7.1 Breach Report

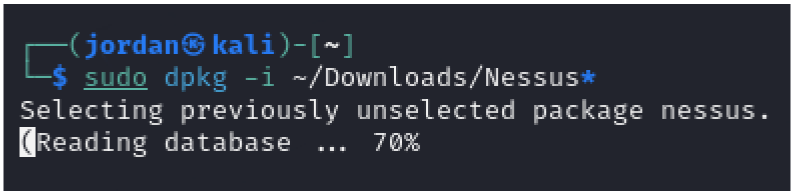
The CrowdStrike 2023 Global Threat Report provides a very good overview of the threat landscape in the year of 2022. This article mainly focuses on nation-state adversaries, eCrime, and news tactics and mitigations. A few events involving nation state threats were Russia, who launched many cyber operations in support of the Ukraine invasion. China, who increased their efforts in espionage and target1ted almost every global industry sector by using zero day vulnerabilities and web facing exploits. Iran and North Korea continue to focus on disruptive cyber operations and theft in terms of cryptocurrency for their own gain.

I also read about the rise of cloud exploitation. Apparently cloud exploit cases rose by 95%, almost double.... as attackers learned more and more misconfigurations and used stolen credentials.
A widely know vulnerability called Log4Shell continued to be widely exploited and adversaries many previously patched systems through rediscovery of known vulnerabilities.

Hacktivists are on the move, particularly in Russia's war with Ukraine. There was a large surge in activity targeting geopolitical entities. This trend is expected to grow even larger.

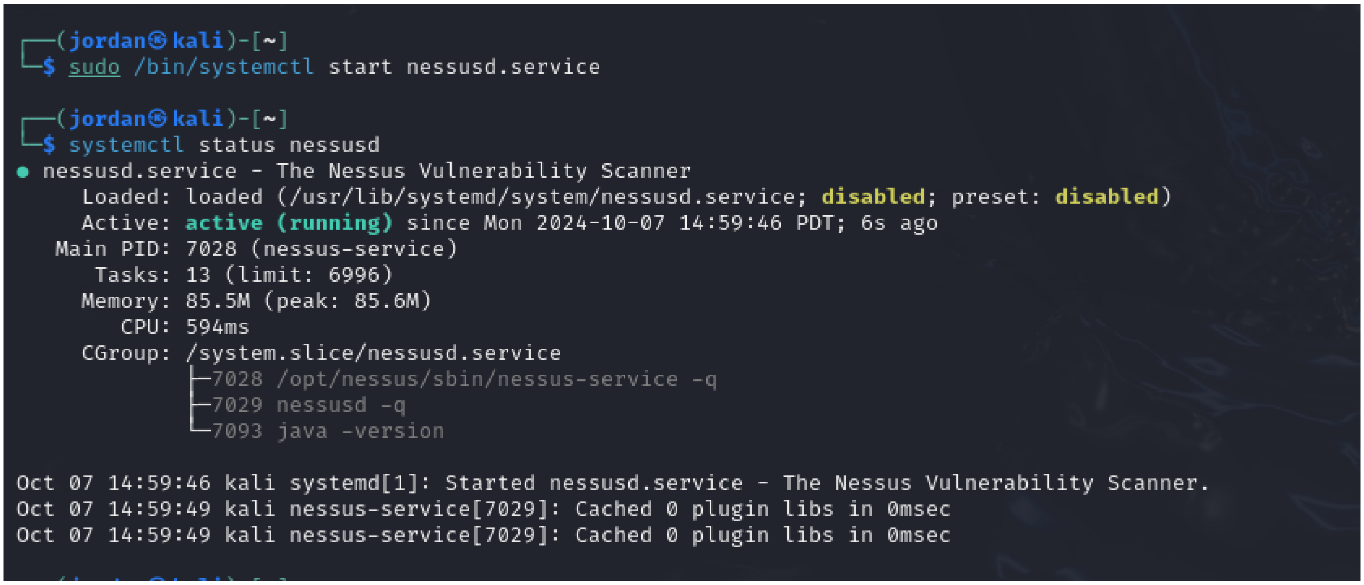
Adversaries will continue to evolve their tactics and more organizations need to adopt a better approach to cybersecurity by improving identity protection, cloud security, and real-time threat monitoring to stay ahead of the adversaries. The main idea of the article is to emphasize the importance of understanding adversaries, leverage cutting edge threat intelligence, and prepare teams through training and practice.

7.2 - Nessus Vulnerability Scan



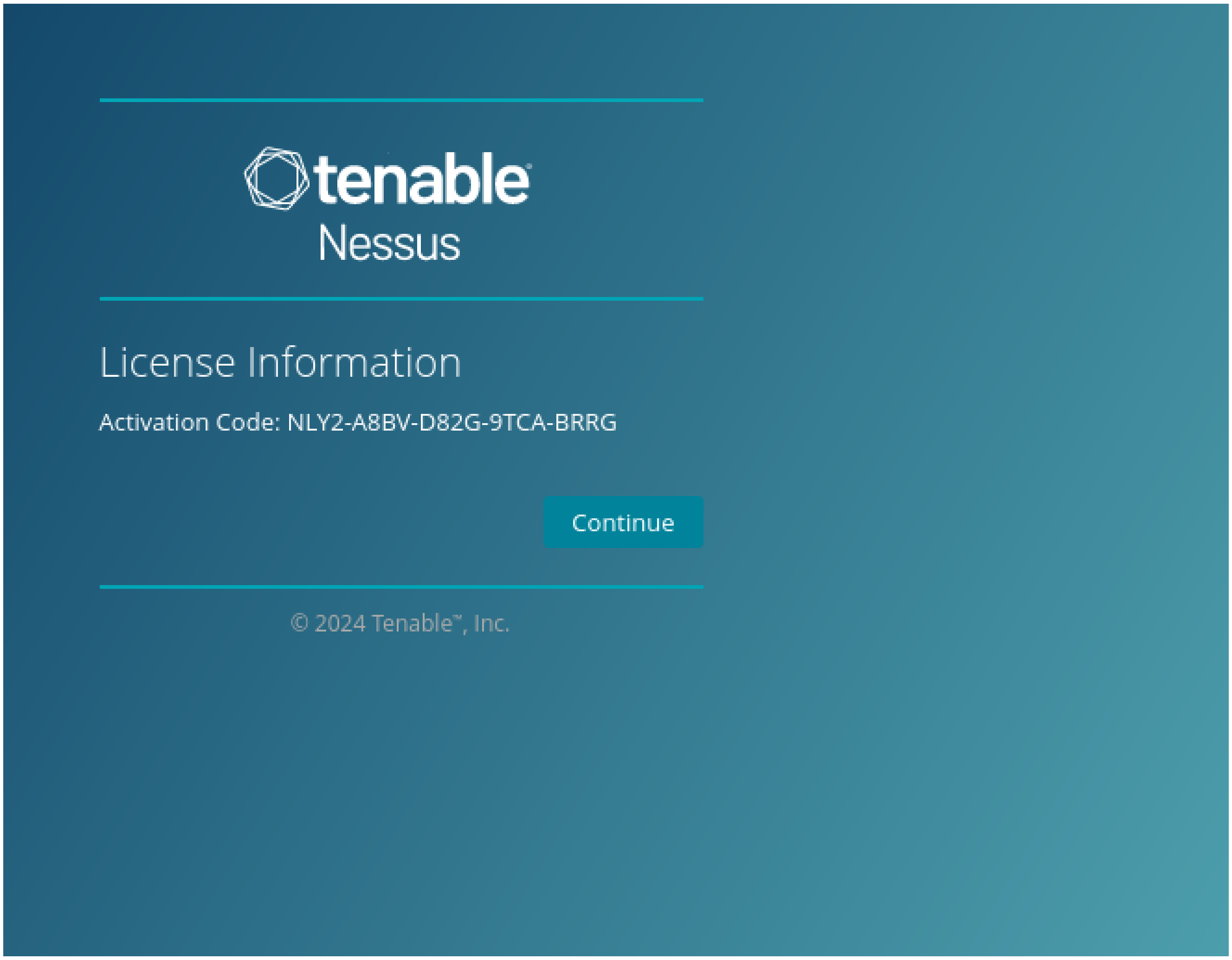
```
┌──(jordan㉿kali)-[~]
└─$ sudo dpkg -i ~/Downloads/Nessus*
Selecting previously unselected package nessus.
(Reading database ... 70%
```

