COMP 8006

Network Administration and Security Level 2

Final Practical Project – Report

(Network Traffic Analysis)

Param Dhaliwal | A00939336

27th March 2021

Contents

Summary

Throughout the project, extensive analysis was in use for three different captured network traffics along with their log files with the help of several tools, and services. The scope of this project was to apply the principles of intrusion detection and packet analysis in a practical application. Each network consisted of different scenarios with interesting complications, yet possible solution. Network 1 appeared to have problems situated with DoS Attacks, attacks which would either flood web services or crash them. It occurs when the attacked system is overwhelmed by large amounts of traffic that the server is unable to handle. The tools used to analyze such network were Linux commands, SnortSnarf, and Splunk Cloud. Network 2, on the other hand, used OSSEC alerts as its intrusion detection system. Its IDS was able to provide alerts as they were being activated, however, network 2 did conclude having large amount of failed login attempts within a short duration; resulting in malicious activities, led to break-in. The tools used to analyze, under this network, were Linux commands, Wireshark, and Splunk Cloud. Lastly, in Network 3, its IDS (snort) was able to warn on the attacks such as attempts for information leak, and the flow of bad traffic. Recorded traffic captures for this network does contribute substantial amount of information. And to analyze it further, tools such as Argus, Snort, Linux commands, and Splunk Cloud were of use.

Introduction

The objective of this report is to analyze the raw data and present a report that will allow the administrator of the network on which the traffic was captured to get an accurate idea of benign as well as anomalous and dangerous network activity in and out of the network. To perform a security audit on a data set consisting of captured network traffic. The data analyzed for this project was distributed by our instructor, Aman Abdulla. Such data consisted of three different network directories where each directory contained files and directories such as snort data, tcpdump captures, and other miscellaneous logfiles. An analytic observation and findings are detailed in this report.

To detect network intrusion or spyware, to figure out which part of the network is being targeted by a DDoS attack, or to determine when and where an issue has come up, such tool(s) should be of a requirement to function more efficiently and with care. The tools used throughout this project, on the basis of analysis, are listed below:

Basic Linux commands

Command like grep was in use to search for matching text in a file, or in output from other commands. It is included by default in most Linux distributions and is also available for Windows and Mac. Other used commands are sort and awk to filter out search on a specific field value instead of doing a full text search. This makes our log analysis more accurate because it will ignore undesired matches from other parts of the log message. Awk is a powerful command line tool that provides a complete scripting language, that can filter and parse out fields more effectively.

Snort and SnortSnarf

The purpose of Snort is to act like a network packet analyzer and listen to every packet sent and received across the wire that is being monitored. Snort has a database of traffic signatures that are common network attacks or other malicious activity. Snort compares every packet to that database. Another benefit for using snort was to read a pcap file and process it according to the rules in the *snort.conf* file. SnortSnarf, on the other hand, is a script to take alerts from the Snort Intrusion Detection System and produce HTML output.

Argus

Argus helped processing packet data and generate summary network flow data. It is a great way of looking at aspects of the data that we cannot readily get from packet analyzers. Argus helps us generate network flow status to answer questions, such as how many hosts are talking, who is talking to whom, how often, is one address sending all the traffic?

Splunk Cloud

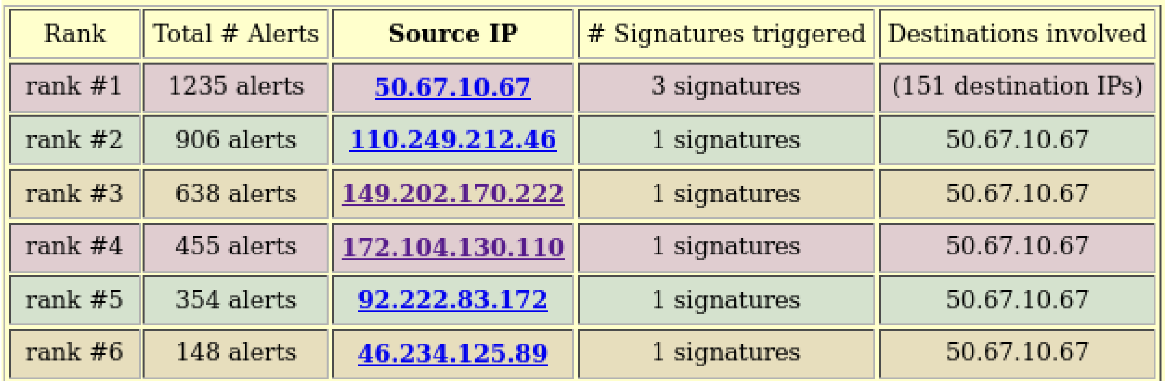
Splunk is a software platform widely used for monitoring, searching, analyzing, and visualizing the machine-generated data in real time. It performs capturing, indexing, and correlating the real time data in a searchable container and produces graphs, alerts, dashboards, and visualizations.

Network 1

Tools used: Linux commands, SnortSnarf, Splunk Cloud

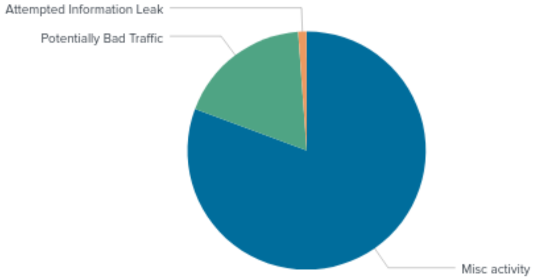
Analysis

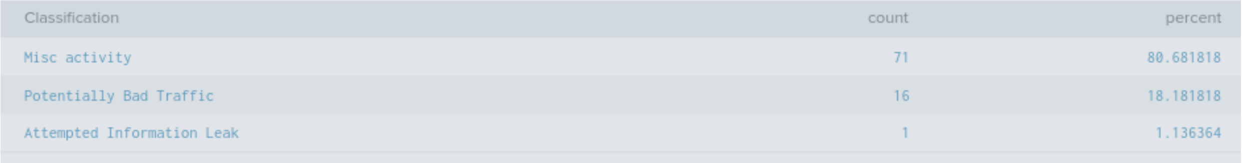
Network 1 had large number of DoS Attacks and other reconnaissance activities which I was able to analyze by looking at the SnortSnarf alert that gave us top sources IP addresses with their classifications.



Rank	Total # Alerts	**Source IP**	# Signatures triggered	Destinations involved
rank #1	1235 alerts	**50.67.10.67**	3 signatures	(151 destination IPs)
rank #2	906 alerts	**110.249.212.46**	1 signatures	50.67.10.67
rank #3	638 alerts	**149.202.170.222**	1 signatures	50.67.10.67
rank #4	455 alerts	**172.104.130.110**	1 signatures	50.67.10.67
rank #5	354 alerts	**92.222.83.172**	1 signatures	50.67.10.67
rank #6	148 alerts	**46.234.125.89**	1 signatures	50.67.10.67

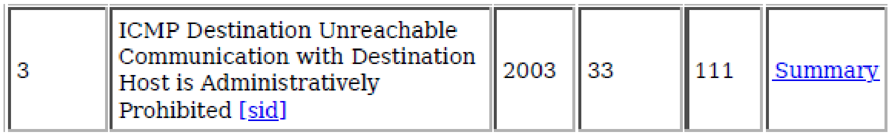
From the Snort, the first top 5 IP addresses, ones with DoS attempts to 50.10.67.10 being their destination are also lying under the classification of Attempted Denial of Service for the top IP address being their source. High number of classifications such as misc activity, potential bad traffic, and information leak can also be seen below (filtered with splunk):





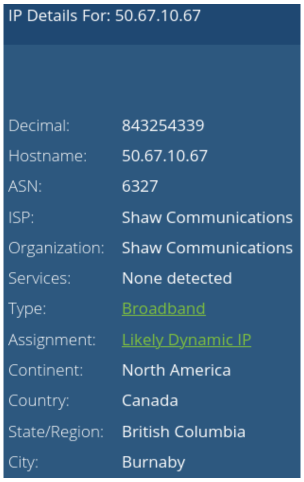
Classification	count	percent
Misc activity	71	80.681818
Potentially Bad Traffic	16	18.181818
Attempted Information Leak	1	1.136364

The top source IP addresses (50.67.10.67, 110.249.212.46, 92.222.83.172) had a large flow of traffic in between and were showing ICMP type 3 code 10 message, stating "ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited" which were being rejected by the firewall rules created on the system. This seemed to be a malicious activity.



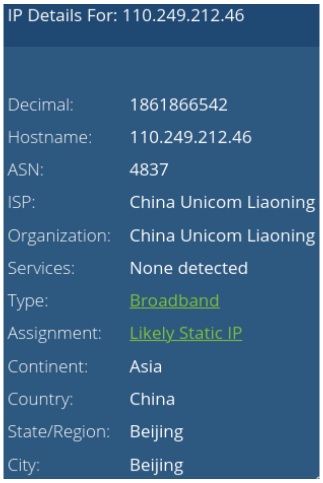
3	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited [sid]	2003	33	111	Summary

IP addresses with high alerts were then looked up on the web with the help of "whatismyipaddress" to get their locations, as we can see below:



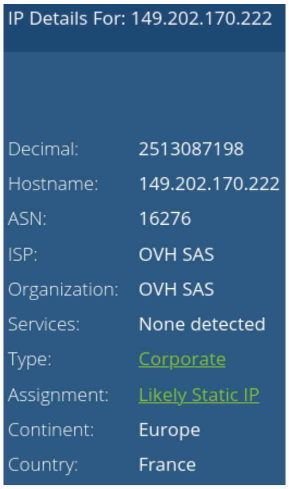
IP Details For: 50.67.10.67

Decimal:	843254339
Hostname:	50.67.10.67
ASN:	6327
ISP:	Shaw Communications
Organization:	Shaw Communications
Services:	None detected
Type:	Broadband
Assignment:	Likely Dynamic IP
Continent:	North America
Country:	Canada
State/Region:	British Columbia
City:	Burnaby



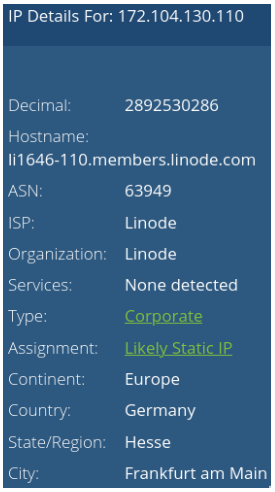
IP Details For: 110.249.212.46

Decimal:	1861866542
Hostname:	110.249.212.46
ASN:	4837
ISP:	China Unicom Liaoning
Organization:	China Unicom Liaoning
Services:	None detected
Type:	Broadband
Assignment:	Likely Static IP
Continent:	Asia
Country:	China
State/Region:	Beijing
City:	Beijing



