Integration of Snort NIDS with Splunk for Threat Hunting and Penetration Testing

PREPARED FOR

Professor Ali Hadi SEC350-02

PREPARED BY

Caitlin Allen & Doug Kapinos Your students

Project Goals and Scope

In an environment set up from scratch, we integrated the network intrusion detection system (NIDS) Snort with our security information and event management (SIEM) solution Splunk. The environment was set up to meet all requirements for segmentation, firewalling, and logging (which will be highlighted in the documentation).

In a working environment, your IDS system, whether it be Snort, Zeek/Bro, Suricata, or Solarwinds, should be right behind the firewall, with the firewall placed in front as the first line of defense. For simplicity, we located ours within the DMZ. Now you may find that Endpoint Detection and Response (EDR) tools are more commonly used in commercial organizations using pricey solutions from top companies like Crowdstrike's Falcon or Elastic's Endgame. But for a smaller organization or a private network like our environment, financially it is not feasible.

Snort is a free NIDS, with the option to pay for different rule sets to use in our firewall. Snort also provides sniffing and packet logging modules to use as well, making it a well-rounded packet analysis tool. We will be using the Community Rules and designing a few of our own local rules. Snort rule sets are easy once you get the syntax down and can understand the rule structure.



This is the general syntax you want to follow. For any additions to the syntax, they will be explained.

Log output can be formatted in a few different ways, like:

- alert_syslog
- alert_fast
- alert_full
- alert_unisock

There are a few different other ways to format them, but we will be using alert_syslog. The alert_syslog format ships logs to the syslog, you can specify the logging facility and priority within the Snort config file and ship these logs to your SIEM solution.



Our penetration testing platform is Mutillidae, is an open-source deliberately vulnerable web-application that allows upcoming pen testers or web app security enthusiasts to practice exploits. This web app is installed using the LAMP stack, composed of four different open-source components:

- Linux
- Apache
- MySQL
- PHP/Perl/Python

Mutillidae is easy to use and allows the user to toggle levels of security, gives hints, and offers OWASP resources on different exploits.





Figure 1: Mutillidae home screen on our Ubuntu Web workstation

Figure 2: Network topology of our environment

1. Setting up Snort on CentOS 7

In these instructions, we will assume your segmented network already exists. Our focus is specifically on our snort server and the Ubuntu web workstation. Your SIEM solution for integration should be properly configured to ingest logs forwarded to it.

We need to install from the source, Snort ran into issues with installing from yum. This documentation is for the current version of Snort available, the download link for a more current version later is available on the <u>Snort website</u> under Binaries.

1. Installing from source.

```
sudo yum install
https://www.snort.org/downloads/snort/daq-2.0.6-1.centos7.x86_64.rpm
sudo yum install
https://www.snort.org/downloads/snort/snort-2.9.16-1.centos7.x86_64.
rpm
```

2. Create folder structure for Snort. It is best to do this now, issues arose if it was done after other steps.

```
mkdir -p /etc/snort/rules
mkdir /var/log/snort
mkdir /usr/local/lib/snort_dynamicrules
```

3. Set directory permissions

```
chmod -R 5775 /etc/snort
```

```
chmod -R 5775 /var/log/snort
```

```
chmod -R 5775 /usr/local/lib/snort_dynamicrules
```

```
chmod -R 5775 /usr/local/lib/snort_dynamicrules
```

```
chown -R snort:snort /var/log/snort
```

```
chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

4. Create new rule files, this is needed so the configuration file is able to execute properly.

```
touch /etc/snort/rule/white_list.rules
```

```
touch /etc/snort/rules/black_list.rules
```

```
touch /etc/snort/rule/local.rules
```

5. Configure Snort to run in NIDS mode

sudo ldconfig

sudo ln -s /usr/local/bin/snort /usr/sbin/snort

6. Set up the Community rules, these are user made and important in making sure Snort runs properly

```
wget https://www.snort.org/rules/community -0 ~/community.tar.gz
sudo tar -xvf ~/community.tar.gz -C ~/
sudo cp ~/community-rules/* /etc/snort/rules
sudo sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/'
/etc/snort/snort.conf
```

7. Configure the network and rule sets by opening the Snort configuration file.

sudo vi /etc/snort/snort.conf

8. Make the changes listed below

In Vim, you can use / to search for lines! This is a long configuration file so be sure to make sure you made these changes.

Add the network address and mask to this line

Setup the network addresses you are protecting ipvar HOME_NET Network_To_BeProtected/XX