After signing up for Nessus, I completed the installation for the package



```
┌──(jordan㉿kali)-[~]
└─$ sudo /bin/systemctl start nessusd.service

┌──(jordan㉿kali)-[~]
└─$ systemctl status nessusd
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/usr/lib/systemd/system/nessusd.service; disabled; preset: disabled)
     Active: active (running) since Mon 2024-10-07 14:59:46 PDT; 6s ago
   Main PID: 7028 (nessus-service)
      Tasks: 13 (limit: 6996)
     Memory: 85.5M (peak: 85.6M)
        CPU: 594ms
     CGroup: /system.slice/nessusd.service
             ├─7028 /opt/nessus/sbin/nessus-service -q
             ├─7029 nessusd -q
             └─7093 java -version

Oct 07 14:59:46 kali systemd[1]: Started nessusd.service - The Nessus Vulnerability Scanner.
Oct 07 14:59:49 kali nessus-service[7029]: Cached 0 plugin libs in 0msec
Oct 07 14:59:49 kali nessus-service[7029]: Cached 0 plugin libs in 0msec
```

We now have nessus running and can access it on firefox

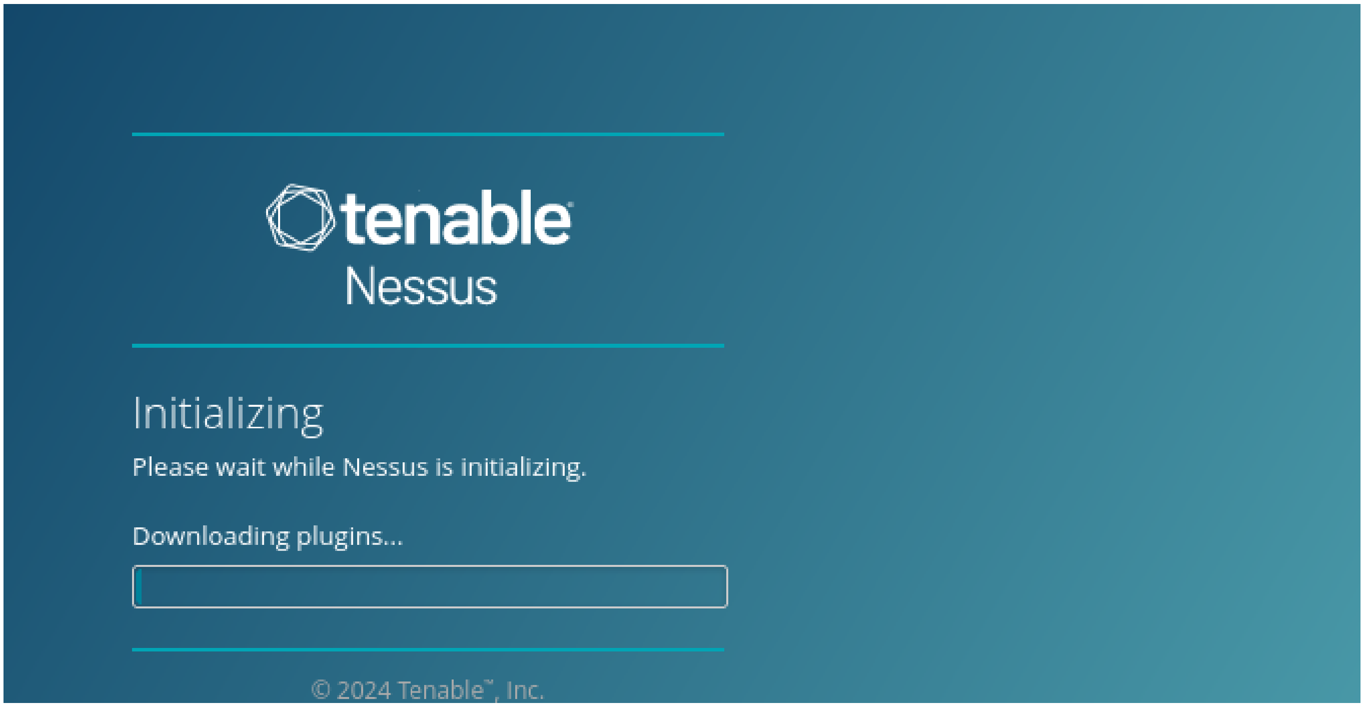


License Information

Activation Code: NLY2-A8BV-D82G-9TCA-BRRG

© 2024 Tenable™, Inc.

Going through the process I was able to put in my activation code for nessus



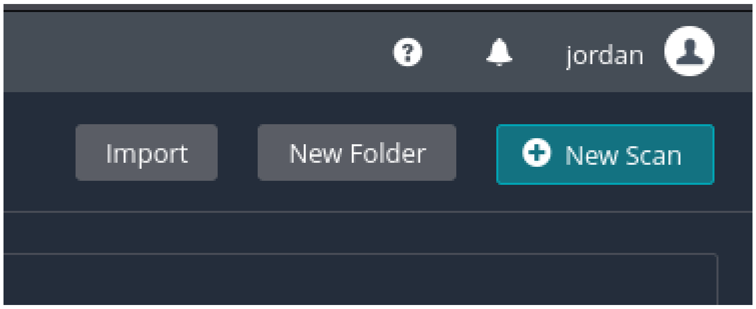
Initializing

Please wait while Nessus is initializing.