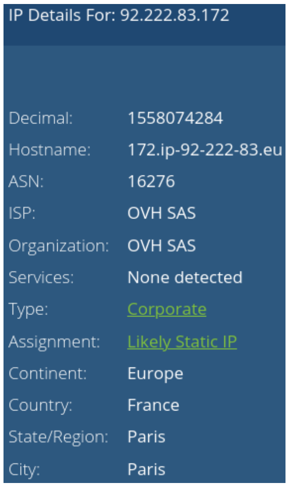
IP Details For: 149.202.170.222

Decimal:	2513087198
Hostname:	149.202.170.222
ASN:	16276
ISP:	OVH SAS
Organization:	OVH SAS
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	France



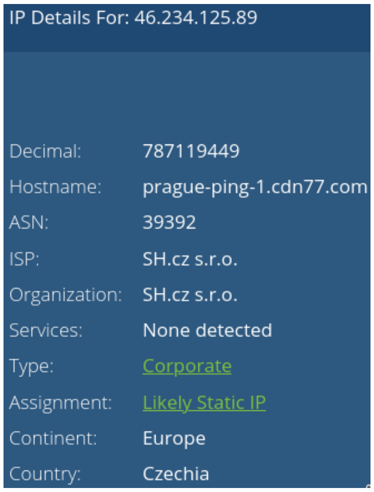
IP Details For: 172.104.130.110

Decimal:	2892530286
Hostname:	li1646-110.members.linode.com
ASN:	63949
ISP:	Linode
Organization:	Linode
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	Germany
State/Region:	Hesse
City:	Frankfurt am Main



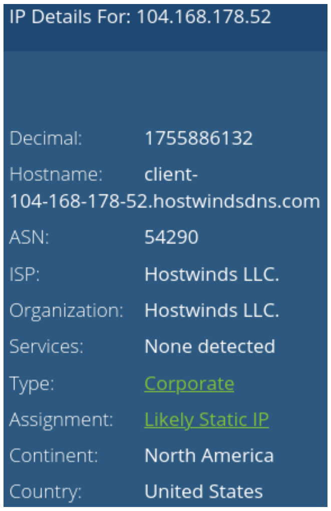
IP Details For: 92.222.83.172

Decimal:	1558074284
Hostname:	172.ip-92-222-83.eu
ASN:	16276
ISP:	OVH SAS
Organization:	OVH SAS
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	France
State/Region:	Paris
City:	Paris



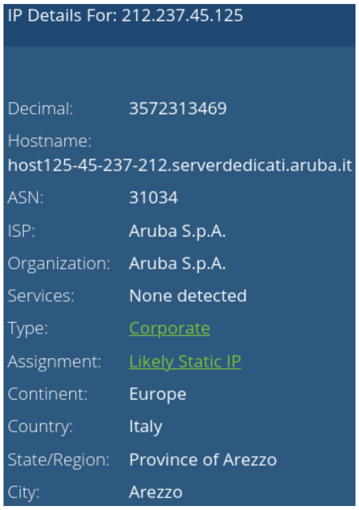
IP Details For: 46.234.125.89

Decimal:	787119449
Hostname:	prague-ping-1.cdn77.com
ASN:	39392
ISP:	SH.cz s.r.o.
Organization:	SH.cz s.r.o.
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	Czechia



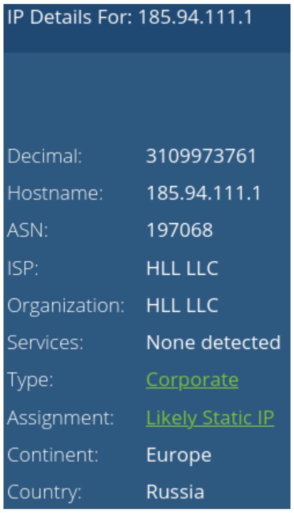
IP Details For: 104.168.178.52

Decimal:	1755886132
Hostname:	client-104-168-178-52.hostwindsdns.com
ASN:	54290
ISP:	Hostwinds LLC.
Organization:	Hostwinds LLC.
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	North America
Country:	United States



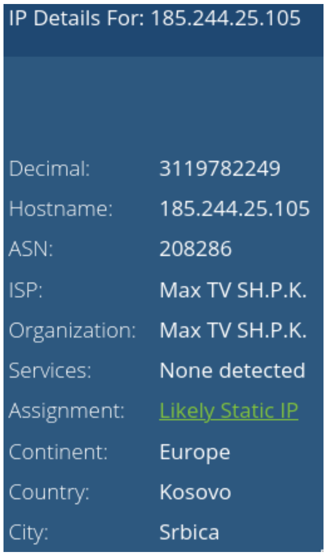
IP Details For: 212.237.45.125

Decimal:	3572313469
Hostname:	host125-45-237-212.serverdedicati.aruba.it
ASN:	31034
ISP:	Aruba S.p.A.
Organization:	Aruba S.p.A.
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	Italy
State/Region:	Province of Arezzo
City:	Arezzo



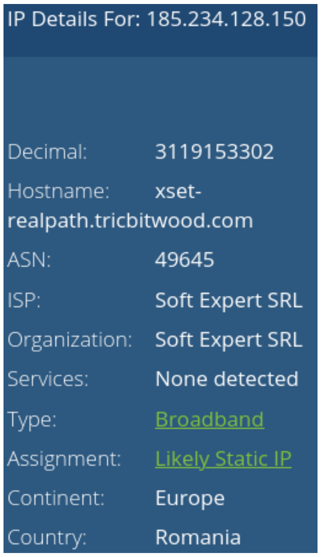
IP Details For: 185.94.111.1

Decimal:	3109973761
Hostname:	185.94.111.1
ASN:	197068
ISP:	HLL LLC
Organization:	HLL LLC
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	Russia



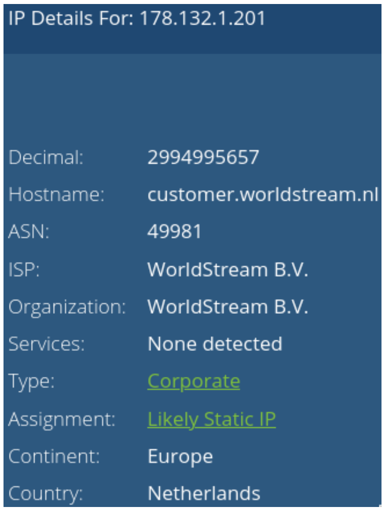
IP Details For: 185.244.25.105

Decimal:	3119782249
Hostname:	185.244.25.105
ASN:	208286
ISP:	Max TV SH.P.K.
Organization:	Max TV SH.P.K.
Services:	None detected
Assignment:	Likely Static IP
Continent:	Europe
Country:	Kosovo
City:	Srbica



IP Details For: 185.234.128.150

Decimal:	3119153302
Hostname:	xset-realpath.tricbitwood.com
ASN:	49645
ISP:	Soft Expert SRL
Organization:	Soft Expert SRL
Services:	None detected
Type:	Broadband
Assignment:	Likely Static IP
Continent:	Europe
Country:	Romania

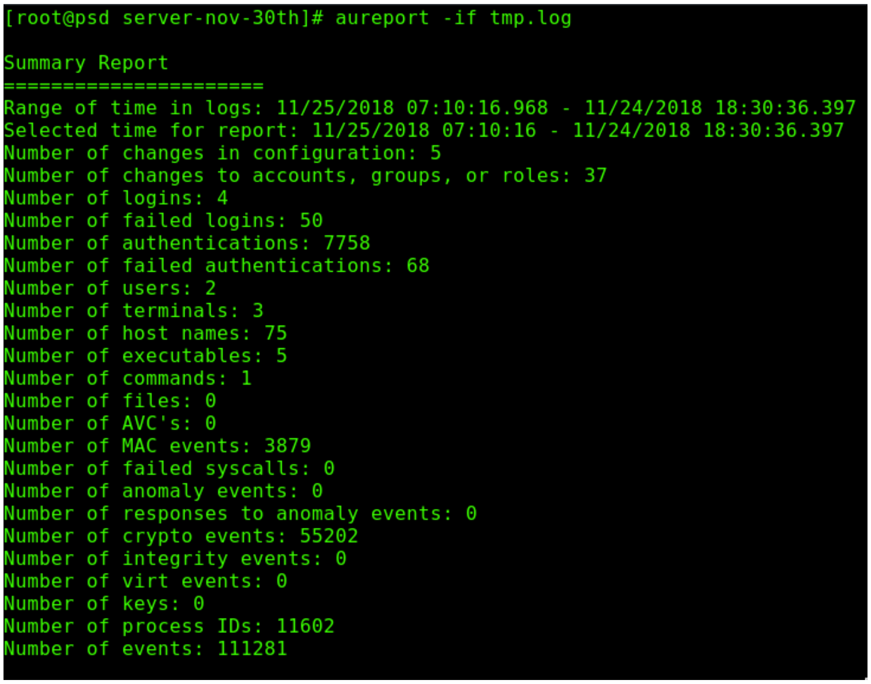


IP Details For: 178.132.1.201

Decimal:	2994995657
Hostname:	customer.worldstream.nl
ASN:	49981
ISP:	WorldStream B.V.
Organization:	WorldStream B.V.
Services:	None detected
Type:	Corporate
Assignment:	Likely Static IP
Continent:	Europe
Country:	Netherlands

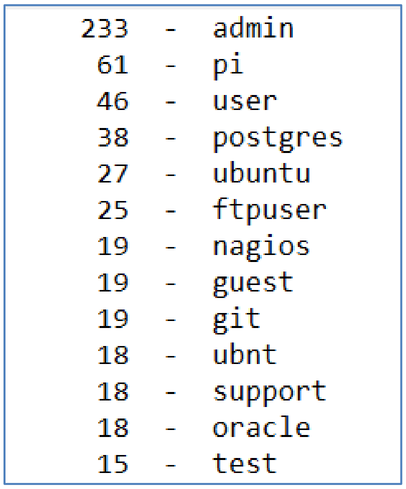
Images from the top tells us that the IP address 50.67.10.67, the one with the high alert, is from Canada, whereas others are located in different parts of world. This seems to be an IP spoofing technique to over flood the system with TCP SYN packets. This type of attack tells us that it is more likely to be "Denial of Service" (DoS) attacks, which can overwhelm computer networks with traffic. Now in this attack, hackers must have spoofed IP addresses to overwhelm computer servers with packets of data, resulting in shutting them down. This geographically dispersed botnets, networks of compromised computers, are often used to send the packets, as we can see from the Snort classifications. Each botnet potentially contains tens of thousands of computers capable of spoofing multiple source IP addresses where such automated attacks are difficult to trace.