```
# Set up the external network addresses. Leave as "any" in most
situations
ipvar EXTERNAL_NET !$HOME_NET
```

Path to your rules files (this can be a relative var RULE_PATH /etc/snort/rules var SO_RULE_PATH /etc/snort/so_rules var PREPROC_RULE_PATH /etc/snort/preproc_rules

Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules

#syslog

Output alert_syslog host=IP_OF_SYSLOG_BOX:514 LOG_AUTH LOG_ALERT



Uncomment the following lines

include \$RULE_PATH/local.rules

include \$RULE_PATH/community.rules

9. Validate the Snort settings

sudo snort -T -c /etc/snort/snort.conf

10. If you get an error, run this command and retry to validate the settings.

ln -s /usr/lib64/libdnet.so.1.0.1 /usr/lib64/libdnet.1

When Snort is successfully installed and initialized, you should see this.



Figure 3: Snort successfully stood up in CentOS 7

11. You now can test the configuration by setting a basic rule to alert ICMP connections. This can also be tweaked to alert for potential nmap attempts since nmap utilizes ICMP for OS Fingerprinting, service detection, and network scanning.

Open the rules up



2. Snort logs on Splunk

Open up the search application on your Splunk host. You can view the logs by host, however, it is easier to search by source. Look for the snortd.log. You can update by latest logs in this to find it easier.

				1
losts (7) Sources (107)	Sourcetypes (1)			
filter	Q			
Host ‡	h	Count ‡	Last Update 💲	
172.16.50.10	al 💌	1	4/26/20 6:31:43.000 PM	
VyOS-Firewall	- di	1,166	4/26/20 7:17:12.000 PM	
WebApp	al 💌	35	4/28/20 3:59:20.000 PM	
edge1	al v	5,840	4/28/20 4:01:21.000 PM	
sec350-snort	al 💌	8,992	4/28/20 3:52:37.000 PM	
sec350-ubuntuWS	al 💌	9	4/28/20 3:30:04.000 PM	
vyos-MGMT	al 💌	7,201	4/28/20 4:01:18.000 PM	

· · · · · · · · · · · · · · · · · · ·		Admir	nistrator ▼ Messages ▼
Data Summary			2
tosts (7) Sources (107) Sourcetypes (1)			
filter Q		< Prev	1 2 3 Next >
Source +	al	Count ‡	Last Update 🗸
/var/log/remote-syslog/sec350-snort/2020.04.28.snortd.log	al 🔻	4	4/28/20 1:31:10.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.rsyslogd.log	al 💌	1	4/28/20 1:12:25.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28log	al 🔻	1	4/28/20 1:12:14.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.NetworkManager.lo g	al 💌	38	4/28/20 1:12:14.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.audispd.log	al 🔻	2	4/28/20 1:12:14.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.auditd.log	al 👻	2	4/28/20 1:12:14.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.augenrules.log	al 🕶	16	4/28/20 1:12:14.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.crond.log	al •	2	4/28/20 1:12:14.000 PM
/var/log/remote-syslog/sec350-snort/2020.04.28.dbus.log	al 🔻	4	4/28/20 1:12:14.000 PM

Figure 4: Splunk snortd.logs and where to find them in the search app

← → C' ŵ	0 🔏 172.16.5	0.3:8000/en-US/app/search/search?q=search source%3D"%2Fvar	%2Flog%2Fremote-syslog 🥬 🗰 🐨 🏠 🐘 🗉 📽
New Search			Save As ▼ Close
source="/var/log/remote-syslog/sec	350-snort/2020.04.	27.snort.log"	Last 24 hours 👻 🔍
✓ 183 events (4/27/20 4:00:00.000 PM to 183 events (4/27/20 4:00:000 PM to 183 events (4/27/20 4:000:000 PM to 183 events (4/27/20 4:000) PM to 184 events (4/20) PM t	o 4/28/20 4:06:54.00	0 PM) No Event Sampling 🔻	Job 🔻 II 🔳 🤌 🎂 🛓 🕴 Smart Mode 💌
Events (183) Patterns Statistics	Visualization		
Format Timeline 💌 🚽 Zoom Out	+ Zoom to Selecti	on ×Deselect	1 hour per column
	List 🔹 🖌 Fo	rmat 20 Per Page 👻	< Prev 1 2 3 4 5 6 7 8 Next >
K Hide Fields 🛛 🗄 All Fields	i Time	Event	