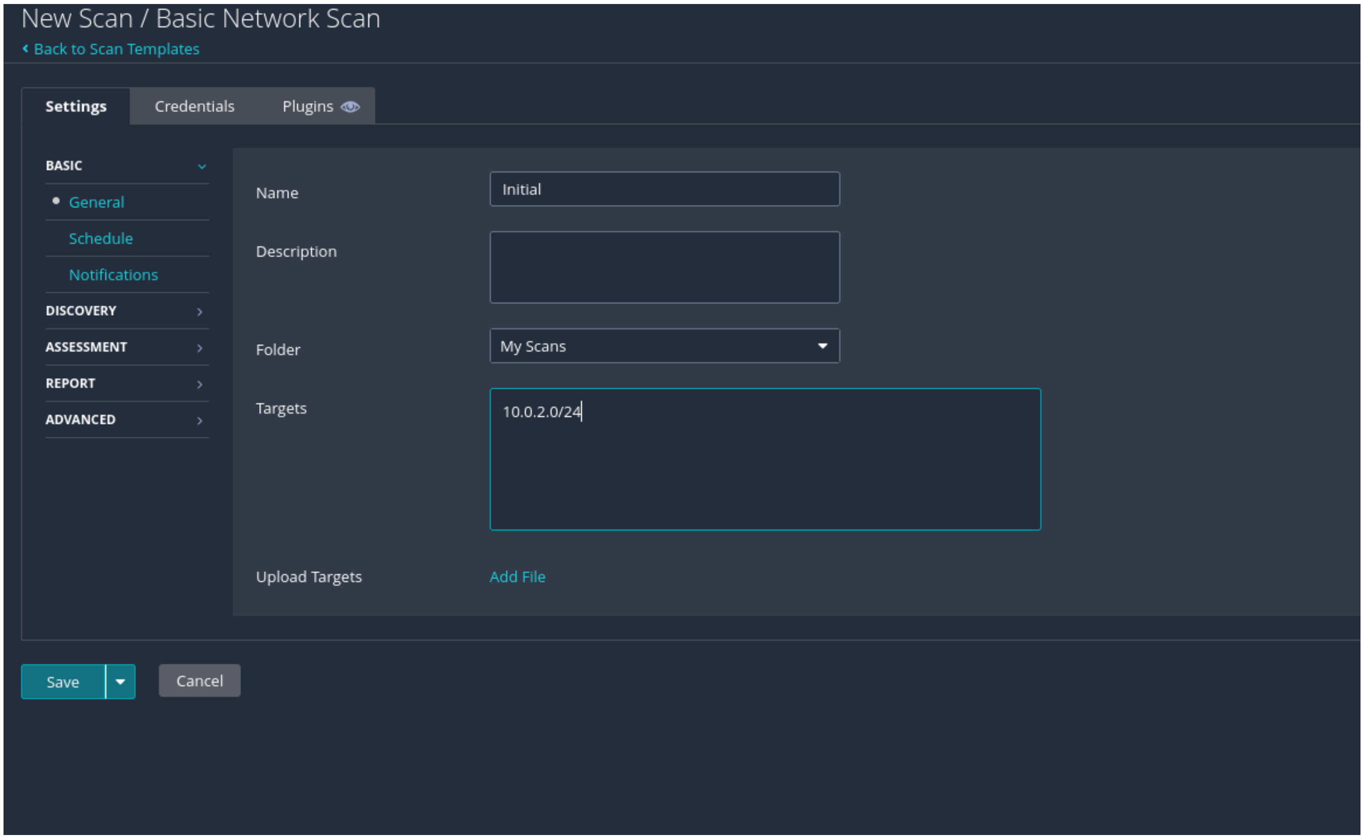
Downloading plugins...

© 2024 Tenable™, Inc.

I am now installing all packages



After a good while of waiting.... we can finally start a scan!



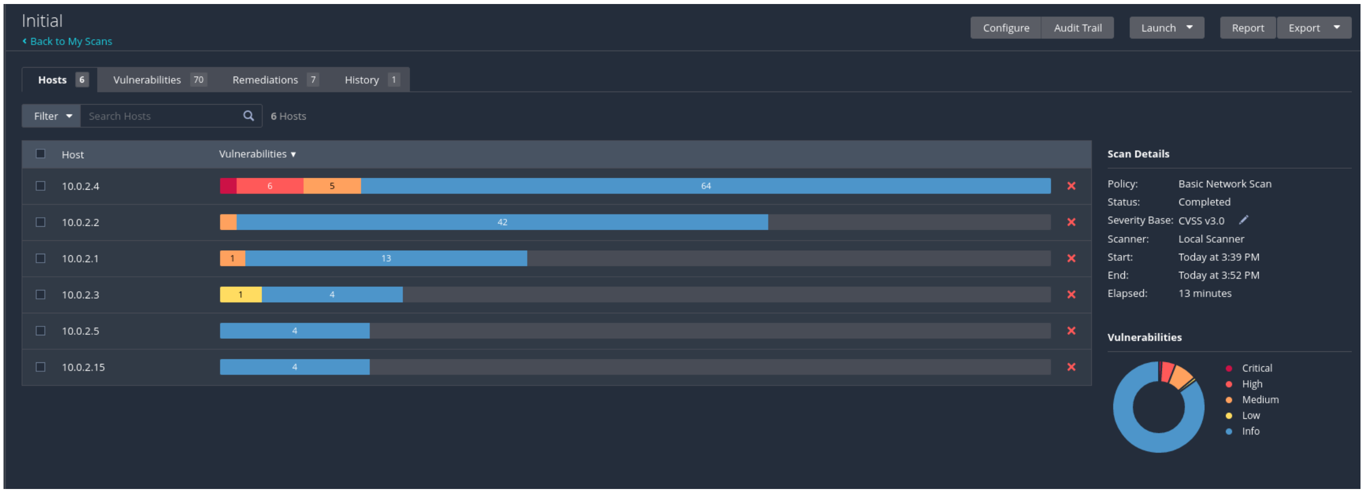
Name: Initial

Description:

Folder: My Scans

Targets: 10.0.2.0/24

Here are my configurations for the initial network scan



Host	Vulnerabilities ▾
10.0.2.4	6 5 64
10.0.2.2	42
10.0.2.1	1 13
10.0.2.3	1 4
10.0.2.5	4
10.0.2.15	4