Another analysis observed was on audit log file with the help of aureport, as seen below:



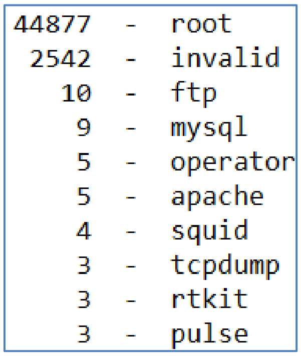
```
[root@psd server-nov-30th]# aureport -if tmp.log

Summary Report
======================
Range of time in logs: 11/25/2018 07:10:16.968 - 11/24/2018 18:30:36.397
Selected time for report: 11/25/2018 07:10:16 - 11/24/2018 18:30:36.397
Number of changes in configuration: 5
Number of changes to accounts, groups, or roles: 37
Number of logins: 4
Number of failed logins: 50
Number of authentications: 7758
Number of failed authentications: 68
Number of users: 2
Number of terminals: 3
Number of host names: 75
Number of executables: 5
Number of commands: 1
Number of files: 0
Number of AVC's: 0
Number of MAC events: 3879
Number of failed syscalls: 0
Number of anomaly events: 0
Number of responses to anomaly events: 0
Number of crypto events: 55202
Number of integrity events: 0
Number of virt events: 0
Number of keys: 0
Number of process IDs: 11602
Number of events: 111281
```

From the aureport, we can see that the most ran activities/attacks are the number of authentications, failed logins, and failed authentications, which led me to analyze secure log file. Analyzing secure log gave me the total number of failed attempts to login with the top IP address of 58.218.92.47 and the ones for accepted logins with the top IP address of 217.23.12.125.



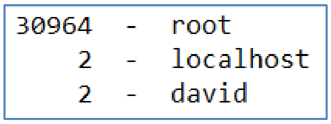
```
    233  -  admin
     61  -  pi
     46  -  user
     38  -  postgres
     27  -  ubuntu
     25  -  ftpuser
     19  -  nagios
     19  -  guest
     19  -  git
     18  -  ubnt
     18  -  support
     18  -  oracle
     15  -  test
```

Invalid Users



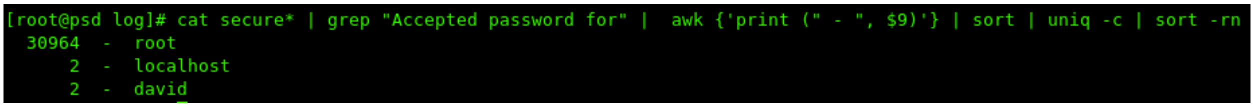
```
44877  -  root
 2542  -  invalid
   10  -  ftp
    9  -  mysql
    5  -  operator
    5  -  apache
    4  -  squid
    3  -  tcpdump
    3  -  rtkit
    3  -  pulse
```

Failed Login Attempts



```
30964  -  root
    2  -  localhost
    2  -  david
```

Accepted Login Attempts



```
[root@psd log]# cat secure* | grep "Accepted password for" |  awk {'print (" - ", $9)'} | sort | uniq -c | sort -rn
  30964  -   root
      2  -   localhost
      2  -   david
```

Conclusion

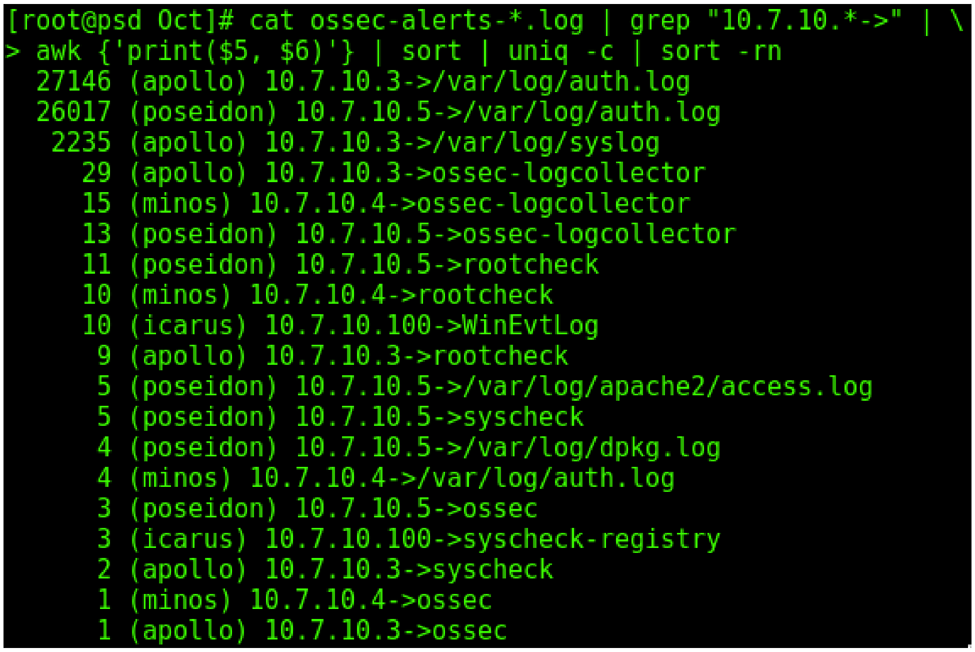
The analyzed audit log file tells us that the IP addresses 58.218.92.47 and 217.23.12.125, are at a high risk that must have had brute force attack and break-in through backdoors. This system had many Denial of Service (DoS) and reconnaissance attacks, including information leak, and break-ins which could have been brought to a stop if proper measures were taken for monitoring networks for a typical activity, deploying packet filtering to detect inconsistencies, including outgoing packets with the source IP addresses, using robust verification methods, authenticating all IP addresses, and using a network attack blocker.

Network 2

Tools used: Linux commands, Wireshark, Splunk Cloud

Analysis
By analyzing and browsing through this network and its directories, it could tell me that this network is using OSSEC generated alerts as its intrusion detection system, under the folder named “labrys”. Another file located under its directories was of packet captures. However, as I noticed IDS Alerts folder, I wanted to get its insights by analyzing through the alerts directories, containing OSSEC alerts.

As I ran through the filters for the type of machine or the system its network is running from, I found out that there are more than one hostname (machine/system) which existed as a private network.

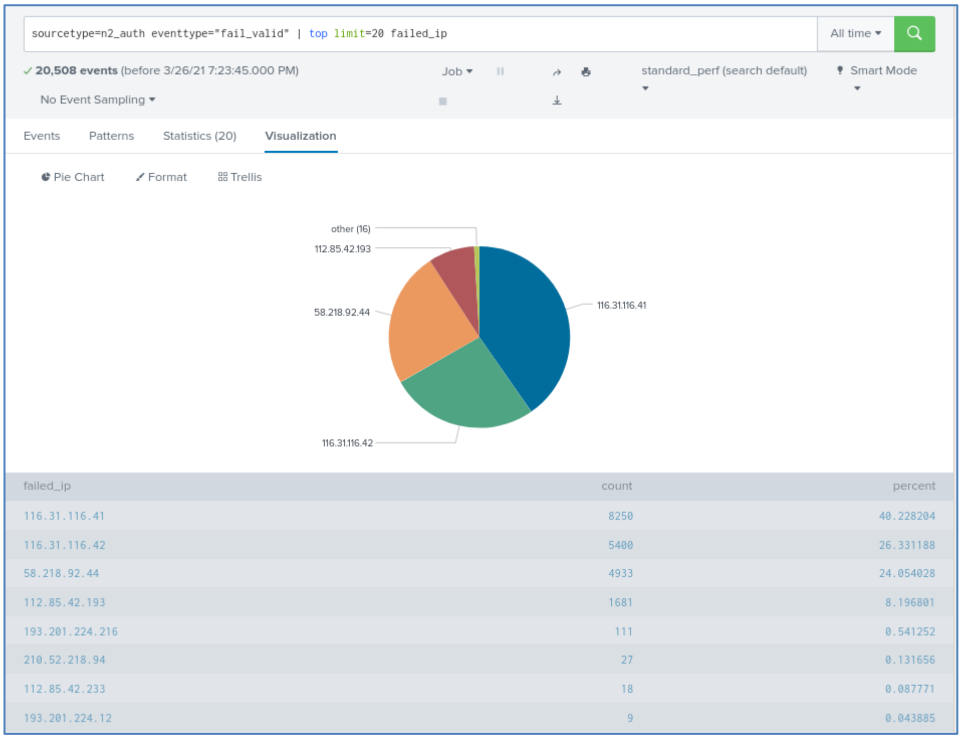


```
[root@psd Oct]# cat ossec-alerts-*.log | grep "10.7.10.*->" | \
> awk {'print($5, $6)'} | sort | uniq -c | sort -rn
  27146 (apollo) 10.7.10.3->/var/log/auth.log
  26017 (poseidon) 10.7.10.5->/var/log/auth.log
   2235 (apollo) 10.7.10.3->/var/log/syslog
     29 (apollo) 10.7.10.3->ossec-logcollector
     15 (minos) 10.7.10.4->ossec-logcollector
     13 (poseidon) 10.7.10.5->ossec-logcollector
     11 (poseidon) 10.7.10.5->rootcheck
     10 (minos) 10.7.10.4->rootcheck
     10 (icarus) 10.7.10.100->WinEvtLog
      9 (apollo) 10.7.10.3->rootcheck
      5 (poseidon) 10.7.10.5->/var/log/apache2/access.log
      5 (poseidon) 10.7.10.5->syscheck
      4 (poseidon) 10.7.10.5->/var/log/dpkg.log
      4 (minos) 10.7.10.4->/var/log/auth.log
      3 (poseidon) 10.7.10.5->ossec
      3 (icarus) 10.7.10.100->syscheck-registry
      2 (apollo) 10.7.10.3->syscheck
      1 (minos) 10.7.10.4->ossec
      1 (apollo) 10.7.10.3->ossec
```

I was able to filter those systems’ names with its IP addresses, which are as listed:

- Apollo – (10.7.10.3)
- Poseidon – (10.7.10.5)
- Minos – (10.7.10.4)
- Icarus – (10.7.10.100)

To dig deeper, I went on analyzing ossec alerts with the failed IP addresses, as captured below:

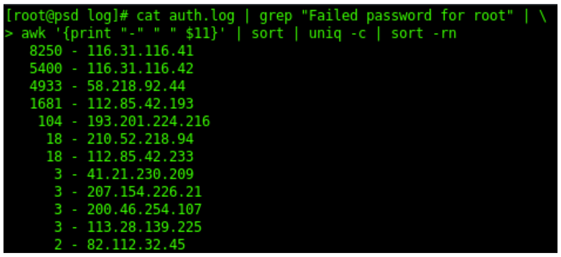


```
sourcetype=n2_auth eventtype="fail_valid" | top limit=20 failed_ip
```

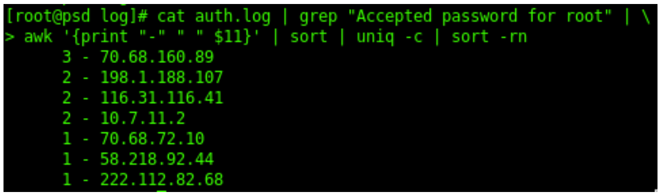
failed_ip	count	percent
116.31.116.41	8250	40.228204
116.31.116.42	5400	26.331188
58.218.92.44	4933	24.054028
112.85.42.193	1681	8.196801
193.201.224.216	111	0.541252
210.52.218.94	27	0.131656
112.85.42.233	18	0.087771
193.201.224.12	9	0.043885

The graph and statistics above show us that the most attempted IP addresses with no success were (116.31.116.41), (116.31.116.42) AND (58.218.92.44). The percentage for the IP under (116.31.116.0) subnet generated is about 40% of the total failed attempts followed by 24%-26%.