SELECTED FIELDS a NmapAttempt 1	> 4/27/20 10:54:37.00	Apr 27 22:54:37 sec350-snort snort: [1:10000001:1] ICMP test DO PM NmapAttempt = ICMP test (ICMP) source = '/var/log/remote-syslog	: (ICMP) 172.217.10.142 -> 172.16.150.6 g/sec350-snort/2020.04.27.snort.log sourcetype = linux_messages_syslog
a source 1 a sourcetype 1	> 4/27/20 10:54:36.0	Apr 27 22:54:36 sec350-snort snort: [1:10000001:1] ICMP test DO PM NmapAttempt = ICMP test (ICMP) source = <mark>/var/log/remote-syslog</mark>	{TCNP} 172.217.10.142 -> 172.16.150.6 g/sec350-snort/2020.04.27.snort.log sourcetype = linux_messages_syslog
INTERESTING FIELDS # date_hour 4	> 4/27/20 10:53:16.00	Apr 27 22:53:16 sec350-snort snort: [1:10000001:1] ICMP test NmapAttempt = ICMP test (ICMP) source = <mark>/var/log/remote-syslog</mark>	: (ICMP) 172.16.150.5 -> 172.16.150.6 g/sec350-snort/2020.04.27.snort.log sourcetype = linux_messages_syslog
# date_month 1 # date_second 60 a date_wday 1 # date_year 1	> 4/27/20 10:53:16.00	Apr 27 22:53:16 sec350-snort snort: [1:10000001:1] ICMP test IO PM NmapAttempt = ICMP test (ICMP) source = /var/log/remote-syslog	: (ICMP) 172.16.150.6 → 172.16.150.5 g/sec350-snort/2020.04.27.snort.log sourcetype = linux_messages_syslog
	> 4/27/20 10:53:15.00	Apr 27 22:53:15 sec350-snort snort: [1:10000001:1] ICMP test NmapAttempt = ICMP test (ICMP) source = <mark>/var/log/remote-syslog</mark>	: (ICMP) 172.16.158.5 -> 172.16.159.6 g/sec350-snort/2020.04.27.snort.log sourcetype = linux_messages_syslog
a date_zone 1 a host 1 a index 1	> 4/27/20 10:53:15.00	Apr 27 22:53:15 sec350-snort snort: [1:10000001:1] ICMP test IO PM NmapAttempt = ICMP test (ICMP) source = /var/log/remote-syslog	: (ICMP) 172.16.150.6 -> 172.16.150.5 g/sec350-snort/2020.04.27.snort.log sourcetype = linux_messages_syslog
# linecount 1	> 4/27/20	Apr 27 20:51:34 sec350-snort snort: [1:10000001:1] ICMP test	: (ICMP) 172.16.150.6 -> 172.16.150.5

Figure 5: ICMP test logs (Field extraction done for Dashboards)

3. Setting Up Mutillidae

On the snort host, you will be installing the following packages.

- 1. httpd
- 2. mariadb-server
- 3. mariadb
- 4. php
- 5. php-mysql
- 6. php-pear
- 7. php-pear-db
- 8. php-mbstring
- 9. git

This is a CentOS box so yum is how you should be downloading these packages!

Start the httpd service after installation and open port 80 with firewall-cmd

We need to do a MySQL secure installation.

Run the following

sudo mysql_secure_installation

This will prompt you for your root password then ask you to change it. You do not need to change your root password if you do not want to.

For the following questions after the password prompt, make sure to allow remote access

Now, run the following to grab Mutillidae off Github



Copy the Mutillidae folder into /var/www/html

Then navigate to http://LOCALHOST_IP/mutillidae

If successful, you should see the following

172.16.150.6/mutillidae	e/ × 🖻 Search Splunk 8.0.3 × 🖻 Search Splunk 8.0.3 ×	+
↔ ∀ ∅	🖲 😼 172.16.150.6/mutillidae/	··· 🗵 🕁
	🐳 OWASP Mutillidae II	: Keep Calm and Pwn On
	Version: 2.7.14 Security Level: 0 (Hosed)	Hints: Enabled (1 - Try easier) Not Logged In
	Home Login/Register Toggle Hints Show Popup Hints Toggle	e Security Enforce SSL Reset DB View Log View Captured Data
OWASP 2017	Hints and Videos	
OWASP 2013		TIP: Click Hint and Videos
OWASP 2010		on each page
OWASP 2007	•	
Web Services	What Should I Do? New What's	New? Click Here
HTML 5		
Help rs	🥞 Help Me!	of vulnerabilities
Resources •		
Donate Want to Help?	下 Video Tutorials 🥵 Release	e Announcements
Video Tutorials	Latest Version 🦉 Helpful	hints and scripts
	Some Useful Firefox Add-ons 🔗 Bug Re	port Email Address

Figure 6: Mutillidae