Scan Details

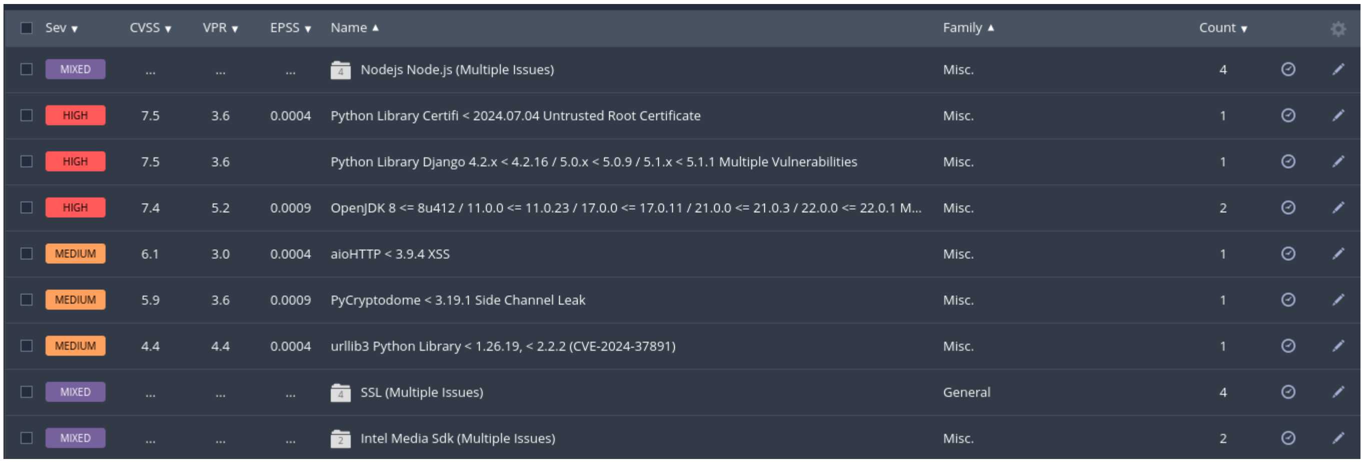
Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 3:39 PM
End:	Today at 3:52 PM
Elapsed:	13 minutes

Vulnerabilities

Here is the result of my scan

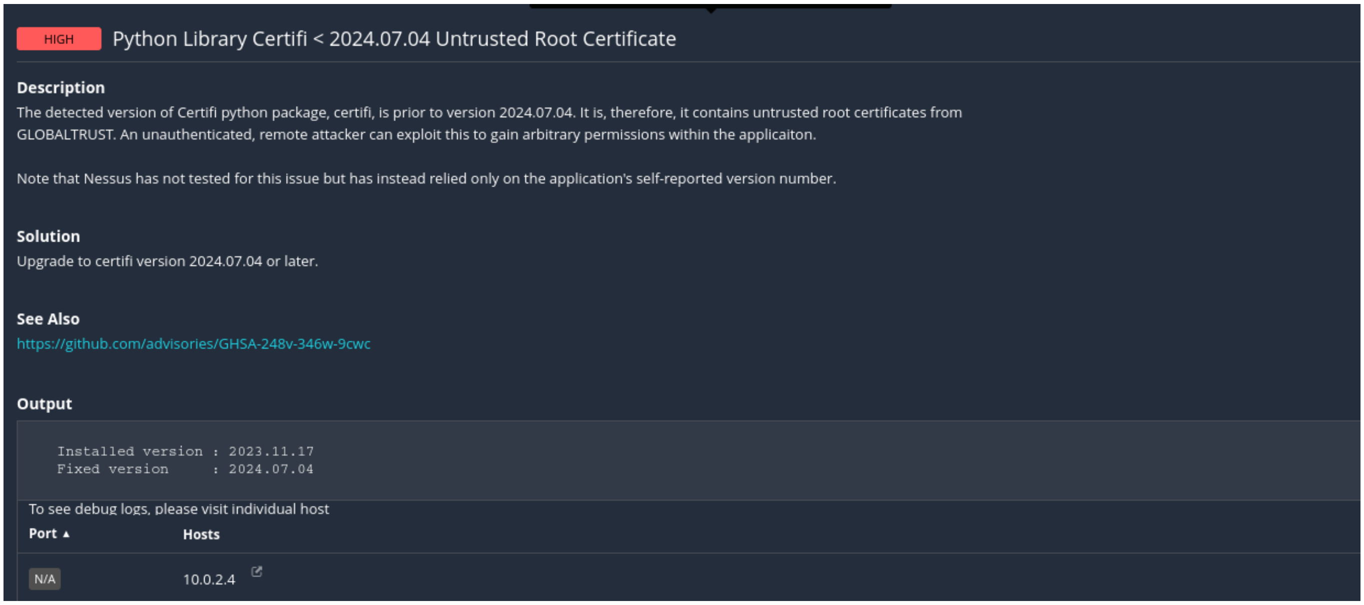
2.4 is my kali machine (makes sense why its so vulnerable)

2.15 is my ubuntu machine and 2.5 is my windows machine



□	Sev ▾	CVSS ▾	VPR ▾	EPSS ▾	Name ▴	Family ▴	Count ▾
□	MIXED	...	...	...	Nodejs Node.js (Multiple Issues)	Misc.	4
□	HIGH	7.5	3.6	0.0004	Python Library Certifi < 2024.07.04 Untrusted Root Certificate	Misc.	1
□	HIGH	7.5	3.6		Python Library Django 4.2.x < 4.2.16 / 5.0.x < 5.0.9 / 5.1.x < 5.1.1 Multiple Vulnerabilities	Misc.	1
□	HIGH	7.4	5.2	0.0009	OpenJDK 8 <= 8u412 / 11.0.0 <= 11.0.23 / 17.0.0 <= 17.0.11 / 21.0.0 <= 21.0.3 / 22.0.0 <= 22.0.1 M...	Misc.	2
□	MEDIUM	6.1	3.0	0.0004	aioHTTP < 3.9.4 XSS	Misc.	1
□	MEDIUM	5.9	3.6	0.0009	PyCryptodome < 3.19.1 Side Channel Leak	Misc.	1
□	MEDIUM	4.4	4.4	0.0004	urllib3 Python Library < 1.26.19, < 2.2.2 (CVE-2024-37891)	Misc.	1
□	MIXED	...	...	...	SSL (Multiple Issues)	General	4
□	MIXED	...	...	...	Intel Media Sdk (Multiple Issues)	Misc.	2

Here are the vulnerabilites of my kali machine. Typically kali machines are very vulnerable since they are mainly used for attacking and not so much for actually housing any sensitive data



HIGH Python Library Certifi < 2024.07.04 Untrusted Root Certificate

Description

The detected version of Certifi python package, certifi, is prior to version 2024.07.04. It is, therefore, it contains untrusted root certificates from GLOBALTRUST. An unauthenticated, remote attacker can exploit this to gain arbitrary permissions within the applicaiton.

Note that Nessus has not tested for this issue but has instead relied only on the application's self-reported version number.

Solution

Upgrade to certifi version 2024.07.04 or later.