Next, I wanted to double check these alerts by investigating under alert and auth logs, which would give me confidence on the findings found

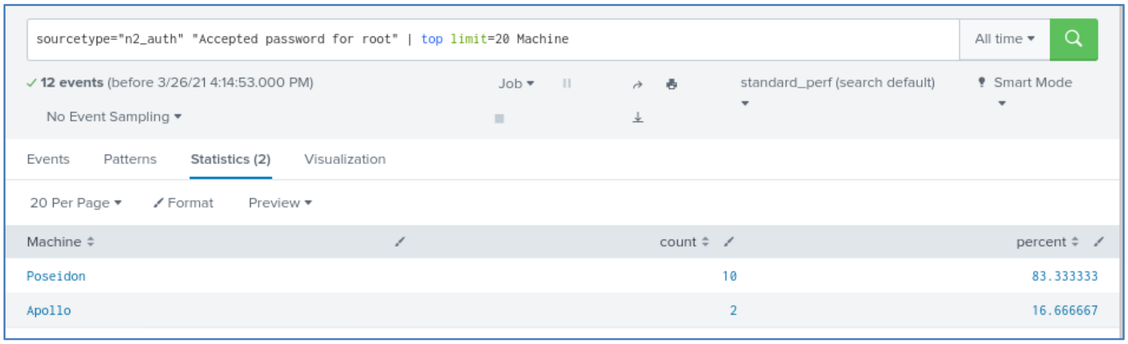


```
[root@psd log]# cat auth.log | grep "Failed password for root" | \
> awk '{print "-" " " $11}' | sort | uniq -c | sort -rn
   8250 - 116.31.116.41
   5400 - 116.31.116.42
   4933 - 58.218.92.44
   1681 - 112.85.42.193
    104 - 193.201.224.216
     18 - 210.52.218.94
     18 - 112.85.42.233
      3 - 41.21.230.209
      3 - 207.154.226.21
      3 - 200.46.254.107
      3 - 113.28.139.225
      2 - 82.112.32.45
```



```
[root@psd log]# cat auth.log | grep "Accepted password for root" | \
> awk '{print "-" " " $11}' | sort | uniq -c | sort -rn
      3 - 70.68.160.89
      2 - 198.1.188.107
      2 - 116.31.116.41
      2 - 10.7.11.2
      1 - 70.68.72.10
      1 - 58.218.92.44
      1 - 222.112.82.68
```

IP addresses with failed and accepted password attempts are of (116.31.116.41) AND (58.218.92.44), from auth log

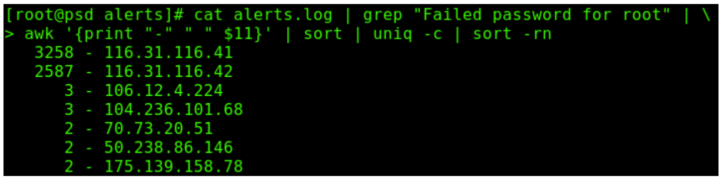


```
sourcetype="n2_auth" "Accepted password for root" | top limit=20 Machine
```

✓ 12 events (before 3/26/21 4:14:53.000 PM)

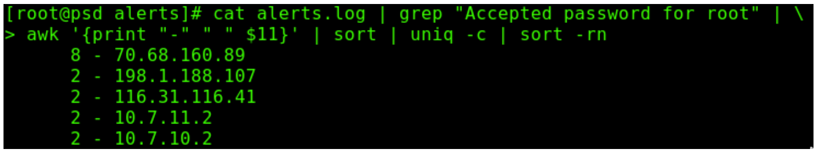
Machine	count	percent
Poseidon	10	83.333333
Apollo	2	16.666667

Auth Log with accepted attempts for Poseidon and Apollo
(To find more on the machines and their uses, look at the Appendix.)



```
[root@psd alerts]# cat alerts.log | grep "Failed password for root" | \
> awk '{print "-" " " $11}' | sort | uniq -c | sort -rn
   3258 - 116.31.116.41
   2587 - 116.31.116.42
      3 - 106.12.4.224
      3 - 104.236.101.68
      2 - 70.73.20.51
      2 - 50.238.86.146
      2 - 175.139.158.78
```

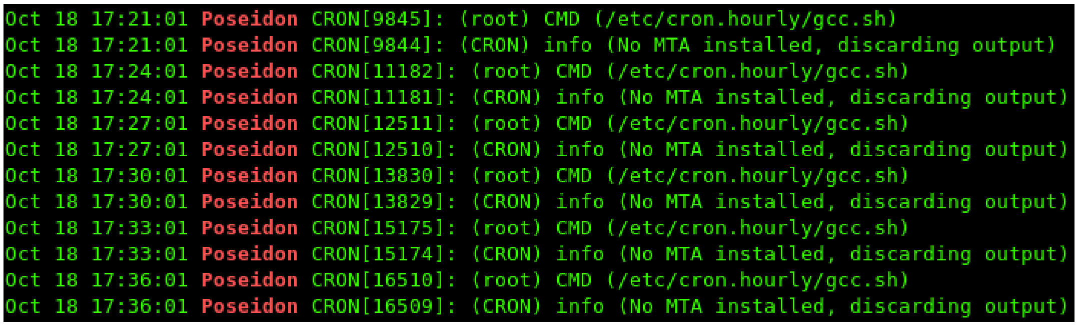
Alert Log with most failed IP for (116.31.116.41)



```
[root@psd alerts]# cat alerts.log | grep "Accepted password for root" | \
> awk '{print "-" " " $11}' | sort | uniq -c | sort -rn
      8 - 70.68.160.89
      2 - 198.1.188.107
      2 - 116.31.116.41
      2 - 10.7.11.2
      2 - 10.7.10.2
```

Alert Log with most failed IP for (116.31.116.41)

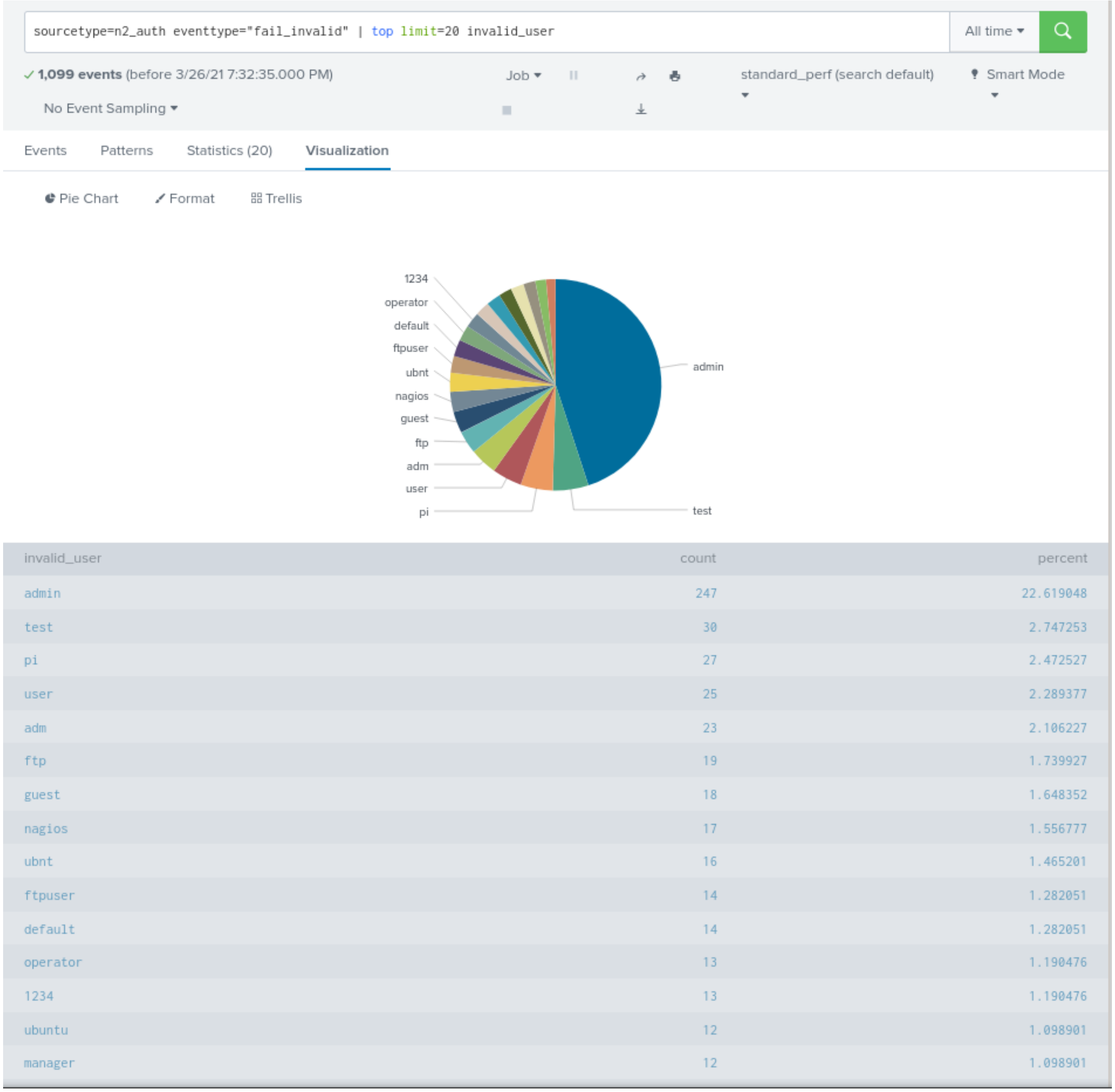
Based on the findings above, I declared IP addresses (116.31.116.41) AND (58.218.92.44) with the highest risk and analyzed further. Another noted observation was that these attacks happened in a very short period of time which tells me that there was a script involved to do so. For which, I started analyzing CRON log file.



```
Oct 18 17:21:01 Poseidon CRON[9845]: (root) CMD (/etc/cron.hourly/gcc.sh)
Oct 18 17:21:01 Poseidon CRON[9844]: (CRON) info (No MTA installed, discarding output)
Oct 18 17:24:01 Poseidon CRON[11182]: (root) CMD (/etc/cron.hourly/gcc.sh)
Oct 18 17:24:01 Poseidon CRON[11181]: (CRON) info (No MTA installed, discarding output)
Oct 18 17:27:01 Poseidon CRON[12511]: (root) CMD (/etc/cron.hourly/gcc.sh)
Oct 18 17:27:01 Poseidon CRON[12510]: (CRON) info (No MTA installed, discarding output)
Oct 18 17:30:01 Poseidon CRON[13830]: (root) CMD (/etc/cron.hourly/gcc.sh)
Oct 18 17:30:01 Poseidon CRON[13829]: (CRON) info (No MTA installed, discarding output)
Oct 18 17:33:01 Poseidon CRON[15175]: (root) CMD (/etc/cron.hourly/gcc.sh)
Oct 18 17:33:01 Poseidon CRON[15174]: (CRON) info (No MTA installed, discarding output)
Oct 18 17:36:01 Poseidon CRON[16510]: (root) CMD (/etc/cron.hourly/gcc.sh)
Oct 18 17:36:01 Poseidon CRON[16509]: (CRON) info (No MTA installed, discarding output)
```