4. Integrating Snort with Mutillidae

Since Snort is a traffic analyzer, let's set some rules to pick up potential attacks from Mutillidae.

Open the local.rules file again.

Lets add some basic rules to detect SQL Injections, Command Injections, and Cross-Site Scripting attacks.

<pre>HSQL Injection rules alert tcp any any -> any 80 (msg: "SQL Injection Detected"; content: "%27"; nocase; sid:100000001 alert tcp any any -> any 80 (msg: "SQL Injection Detected"; content: "%27"; nocase; sid:100000001 HXSS Rules alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "%30Script%3E nocase; sid:100000013;) alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "%30Script%3E alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "%30Script%3E alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "%script%"; se; sid:100000001;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passwd"; noc sid:1000000015;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/shadow"; noc sid:1000000015;)</pre>	
<pre>alert tcp any any -> any 80 (msg: "SQL injection Detected"; content: "227"; nocase; sid:100000001 #XSS Rules alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "2305cript23E mocase; sid:1000000013;) alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "2505cript23E #Command Injection Attempts alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passud"; noc sid:100000015;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passud"; noc sid:100000015;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passud"; noc sid:100000015;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passud"; noc sid:100000015;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/shadow"; noc sid:100000016;) </pre>	
<pre>alert tcp any any -> any 80 (msg: "SQL Injection Detected"; content: "22" ; nocase; sid:100000001 #XSS Rules alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "/>Content: "/>Coscript/>E scripting Attack Detected"; content: "/>Content: "/>Content: "/>Content: "/>Command Injection Attempts alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passud"; noc sid:1000000016;) alert tcp any any -> any 80 (msg: "Command Injection Detected"; content: "cat /etc/passud"; noc sid:1000000016;) </pre>	1999113
<pre>#XSS Rules alert tcp any any -> any 88 (msg: "Cross-Site Scripting Attack Detected"; content: "%3Cscript%3E mocase; sid:1800000013;) alert tcp any any -> any 80 (msg: "Cross-Site Scripting Attack Detected"; content: "<script></script></pre>	

Figure 7: Snort local.rules

Our Snort rules are pretty basic, but <u>here</u> is a good place to find some more advanced SQL and XSS related rules.

Lets test our Cross-Site Scripting rule.

In Mutillidae:

1. OWASP 2017 > Cross Site Scripting (XSS) > Reflected (First Order) > Echo Message

<u>w 2 5 7</u>	V 🛛 1/2.10.150.0/mutit	udae/index.pnp?page=ecno.pnp			๛ พ
	😽 OWAS	5P Mutillidae II	:	Keep Calm and	Pwn On
	Version: 2.7.14	Security Level: 0 (Hosed)	Hi	nts: Enabled (1 - Try easier)	Not Logged In
	Home Login/Register Togg	le Hints Show Popup Hints Toggle	s Se	curity Enforce SSL Reset DB View	Log View Captured
OWASP 2017	A1 - Injection (SQL)				
OWASP 2013	A1 - Injection (Other)	EC	n	O, Echo, Echo	
OWASP 2010	A2 - Broken Authentication and Session Management	🔰 Help Me!			
OWASP 2007	A3 - Sensitive Data Exposure		_		
Web Services	A4 - XML External Entities	s and Videos			
HTML 5	A5 - Broken Access Control	En	+0	r moscogo to ocho	
01	A6 - Security Misconfiguration		Le	Thessage to echo	
others	A7 - Cross Site Scripting (XSS)	Reflected (First Order)	Þ	DNS Lookup	1
Documentation	A8 - Insecure Deserialization	Persistent (Second Order)	Þ	Echo Message	