See Also

https://github.com/advisories/GHSA-248v-346w-9cwc

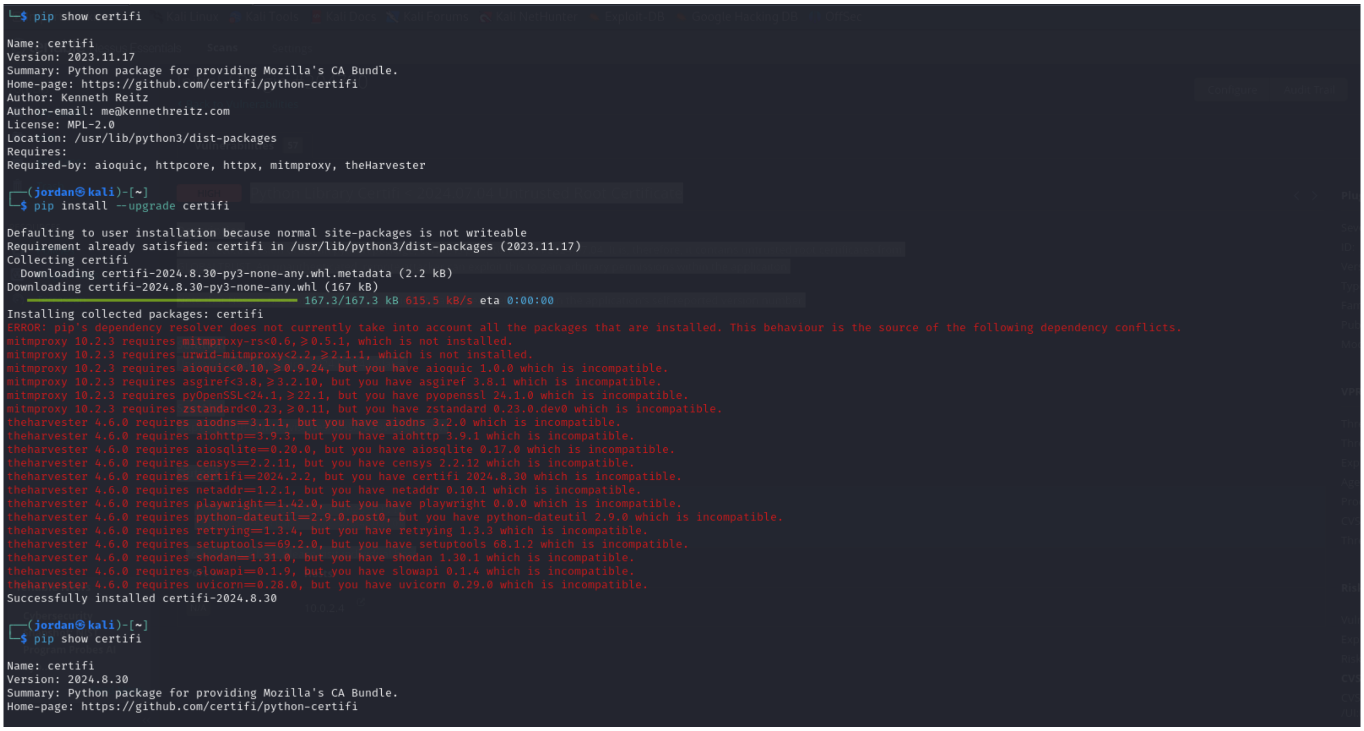
Output

```
  Installed version : 2023.11.17
  Fixed version     : 2024.07.04
```

To see debug logs, please visit individual host

Port ▴	Hosts
N/A	10.0.2.4

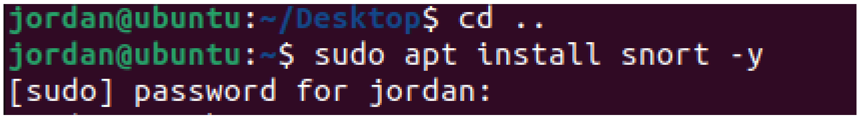
One of the vulnerabilities I'll look at is python library certificate untrusted root certificate. this is a big vulnerability since a remote attacker can exploit this to gain permissions within the app. This is due to the package being an old version making it contain an untrusted root certificate from GLOBALTRUST.



```
└─$ pip show certifi

Name: certifi
Version: 2023.11.17
Summary: Python package for providing Mozilla's CA Bundle.
Home-page: https://github.com/certifi/python-certifi
Author: Kenneth Reitz
Author-email: me@kennethreitz.com
License: MPL-2.0
Location: /usr/lib/python3/dist-packages
Requires:
Required-by: aioquic, httpcore, httpx, mitmproxy, theHarvester

┌──(jordan㉿kali)-[~]
└─$ pip install --upgrade certifi

Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: certifi in /usr/lib/python3/dist-packages (2023.11.17)
Collecting certifi
  Downloading certifi-2024.8.30-py3-none-any.whl.metadata (2.2 kB)
Downloading certifi-2024.8.30-py3-none-any.whl (167 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 167.3/167.3 kB 615.5 kB/s eta 0:00:00
Installing collected packages: certifi
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
mitmproxy 10.2.3 requires mitmproxy-rs<0.6,≥0.5.1, which is not installed.
mitmproxy 10.2.3 requires urwid-mitmproxy<2.2,≥2.1.1, which is not installed.
mitmproxy 10.2.3 requires aioquic<0.10,≥0.9.24, but you have aioquic 1.0.0 which is incompatible.
mitmproxy 10.2.3 requires asgiref<3.8,≥3.2.10, but you have asgiref 3.8.1 which is incompatible.
mitmproxy 10.2.3 requires pyOpenSSL<24.1,≥22.1, but you have pyopenssl 24.1.0 which is incompatible.
mitmproxy 10.2.3 requires zstandard<0.23,≥0.11, but you have zstandard 0.23.0.dev0 which is incompatible.
theharvester 4.6.0 requires aiodns==3.1.1, but you have aiodns 3.2.0 which is incompatible.
theharvester 4.6.0 requires aiohttp==3.9.3, but you have aiohttp 3.9.1 which is incompatible.
theharvester 4.6.0 requires aiosqlite==0.20.0, but you have aiosqlite 0.17.0 which is incompatible.
theharvester 4.6.0 requires censys==2.2.11, but you have censys 2.2.12 which is incompatible.
theharvester 4.6.0 requires certifi==2024.2.2, but you have certifi 2024.8.30 which is incompatible.
theharvester 4.6.0 requires netaddr==1.2.1, but you have netaddr 0.10.1 which is incompatible.
theharvester 4.6.0 requires playwright==1.42.0, but you have playwright 0.0.0 which is incompatible.
theharvester 4.6.0 requires python-dateutil==2.9.0.post0, but you have python-dateutil 2.9.0 which is incompatible.
theharvester 4.6.0 requires retrying==1.3.4, but you have retrying 1.3.3 which is incompatible.
theharvester 4.6.0 requires setuptools==69.2.0, but you have setuptools 68.1.2 which is incompatible.
theharvester 4.6.0 requires shodan==1.31.0, but you have shodan 1.30.1 which is incompatible.
theharvester 4.6.0 requires slowapi==0.1.9, but you have slowapi 0.1.4 which is incompatible.
theharvester 4.6.0 requires uvicorn==0.28.0, but you have uvicorn 0.29.0 which is incompatible.
Successfully installed certifi-2024.8.30

┌──(jordan㉿kali)-[~]
└─$ pip show certifi

Name: certifi
Version: 2024.8.30
Summary: Python package for providing Mozilla's CA Bundle.
Home-page: https://github.com/certifi/python-certifi
```

