.798061	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
395	03/07-10:46:32.820090	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
396	03/07-10:46:32.822712	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
397	03/07-10:46:32.822775	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
398	03/07-10:46:32.824703	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
399	03/07-10:46:32.827544	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
400	03/07-10:46:32.827570	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
401	03/07-10:46:32.831381	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0xC0	***AP***	0x7605
402	03/07-10:46:32.832077	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x70	***AP***	0xA65A
403	03/07-10:46:32.841318	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x9F	***AP***	0x7605
404	03/07-10:46:32.845008	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0xE3	***AP***	0xA65A
405	03/07-10:46:32.847390	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x37B	***AP***	0x7605
406	03/07-10:46:32.848247	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x83	***AP***	0xA65A
407	03/07-10:46:32.885201	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
408	03/07-10:46:32.885371	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x68	***AP***	0x7605
409	03/07-10:46:32.930077	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
410	03/07-10:46:32.934538	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
411	03/07-10:46:32.934558	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	60092	34.232.189.211	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x7605
412	03/07-10:46:32.935953	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
413	03/07-10:46:32.936471	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x70	***AP***	0xA65A
414	03/07-10:46:32.945353	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x4A	******S*	0x5EB0
415	03/07-10:46:32.952259	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x9A	***AP***	0xA65A
416	03/07-10:46:32.955754	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x87	***AP***	0xA65A
417	03/07-10:46:33.002059	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x5EB0
418	03/07-10:46:33.002647	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x110	***AP***	0x5EB0
419	03/07-10:46:33.038223	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0xA65A
420	03/07-10:46:33.040553	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	53312	35.186.213.138	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x70	***AP***	0xA65A
421	03/07-10:46:33.061813	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x5EB0
422	03/07-10:46:33.061961	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x5EB0
423	03/07-10:46:33.061973	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0x42	***A****	0x5EB0
424	03/07-10:46:33.064387	1	2690588673	1	detect HTTPs traffic	TCP	192.168.0.169	37956	107.23.53.113	443	90:CD:B6:40:45:F7	BE:D1:65:87:D0:91	0xC0	***AP***	0x5EB0

IDS Rules Best Practice

1. **Disable** all rules and only enable the rules that match your organization security policy.
2. Use **passive actions** like log and alert until you are confident that the rules are correct. Then, you may consider more active actions.
3. Use **simple rules** and avoid complex rules that try to detect multiple attacks patterns.

IDS Rules Best Practice

4. Avoid **overgeneralized** rules; overgeneralized rules will result in false positives

Example: False Positive

alert tcp any any ⇒ HOME_NET 22 (msg: "SSH Brute Force Attempt")

IDS Rules Best Practice

5. Avoid **overfitting** rules that use single attribute for the matching like port number or content. This is usually will increase the false negative.

Example: False Negative

```
alert tcp any any ⇒ HOME_NET 22 (msg:"Potential SSH Brute
Force Attack"; flow:to_server; flags:S; threshold:type threshold,
track by_src, count 30, seconds 60; classtype:attempted-dos;
sid:2001219; rev:4  resp:rst_all  )
```

IDS Rules Best Practice

6. **Group** your custom rules by application, services or attacks.
7. Only **log packages** that you plan to inspect or use for forensic analysis. Example log packets about a virus and malware but do not log packets with spoofed IP
8. Use **informative** log and alert **messages**.
9. Check the available ruleset to make sure you are not creating duplicate rules

IDS Rules Best Practice

10. Add as many as possible information about the attack pattern or signature

1/31-17:37:39.987506 **[1:671:4] "SMTP sendmail 8.6.9c exploit [Classification: Attempted_User_Privilege_Gain]" [Priority: 1]**
{TCP} 1.2.3.4:27191 -> 192.168.1.97:25

Sid: 1:671

Impact: Severe. Remote execution of arbitrary code, leading to remote root compromise.

Affected Systems: Systems running unpatched versions of Sendmail 8.6.10 or earlier.

Corrective Action: Upgrade to the latest version of Sendmail.

Attack Scenario: An attacker sends an email with newline characters and a carriage return, including a path variable of P=/bin/sh. Directives included in the transmission are executed while the message remains in the Sendmail queue.

Certified Intrusion Analyst

https://www.giac.org/certification/certified-intrusion-analyst-gcia

Requirements

- 1 proctored exam
- 100-150 questions
- Time limit of 4 hours
- Minimum Passing Score of 67%

The End

Questions??