Resources	A9 - Using Components with Known	DOM-Based	Þ	Pen Test Tool Lookup	
	Vulnerabilities	Cross Site Request Forgery (CSRF)	Þ	Text File Viewer	
Donate	A10 - Insufficient Logging and Monitoring	Via "Input" (GET/POST)		User Info (SQL)	
want to Help?		Via HTTP Headers		Set Background Color	
		Via HTTP Attribute	Þ	HTML5 Web Storage	
Video Tutorials		Via Misconfiguration	•	Capture Data Page	
race futorials		Against HTML5 Web Storage		Document Viewer	
<u></u>		Against JSON		Arbitrary File Inclusion	
2.16.150.6/mutillidae,	/index.php?page=echo.php	Via Cookie Injection	Þ	XML Validator	

2. Enter the Following

Ente	er message to echo
Message	<pre><script>alert("hacked");</script></pre>

3. An alert should pop up if successful. Lets go to Splunk

- -

jle Hints	Show Popup Hints	Toggle Security	Enforce SSL	Reset DB	View Log	View Cap
		Echo F	iche Fot	10		
🥮 He	lp Me	hacked				
ts and '	Video					
			OK	echo		
	M	essage				
		Ech	o Message			

4. If successful, logs should roll through.

New Search				Save As Close
sourcetype=linux_me	essages_syslog >	SS="Cross-Site Scripting	Attack Detected*	Last 24 hours • Q
✓ 2 events (4/27/20 5:0	00:00.000 PM to	4/28/20 5:08:15.000 PM)	No Event Sampling 🔻	Job 🕶 💷 🕐 💩 🛓 📍 Smart Mode 🔻
Events (2) Patterns	s Statistics	Visualization		
Format Timeline •	- Zoom Out	+ Zoom to Selection	× Deselect	1 hour per colum
< Hide Fields	i≣ All Fields	i Time > 4/28/20	Event Aur 28 00-42-46 sec350-sport sport- F1-100000013-01 Cross-Site Scripting Attack D	letected (TCP) 172 16 150 5-48912 → 172 16 150 6-80
 K Hide Fields SELECTED FIELDS 	I All Fields	> 4/28/20	Apr 28 00:42:46 sec350-snort snort: [1:100000013:0] Cross-Site Scripting Attack D	etected {TCP} 172.16.150.5:48912 -> 172.16.150.6:80
a source 1		12:42:46.000 AM	XSS = Cross-Site Scripting Attack Detected source = /var/log/remote-syslog/sec350-snor	t/2020.04.28.snort.log sourcetype = linux_messages_syslog
a XSS 1		> 4/28/20 12:39:09.000 AM	Apr 28 00:39:09 sec350-snort snort: [1:100000013:0] Cross-Site Scripting Attack D XSS = Cross-Site Scripting Attack Detected source = /var/log/remote-syslog/sec350-snor	etected {TCP} 172.16.150.5:48904 -> 172.16.150.6:80 t/2020.04.28.snort.log sourcetype = linux_messages_syslog
INTERESTING FIELDS # date_hour 1				
# date_mday 1 # date_minute_2				

Try other methods to ping off your command injection alert and SQL injection alert. Heres some hints

Who	would	you like to do a DNS lookup on Enter IP or hostname
Hostnam	Please	172.16.150.6&cat /etc/passwd
Username Password	['OR'1'==	='1; ••

Try to get some logs like this and we can begin to make dashboards

Lis	t 🔹 🖌 Format	20 Per Page ▼
i	Time	Event
>	4/28/20 1:58:02.000 AM	Apr 28 01:58:02 sec350-snort snort: [1:100000011:0] SQL Injection Detected (TCP) 172.16.150.5:55742 -> 172.16.150.6:80 source = /var/log/remote-syslog/sec350-snort/2020.04.28.snort.log sourcetype = linux_messages_syslog sqLinject = SQL Injection Detected
>	4/28/20 1:58:02.000 AM	Apr 28 01:58:02 sec350-snort snort: [1:100000011:0] SQL Injection Detected (TCP) 172.16.150.5:55742 -> 172.16.150.6:80 source = /var/log/remote-syslog/sec350-snort/2020.04.28.snort.log sourcetype = linux_messages_syslog sqLinject = SQL Injection Detected
>	4/28/20	Anr 28 01-16-21 sec350-snort snort+ F1-100000011-01 SOL Injection Detected (TCP) 172-16 150 5-40184 -> 172-16 150 6-80

5. Dashboard Building

Every good SOC analyst wants a comprehensive dashboard they can open Splunk right up to.



Figure 15: Home dashboard

In the search app, open up the Dashboard module.

Lets make a dashboard like this one, but for our Snort logs.