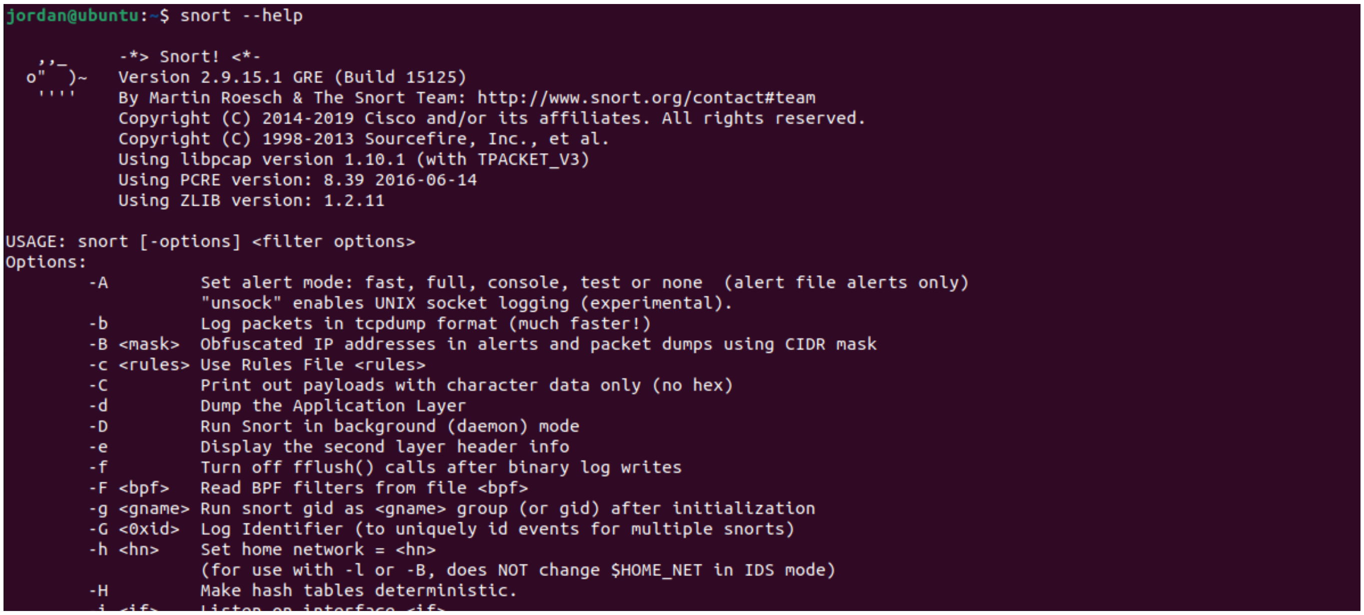
Here is the fix to the vulnerability, my pip show certifi command shows I updated the package and this vulnerability should now be patched

7.3 - Snort Detection



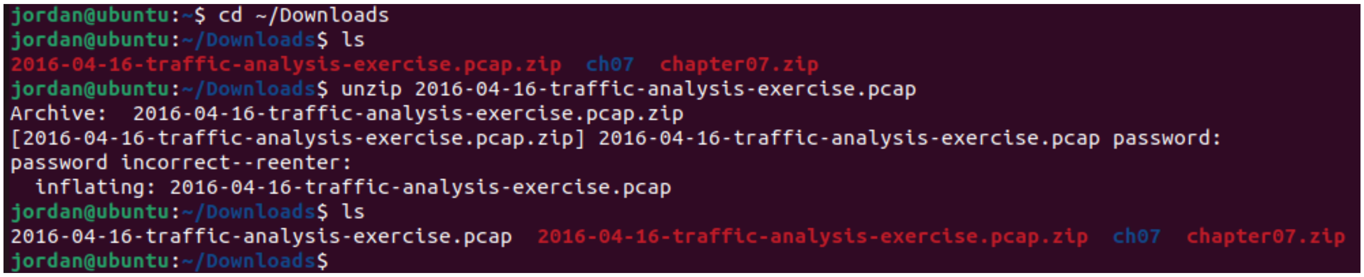
```
jordan@ubuntu:~/Desktop$ cd ..
jordan@ubuntu:~$ sudo apt install snort -y
[sudo] password for jordan:
```

I am now installing snort



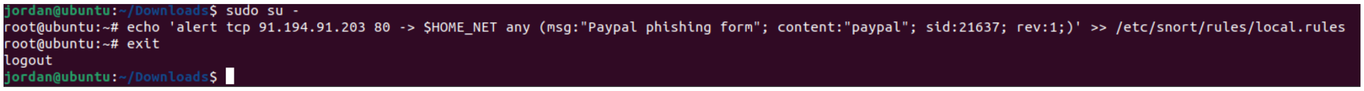
```
jordan@ubuntu:~$ snort --help

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

USAGE: snort [-options] <filter options>
Options:
        -A         Set alert mode: fast, full, console, test or none  (alert file alerts only)
                   "unsock" enables UNIX socket logging (experimental).
        -b         Log packets in tcpdump format (much faster!)
        -B <mask>  Obfuscated IP addresses in alerts and packet dumps using CIDR mask
        -c <rules> Use Rules File <rules>
        -C         Print out payloads with character data only (no hex)
        -d         Dump the Application Layer
        -D         Run Snort in background (daemon) mode
        -e         Display the second layer header info
        -f         Turn off fflush() calls after binary log writes
        -F <bpf>   Read BPF filters from file <bpf>
        -g <gname> Run snort gid as <gname> group (or gid) after initialization
        -G <0xid>  Log Identifier (to uniquely id events for multiple snorts)
        -h <hn>    Set home network = <hn>
                   (for use with -l or -B, does NOT change $HOME_NET in IDS mode)
        -H         Make hash tables deterministic.
        -i <if>    Listen on interface <if>
```