CRON log with assigned *gcc.sh* script to be installed

The finding above shows us that there must have been a script running to breach into the system with many tries in a short amount of time. Furthermore, here is another example of the invalid users prompted to breach into with password failures, as captured below:



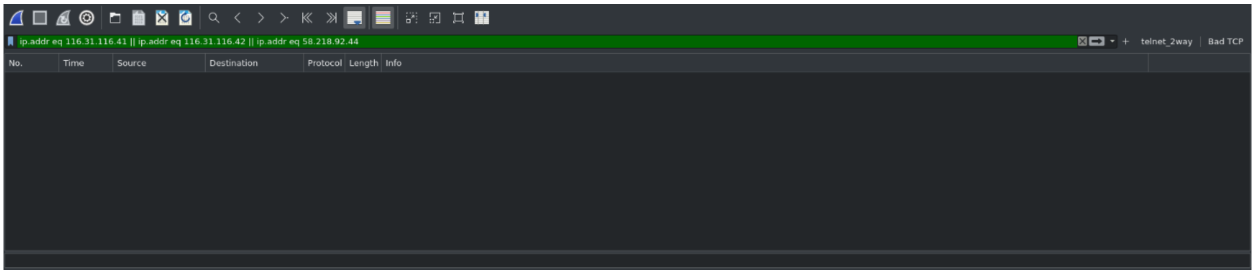
```
sourcetype=n2_auth eventtype="fail_invalid" | top limit=20 invalid_user
```

✓ **1,099 events** (before 3/26/21 7:32:35.000 PM)

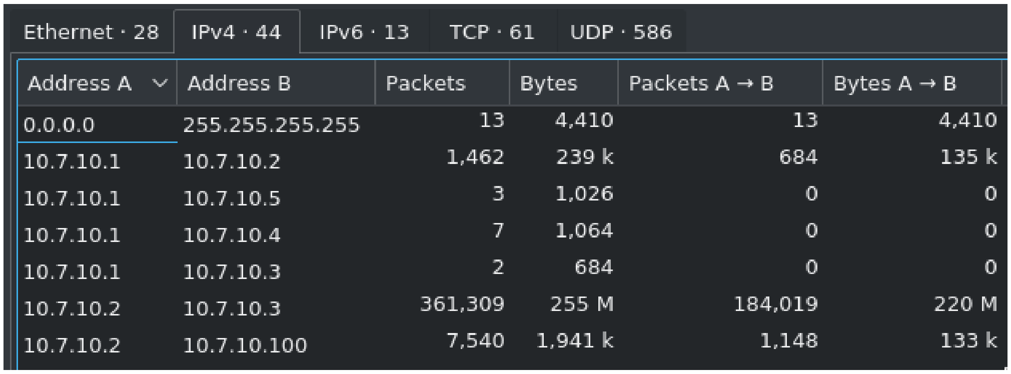
invalid_user	count	percent
admin	247	22.619048
test	30	2.747253
pi	27	2.472527
user	25	2.289377
adm	23	2.106227
ftp	19	1.739927
guest	18	1.648352
nagios	17	1.556777
ubnt	16	1.465201
ftpuser	14	1.282051
default	14	1.282051
operator	13	1.190476
1234	13	1.190476
ubuntu	12	1.098901
manager	12	1.098901

Invalid usernames used are seemed to be very common the reason these were being declared in a script to run for in order to get into the system, out of which root was accepted as seen before.

To further investigate, I entered the IP addresses, with the highest risk, into the shared packet capture but was not successful in finding any conversation between those IP addresses. That means, the attacker or the script is making sure to delete any traces for those IP addresses.

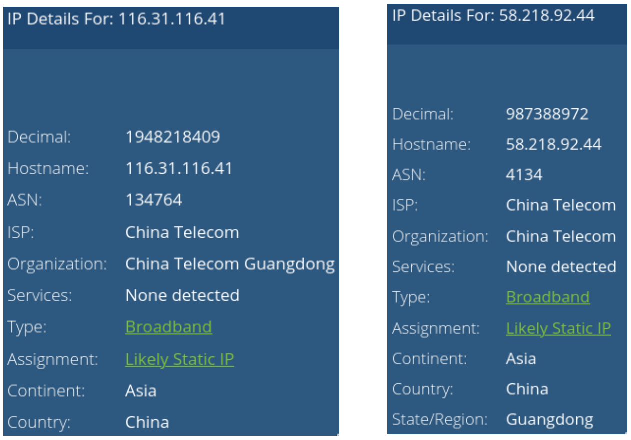


```
ip.addr eq 116.31.116.41 || ip.addr eq 116.31.116.42 || ip.addr eq 58.218.92.44
```



Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B
0.0.0.0	255.255.255.255	13	4,410	13	4,410
10.7.10.1	10.7.10.2	1,462	239 k	684	135 k
10.7.10.1	10.7.10.5	3	1,026	0	0
10.7.10.1	10.7.10.4	7	1,064	0	0
10.7.10.1	10.7.10.3	2	684	0	0
10.7.10.2	10.7.10.3	361,309	255 M	184,019	220 M
10.7.10.2	10.7.10.100	7,540	1,941 k	1,148	133 k

Scanning through the packet captures, I was only able to retrieve the conversations being done between the hosts, and only found UDP traffic being flown. Wireshark captures does not seem to have any bad traffic, information leaking, or malicious activity of any type.
IP addresses with high alerts (116.31.116.41) AND (58.218.92.44) were then looked up on the web with the help of "whatismyipaddress" to get their locations, as we can see below:



IP Details For: 116.31.116.41

Decimal:	1948218409
Hostname:	116.31.116.41
ASN:	134764
ISP:	China Telecom
Organization:	China Telecom Guangdong
Services:	None detected
Type:	Broadband
Assignment:	Likely Static IP
Continent:	Asia
Country:	China

IP Details For: 58.218.92.44

Decimal:	987388972
Hostname:	58.218.92.44
ASN:	4134
ISP:	China Telecom
Organization:	China Telecom
Services:	None detected
Type:	Broadband
Assignment:	Likely Static IP
Continent:	Asia
Country:	China
State/Region:	Guangdong

IP addresses showing locations for China

Conclusion

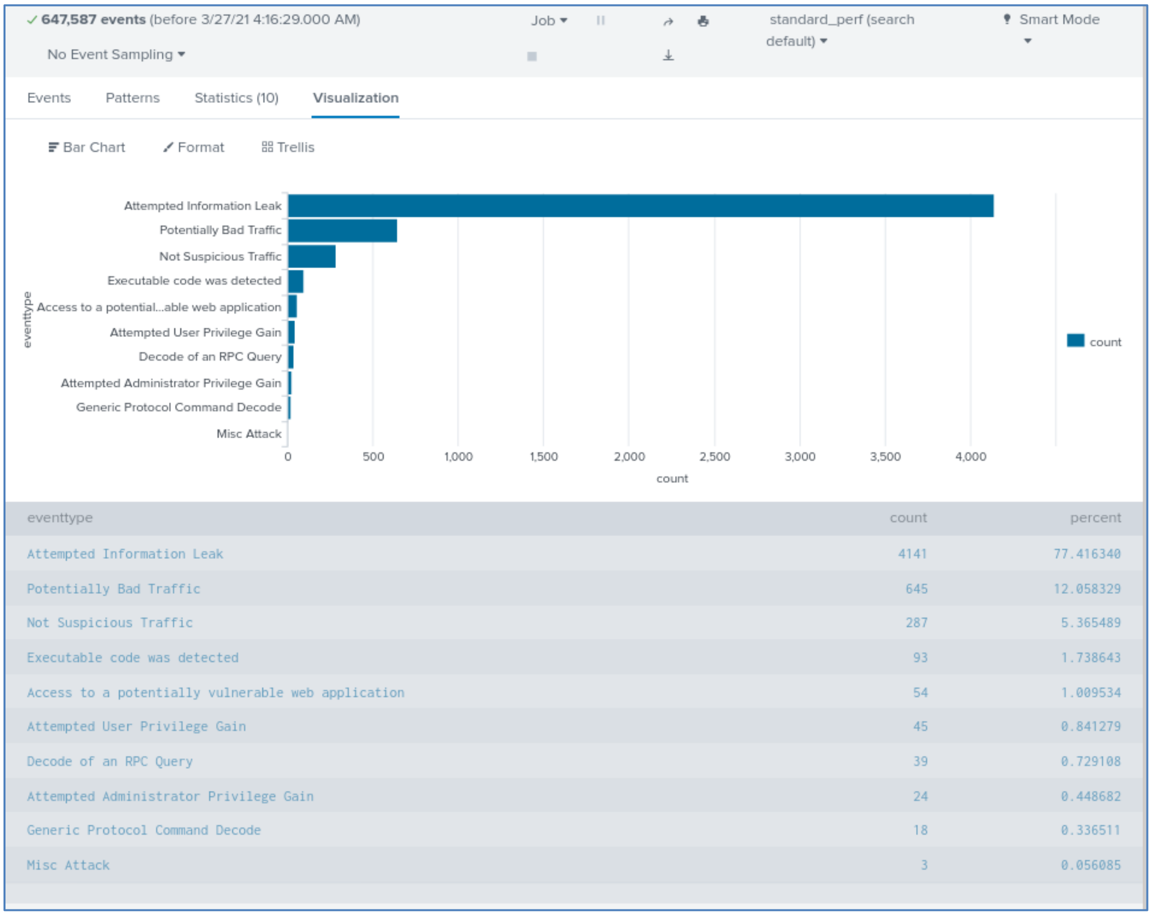
The analyzed auth log and alert logs (ossec, alerts) tells us that the IP addresses (116.31.116.41) AND (58.218.92.44), are at a high risk which are most likely to have SSH Brute force attack. Sadly, OSSEC IDS was unable to stop these attacks from happening. This attack must have breached into the network with the help of gcc.sh malware script, being handled by Poseidon (as we have seen before). OSSEC should have created high end rules to stop such attacks, or even bringing those attacks into an attention by activating them as they happen. To make the network more secure, passwords must be difficult to discover, where IT administrations should enforce strict password policies with minimum length and complexity requirements.