172.16.150.6/mutillidae/ ×	Authentication	Dashboar X	Search S	plunk 8.0.3	× +							<u> </u>		
← → ♂ ŵ	0 🔏 172.16.5	0.3:8000/en-U	S/app/search/a	authentication	n_dashboard			90%	🛛 1	2) 🔹	Ξ
splunk>enterprise App: Searc	h & Reporting 🔻					0 4	dministrator 🔻	Messages 🔻	Settings 🕶	Activity • F	lelp 🕶	Find		Q
											>	Search a	& Repor	ting
Authentication Dashbo	bard										Edit	Export	•	
Successful Logons by Host	12:00 AM Mon Apr 27	4.00 AM	SOO AM	12:00 PM	4:00 PM time	Leco PM	12:00 AM Tue Apr 28	4.00 AM	8.00 AM	12:00 Pi	м		ec350-sno	
Failed Logons by Host 75 5 25 26 600 PM 800 PM 800 PM 800 PM	10:00 PM	12:00 AM Tue Apr 28	2:00 AM	400 AM	6.00 AM	800 AM	MA DO OF	12:00 PM	200 F	**************************************	O PM		edge1 vyos-MGM	

Figure 16: Authentication dashboard

Lets make a line graph of Cross-Site Scripting Attempts

Create a line graph dashboard panel, for the search, we'll want to see what hosts are spiking in activity.

Enter this for your search using your snort.log

Title	
rch String	source="/var/log/remote-syslog/sec350-snort/2020.04.28.snort.log" XSS="Cross-Site Scripting Attack Detected" timechart count by host
	Run Search 🖉
ne Range	Use time picker 🔻
	Last 24 hours ►
h Delay ?	No auto refresh 🔻
Indicator	Progress bar 🔹
	Cancel Convert to Report Apply

Figure 17: Search syntax for panels

You want to pipe your results to timechart to show the timing of the attacks, which would be needed for incident response to generate a timeline and identify when an attacker made attempts, you will then count by host to count each time a host has been breached. Figure this out for the other attacks and even add a few other panels, we included nmap attempts for ICMP tests and failed logons by host in case of brute force. To make this easier, use field extraction to extract the exact log messages from Snort you are looking for.



Figure 18: Final dashboard of Snort logs and failed logons with ICMP connections

You can add this to your home page if you like!

7. Troubleshooting & Issues

Our environment was built on Caitlin's HP Proliant DL380 Gen 7 server with ESXi hypervisor. Different network segments were created within ESXi to segment the network and configure the firewalls.

Each VM had to be loaded as an ISO then deployed. One issue we discovered early on was with the unstable version of VyOS we were using, since stable current versions cost money, anytime Caitlin shut down the server, our VyOS settings were wiped from the entire VM. Committing to the configuration file did not matter, VyOS essentially had volatile memory. To solve this, anytime any work was done on VyOS, we snapshotted the machine and were able to overcome this by restoring from the snapshot.

Another issue was Snort's network placement. It was not intended to be set up as a host but as a NIDS for the DMZ. However, just due to configuration issues and realizing that it would not intercept until too late, we decided to put Mutillidae on snort and log directly from the box. To solve this, we would have had to do major network reconfigurations.

8. Conclusion

Snort is a fun packet analyzer with many capabilities! Using the provided link and Snort documentation, building rules to integrate with Splunk

You can use simple rules to get started and use regex, encoding, etc. to get more precise in what your NIDS will detect and ship to Splunk.

Our environment was built on Caitlin's HP Proliant DL380 Gen 7 server with ESXi hypervisor. Different network segments were created within ESXi to segment the network and configure the firewalls.