Snort is now installed



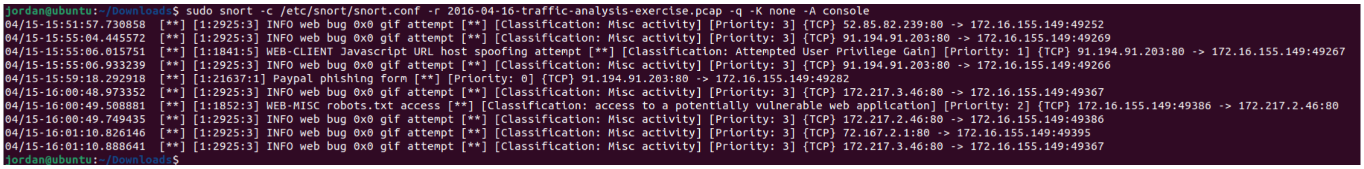
```
jordan@ubuntu:~$ cd ~/Downloads
jordan@ubuntu:~/Downloads$ ls
2016-04-16-traffic-analysis-exercise.pcap.zip  ch07  chapter07.zip
jordan@ubuntu:~/Downloads$ unzip 2016-04-16-traffic-analysis-exercise.pcap
Archive:  2016-04-16-traffic-analysis-exercise.pcap.zip
[2016-04-16-traffic-analysis-exercise.pcap.zip] 2016-04-16-traffic-analysis-exercise.pcap password:
password incorrect--reenter:
  inflating: 2016-04-16-traffic-analysis-exercise.pcap
jordan@ubuntu:~/Downloads$ ls
2016-04-16-traffic-analysis-exercise.pcap  2016-04-16-traffic-analysis-exercise.pcap.zip  ch07  chapter07.zip
jordan@ubuntu:~/Downloads$
```

I installed the pcap file and unzipped it into my downloads folder



```
jordan@ubuntu:~/Downloads$ sudo su -
root@ubuntu:~# echo 'alert tcp 91.194.91.203 80 -> $HOME_NET any (msg:"Paypal phishing form"; content:"paypal"; sid:21637; rev:1;)' >> /etc/snort/rules/local.rules
root@ubuntu:~# exit
logout
jordan@ubuntu:~/Downloads$ 
```

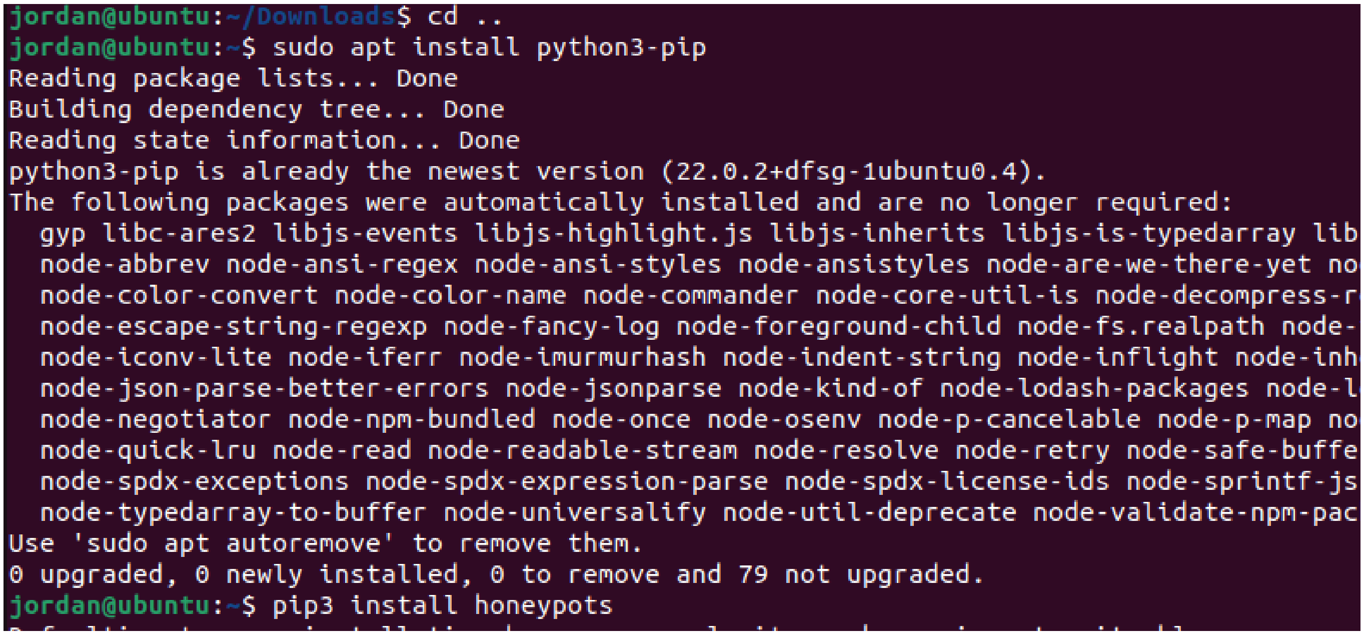
I created a custom rule to detect if a known malicious webserver was accessed and credential form submitted



```
jordan@ubuntu:~/Downloads$ sudo snort -c /etc/snort/snort.conf -r 2016-04-16-traffic-analysis-exercise.pcap -q -K none -A console
04/15-15:51:57.730858  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 52.85.82.239:80 -> 172.16.155.149:49252
04/15-15:55:04.445572  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 91.194.91.203:80 -> 172.16.155.149:49269
04/15-15:55:06.015751  [**] [1:1841:5] WEB-CLIENT Javascript URL host spoofing attempt [**] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 91.194.91.203:80 -> 172.16.155.149:49267
04/15-15:55:06.933239  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 91.194.91.203:80 -> 172.16.155.149:49266
04/15-15:59:18.292918  [**] [1:21637:1] Paypal phishing form [**] [Priority: 0] {TCP} 91.194.91.203:80 -> 172.16.155.149:49282
04/15-16:00:48.973352  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 172.217.3.46:80 -> 172.16.155.149:49367
04/15-16:00:49.508881  [**] [1:1852:3] WEB-MISC robots.txt access [**] [Classification: access to a potentially vulnerable web application] [Priority: 2] {TCP} 172.16.155.149:49386 -> 172.217.2.46:80
04/15-16:00:49.749435  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 172.217.2.46:80 -> 172.16.155.149:49386
04/15-16:01:10.826146  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 72.167.2.1:80 -> 172.16.155.149:49395
04/15-16:01:10.888641  [**] [1:2925:3] INFO web bug 0x0 gif attempt [**] [Classification: Misc activity] [Priority: 3] {TCP} 172.217.3.46:80 -> 172.16.155.149:49367
jordan@ubuntu:~/Downloads$
```