Network 3

Tools used: Linux commands, Snort, SnortSnarf, Argus, Wireshark, Splunk Cloud

Analysis

The machines involved in this network had large number of snort logs that presented the types of activities, classifications, being transferred. The data was mostly being captured within a database. Listed below are the classifications being rendered through this network:

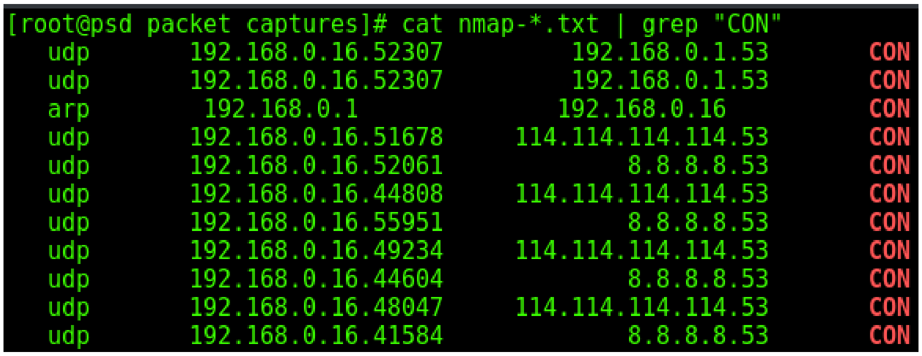


✓ 647,587 events (before 3/27/21 4:16:29.000 AM)

eventtype	count	percent
Attempted Information Leak	4141	77.416340
Potentially Bad Traffic	645	12.058329
Not Suspicious Traffic	287	5.365489
Executable code was detected	93	1.738643
Access to a potentially vulnerable web application	54	1.009534
Attempted User Privilege Gain	45	0.841279
Decode of an RPC Query	39	0.729108
Attempted Administrator Privilege Gain	24	0.448682
Generic Protocol Command Decode	18	0.336511
Misc Attack	3	0.056085

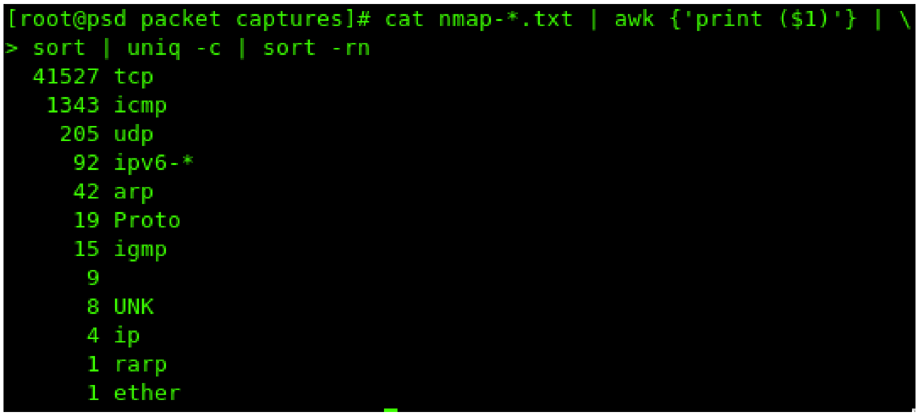
By the finding above, we can see that the most activities being generated within this network are of information leak, potential bad traffic, and an executable code. To investing further I went on transferring the packets in the .ra files and then the text (as attached with the project).

Here are some findings from the argus filtered files:



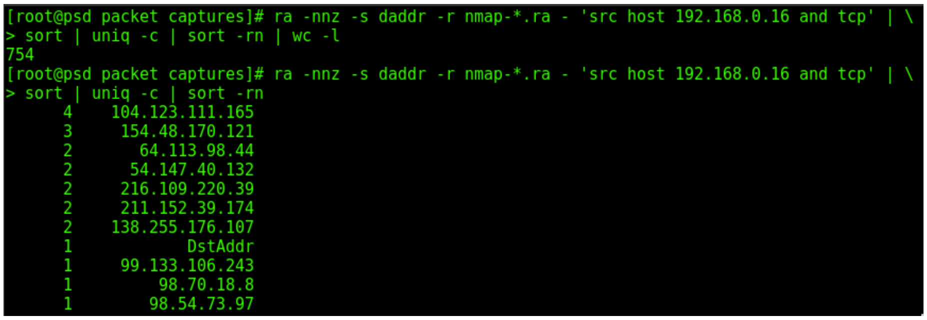
```
[root@psd packet captures]# cat nmap-*.txt | grep "CON"
   udp        192.168.0.16.52307        192.168.0.1.53       CON
   udp        192.168.0.16.52307        192.168.0.1.53       CON
   arp         192.168.0.1             192.168.0.16          CON
   udp        192.168.0.16.51678    114.114.114.114.53       CON
   udp        192.168.0.16.52061            8.8.8.8.53       CON
   udp        192.168.0.16.44808    114.114.114.114.53       CON
   udp        192.168.0.16.55951            8.8.8.8.53       CON
   udp        192.168.0.16.49234    114.114.114.114.53       CON
   udp        192.168.0.16.44604            8.8.8.8.53       CON
   udp        192.168.0.16.48047    114.114.114.114.53       CON
   udp        192.168.0.16.41584            8.8.8.8.53       CON
```

Several systems have responded with the port scanners



```
[root@psd packet captures]# cat nmap-*.txt | awk {'print ($1)'} | \
> sort | uniq -c | sort -rn
  41527 tcp
   1343 icmp
    205 udp
     92 ipv6-*
     42 arp
     19 Proto
     15 igmp
      9
      8 UNK
      4 ip
      1 rarp
      1 ether
```

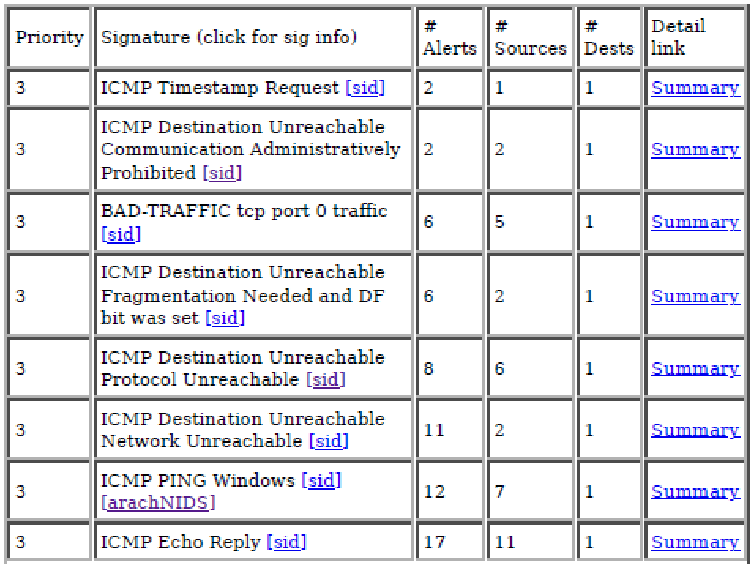
Amount of protocols used, TCP being the highest



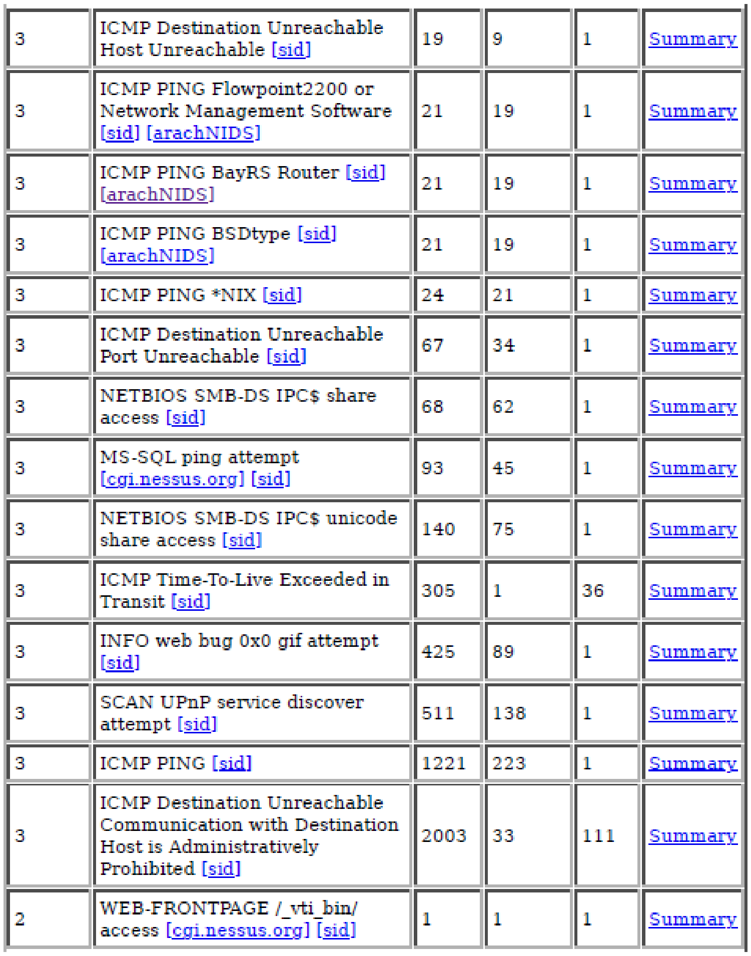
```
[root@psd packet captures]# ra -nnz -s daddr -r nmap-*.ra - 'src host 192.168.0.16 and tcp' | \
> sort | uniq -c | sort -rn | wc -l
754
[root@psd packet captures]# ra -nnz -s daddr -r nmap-*.ra - 'src host 192.168.0.16 and tcp' | \
> sort | uniq -c | sort -rn
      4    104.123.111.165
      3     154.48.170.121
      2       64.113.98.44
      2      54.147.40.132
      2     216.109.220.39
      2     211.152.39.174
      2    138.255.176.107
      1            DstAddr
      1     99.133.106.243
      1         98.70.18.8
      1        98.54.73.97
```

Appendix

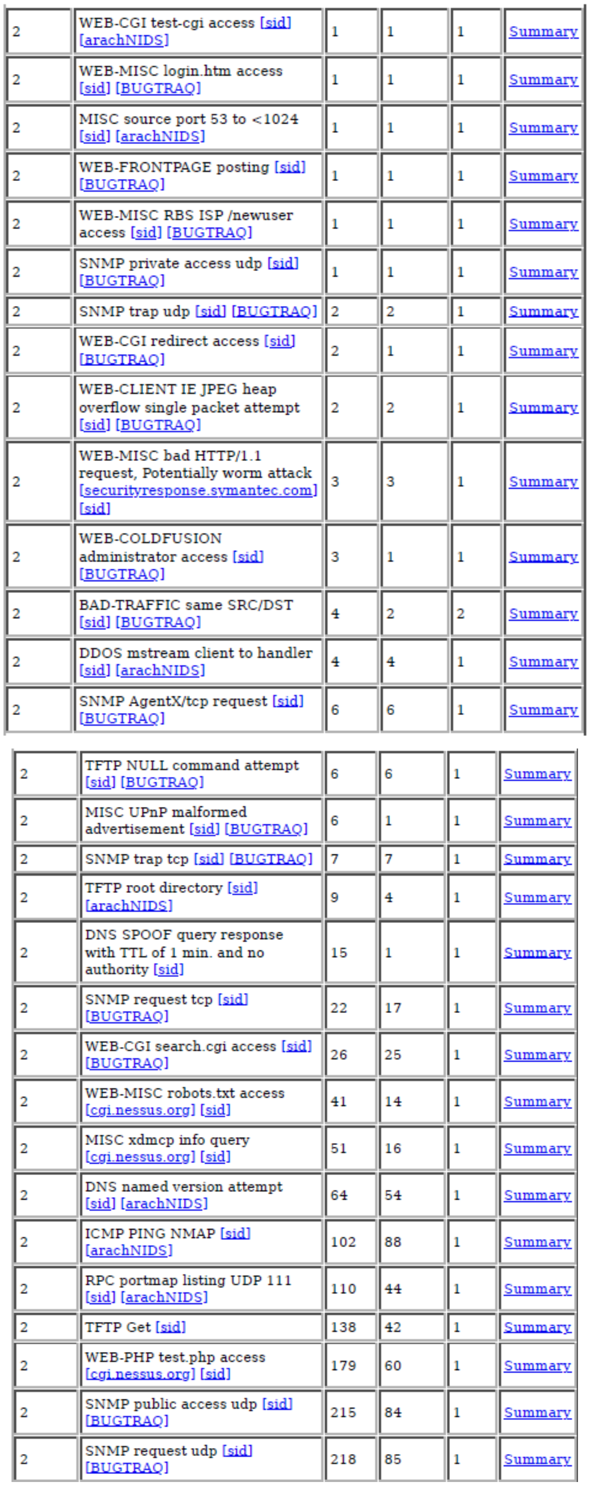
Network 1



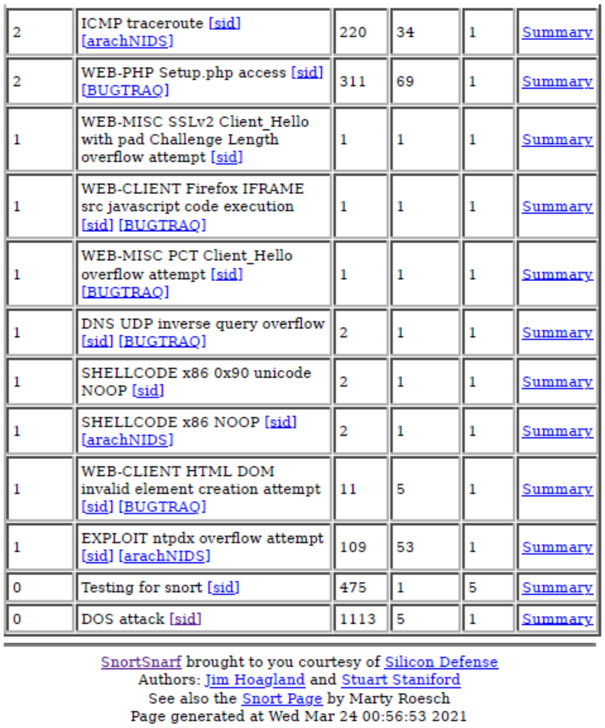
Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests	Detail link
3	ICMP Timestamp Request [sid]	2	1	1	Summary
3	ICMP Destination Unreachable Communication Administratively Prohibited [sid]	2	2	1	Summary
3	BAD-TRAFFIC tcp port 0 traffic [sid]	6	5	1	Summary
3	ICMP Destination Unreachable Fragmentation Needed and DF bit was set [sid]	6	2	1	Summary
3	ICMP Destination Unreachable Protocol Unreachable [sid]	8	6	1	Summary
3	ICMP Destination Unreachable Network Unreachable [sid]	11	2	1	Summary
3	ICMP PING Windows [sid] [arachNIDS]	12	7	1	Summary
3	ICMP Echo Reply [sid]	17	11	1	Summary



3	ICMP Destination Unreachable Host Unreachable [sid]	19	9	1	Summary
3	ICMP PING Flowpoint2200 or Network Management Software [sid] [arachNIDS]	21	19	1	Summary
3	ICMP PING BayRS Router [sid] [arachNIDS]	21	19	1	Summary
3	ICMP PING BSDtype [sid] [arachNIDS]	21	19	1	Summary
3	ICMP PING *NIX [sid]	24	21	1	Summary
3	ICMP Destination Unreachable Port Unreachable [sid]	67	34	1	Summary
3	NETBIOS SMB-DS IPC$ share access [sid]	68	62	1	Summary
3	MS-SQL ping attempt [cgi.nessus.org] [sid]	93	45	1	Summary
3	NETBIOS SMB-DS IPC$ unicode share access [sid]	140	75	1	Summary
3	ICMP Time-To-Live Exceeded in Transit [sid]	305	1	36	Summary
3	INFO web bug 0x0 gif attempt [sid]	425	89	1	Summary
3	SCAN UPnP service discover attempt [sid]	511	138	1	Summary
3	ICMP PING [sid]	1221	223	1	Summary
3	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited [sid]	2003	33	111	Summary
2	WEB-FRONTPAGE /_vti_bin/ access [cgi.nessus.org] [sid]	1	1	1	Summary



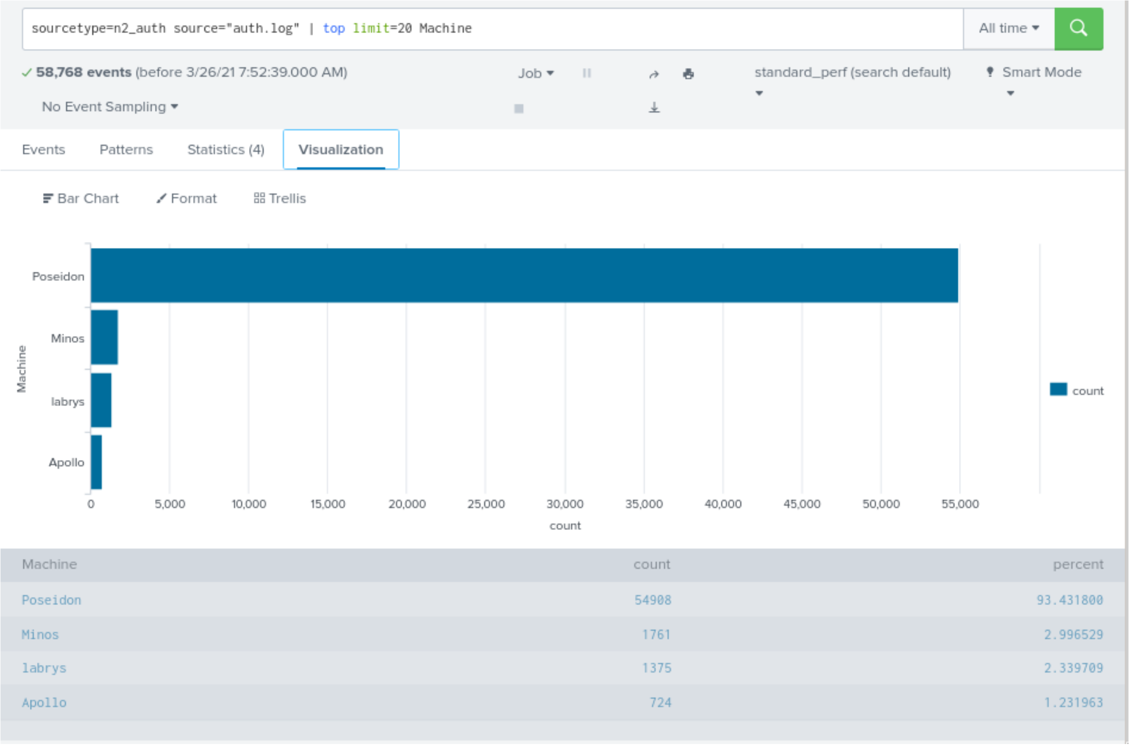
2	WEB-CGI test-cgi access [sid] [arachNIDS]	1	1	1	Summary
2	WEB-MISC login.htm access [sid] [BUGTRAQ]	1	1	1	Summary
2	MISC source port 53 to <1024 [sid] [arachNIDS]	1	1	1	Summary
2	WEB-FRONTPAGE posting [sid] [BUGTRAQ]	1	1	1	Summary
2	WEB-MISC RBS ISP /newuser access [sid] [BUGTRAQ]	1	1	1	Summary
2	SNMP private access udp [sid] [BUGTRAQ]	1	1	1	Summary
2	SNMP trap udp [sid] [BUGTRAQ]	2	2	1	Summary
2	WEB-CGI redirect access [sid] [BUGTRAQ]	2	1	1	Summary
2	WEB-CLIENT IE JPEG heap overflow single packet attempt [sid] [BUGTRAQ]	2	2	1	Summary
2	WEB-MISC bad HTTP/1.1 request, Potentially worm attack [securityresponse.symantec.com] [sid]	3	3	1	Summary
2	WEB-COLDFUSION administrator access [sid] [BUGTRAQ]	3	1	1	Summary
2	BAD-TRAFFIC same SRC/DST [sid] [BUGTRAQ]	4	2	2	Summary
2	DDOS mstream client to handler [sid] [arachNIDS]	4	4	1	Summary
2	SNMP AgentX/tcp request [sid] [BUGTRAQ]	6	6	1	Summary
2	TFTP NULL command attempt [sid] [BUGTRAQ]	6	6	1	Summary
2	MISC UPnP malformed advertisement [sid] [BUGTRAQ]	6	1	1	Summary
2	SNMP trap tcp [sid] [BUGTRAQ]	7	7	1	Summary
2	TFTP root directory [sid] [arachNIDS]	9	4	1	Summary
2	DNS SPOOF query response with TTL of 1 min. and no authority [sid]	15	1	1	Summary
2	SNMP request tcp [sid] [BUGTRAQ]	22	17	1	Summary
2	WEB-CGI search.cgi access [sid] [BUGTRAQ]	26	25	1	Summary
2	WEB-MISC robots.txt access [cgi.nessus.org] [sid]	41	14	1	Summary
2	MISC xdmcp info query [cgi.nessus.org] [sid]	51	16	1	Summary
2	DNS named version attempt [sid] [arachNIDS]	64	54	1	Summary
2	ICMP PING NMAP [sid] [arachNIDS]	102	88	1	Summary
2	RPC portmap listing UDP 111 [sid] [arachNIDS]	110	44	1	Summary
2	TFTP Get [sid]	138	42	1	Summary
2	WEB-PHP test.php access [cgi.nessus.org] [sid]	179	60	1	Summary
2	SNMP public access udp [sid] [BUGTRAQ]	215	84	1	Summary
2	SNMP request udp [sid] [BUGTRAQ]	218	85	1	Summary



2	ICMP traceroute [sid] [arachNIDS]	220	34	1	Summary
2	WEB-PHP Setup.php access [sid] [BUGTRAQ]	311	69	1	Summary
1	WEB-MISC SSLv2 Client_Hello with pad Challenge Length overflow attempt [sid]	1	1	1	Summary
1	WEB-CLIENT Firefox IFRAME src javascript code execution [sid] [BUGTRAQ]	1	1	1	Summary
1	WEB-MISC PCT Client_Hello overflow attempt [sid] [BUGTRAQ]	1	1	1	Summary
1	DNS UDP inverse query overflow [sid] [BUGTRAQ]	2	1	1	Summary
1	SHELLCODE x86 0x90 unicode NOOP [sid]	2	1	1	Summary
1	SHELLCODE x86 NOOP [sid] [arachNIDS]	2	1	1	Summary
1	WEB-CLIENT HTML DOM invalid element creation attempt [sid] [BUGTRAQ]	11	5	1	Summary
1	EXPLOIT ntpdx overflow attempt [sid] [arachNIDS]	109	53	1	Summary
0	Testing for snort [sid]	475	1	5	Summary
0	DOS attack [sid]	1113	5	1	Summary

SnortSnarf brought to you courtesy of Silicon Defense
Authors: Jim Hoagland and Stuart Staniford
See also the Snort Page by Marty Roesch
Page generated at Wed Mar 24 00:56:53 2021

Network 2

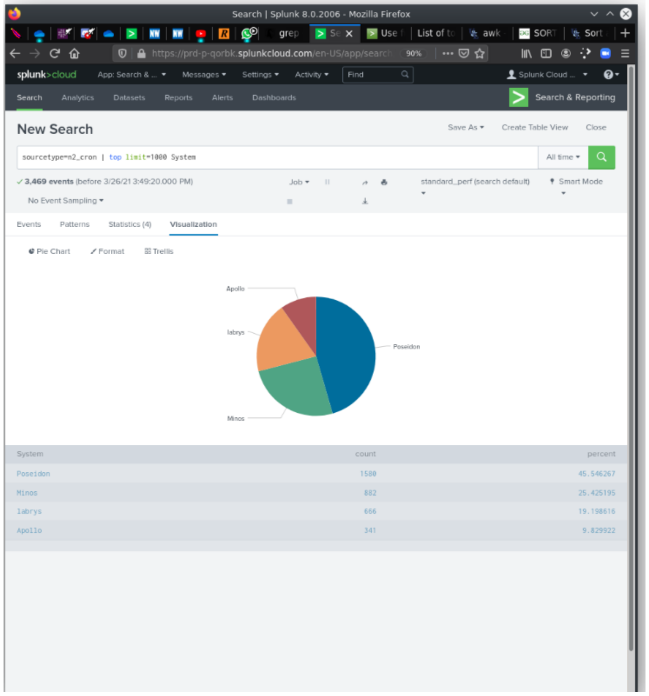


```
sourcetype=n2_auth source="auth.log" | top limit=20 Machine
```

✓ **58,768 events** (before 3/26/21 7:52:39.000 AM)

No Event Sampling ▾

Machine	count	percent
Poseidon	54908	93.431800
Minos	1761	2.996529
labrys	1375	2.339709
Apollo	724	1.231963



New Search

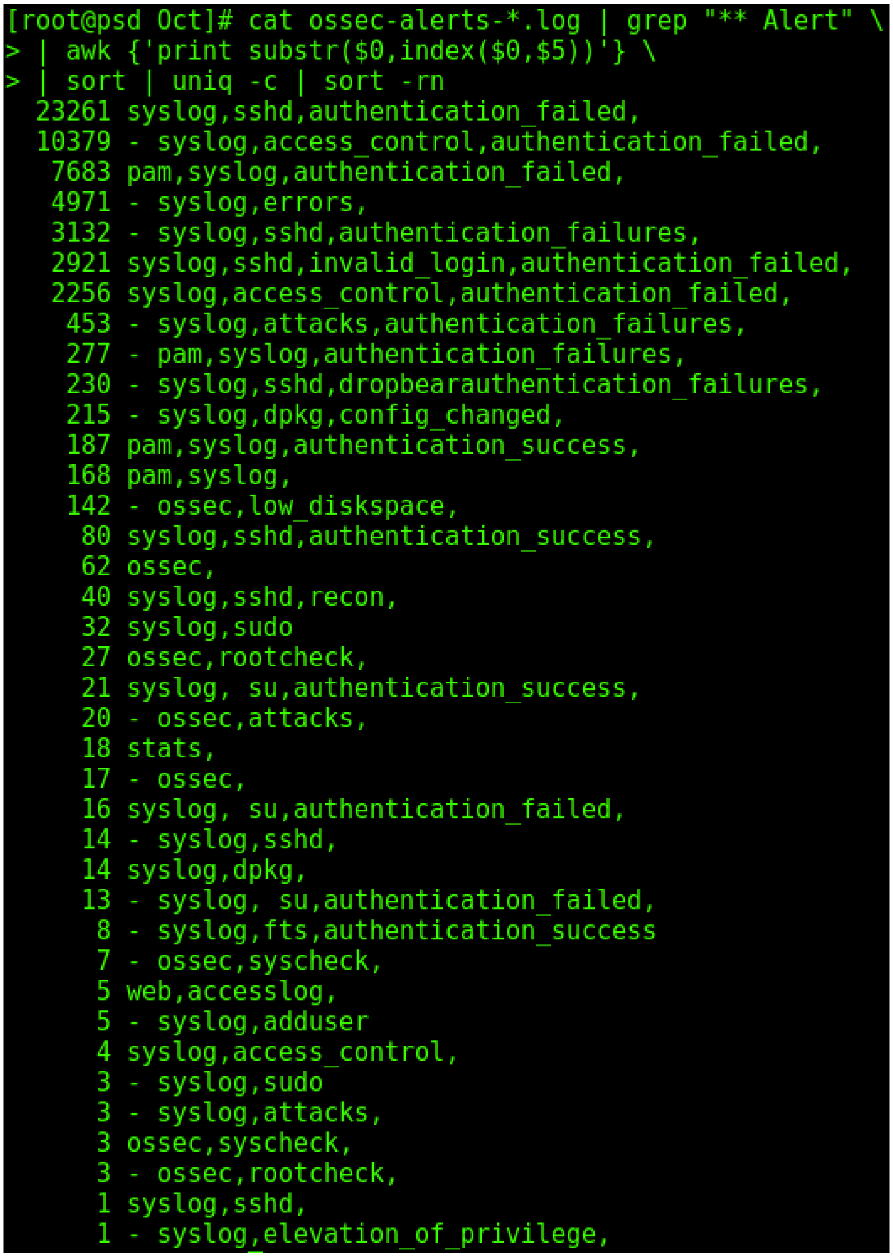
```
sourcetype=n2_cron | top limit=1000 System
```

✓ **3,469 events** (before 3/26/21 3:49:20.000 PM)

No Event Sampling ▾

System	count	percent
Poseidon	1580	45.546267
Minos	882	25.425195
labrys	666	19.198616
Apollo	341	9.829922

Network 3



```
[root@psd Oct]# cat ossec-alerts-*.log | grep "** Alert" \
> | awk {'print substr($0,index($0,$5))'} \
> | sort | uniq -c | sort -rn
  23261 syslog,sshd,authentication_failed,
  10379 - syslog,access_control,authentication_failed,
   7683 pam,syslog,authentication_failed,
   4971 - syslog,errors,
   3132 - syslog,sshd,authentication_failures,
   2921 syslog,sshd,invalid_login,authentication_failed,
   2256 syslog,access_control,authentication_failed,
    453 - syslog,attacks,authentication_failures,
    277 - pam,syslog,authentication_failures,
    230 - syslog,sshd,dropbearauthentication_failures,
    215 - syslog,dpkg,config_changed,
    187 pam,syslog,authentication_success,
    168 pam,syslog,
    142 - ossec,low_diskspace,
     80 syslog,sshd,authentication_success,
     62 ossec,
     40 syslog,sshd,recon,
     32 syslog,sudo
     27 ossec,rootcheck,
     21 syslog, su,authentication_success,
     20 - ossec,attacks,
     18 stats,
     17 - ossec,
     16 syslog, su,authentication_failed,
     14 - syslog,sshd,
     14 syslog,dpkg,
     13 - syslog, su,authentication_failed,
      8 - syslog,fts,authentication_success
      7 - ossec,syscheck,
      5 web,accesslog,
      5 - syslog,adduser
      4 syslog,access_control,
      3 - syslog,sudo
      3 - syslog,attacks,
      3 ossec,syscheck,
      3 - ossec,rootcheck,
      1 syslog,sshd,
      1 - syslog,elevation_of_privilege,
```

References

1. https://us.norton.com/internetsecurity-emerging-threats-dos-attacks-explained.html
2. https://www.loggly.com/ultimate-guide/analyzing-linux-logs/
3. https://www.coresentinel.com/processing-pcap-files-snort/
4. https://github.com/ebdavison/snortsnarf
5. https://openargus.org/using-argus
6. https://www.kaspersky.com/resource-center/threats/ip-spoofing
7. https://www.extrahop.com/resources/attacks/brute-force/