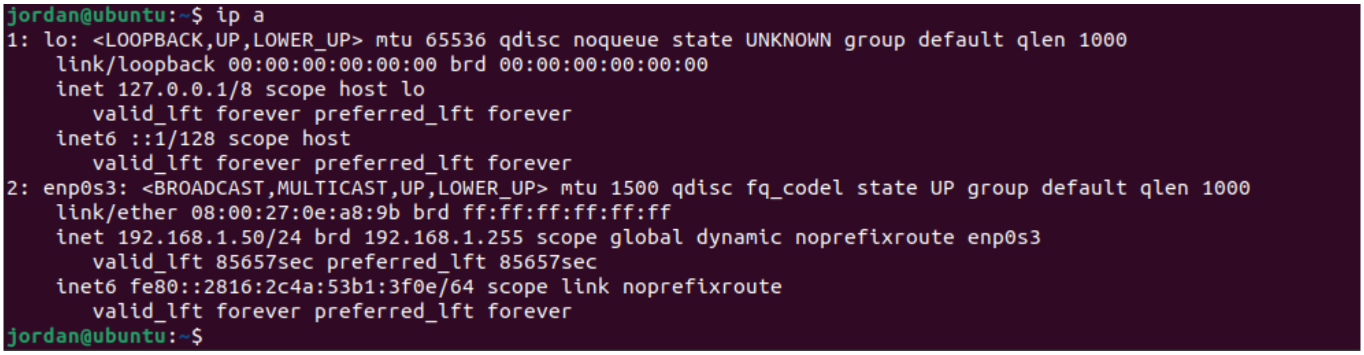
By running snort with the pcap file we triggered the paypal rule!

7.4 - MySQL Honeypot



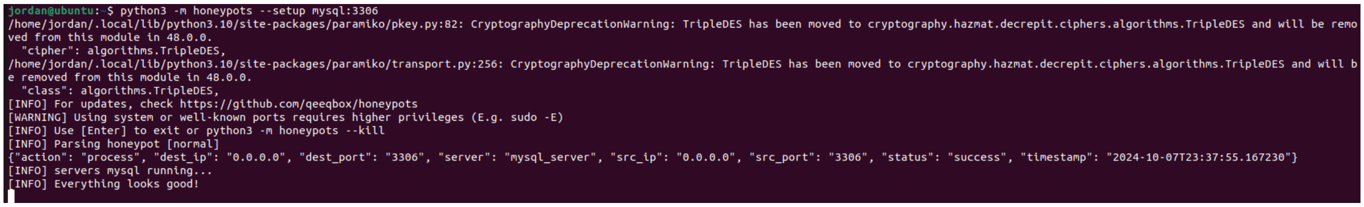
```
jordan@ubuntu:~/Downloads$ cd ..
jordan@ubuntu:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2+dfsg-1ubuntu0.4).
The following packages were automatically installed and are no longer required:
  gyp libc-ares2 libjs-events libjs-highlight.js libjs-inherits libjs-is-typedarray lib
  node-abbrev node-ansi-regex node-ansi-styles node-ansistyles node-are-we-there-yet no
  node-color-convert node-color-name node-commander node-core-util-is node-decompress-r
  node-escape-string-regexp node-fancy-log node-foreground-child node-fs.realpath node-
  node-iconv-lite node-iferr node-imurmurhash node-indent-string node-inflight node-inh
  node-json-parse-better-errors node-jsonparse node-kind-of node-lodash-packages node-l
  node-negotiator node-npm-bundled node-once node-osenv node-p-cancelable node-p-map no
  node-quick-lru node-read node-readable-stream node-resolve node-retry node-safe-buffe
  node-spdx-exceptions node-spdx-expression-parse node-spdx-license-ids node-sprintf-js
  node-typedarray-to-buffer node-universalify node-util-deprecate node-validate-npm-pac
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 79 not upgraded.
jordan@ubuntu:~$ pip3 install honeypots
```

First I installed pip package and the honeypots package from pip



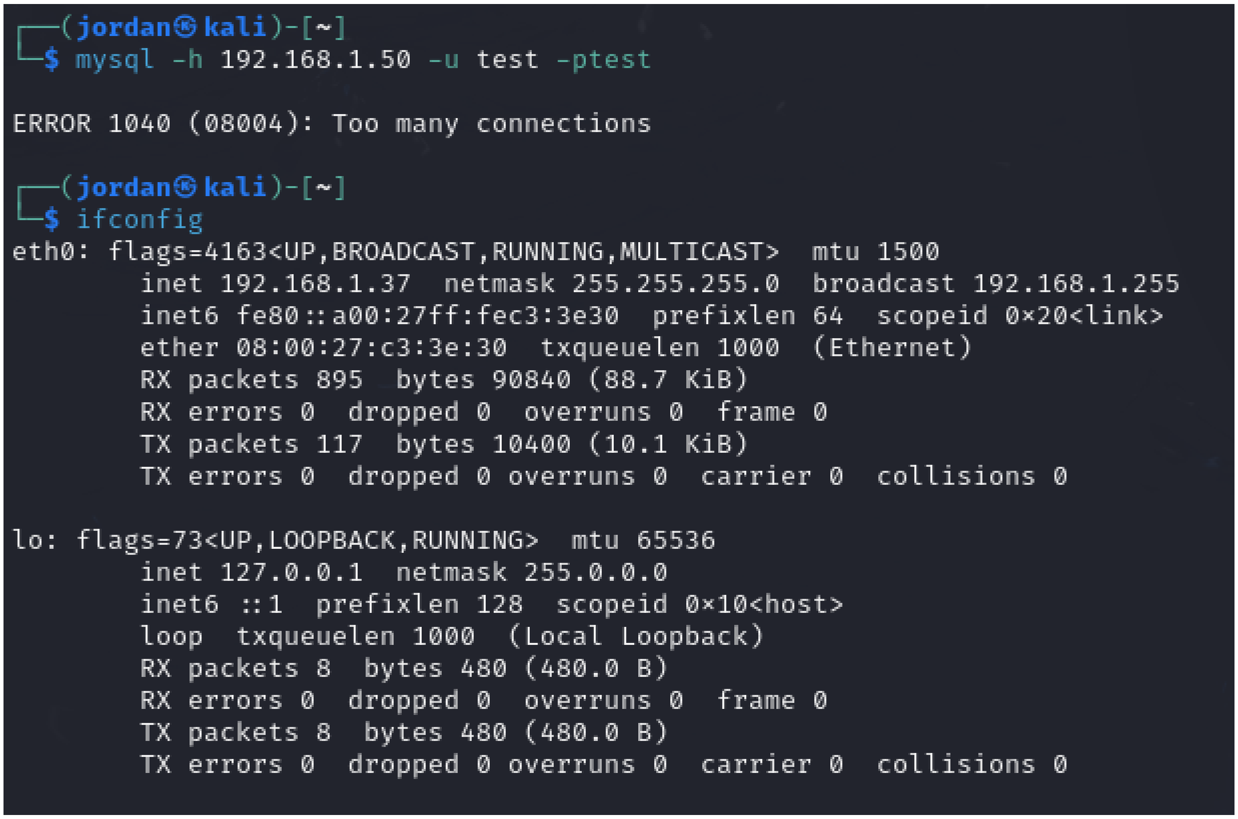
```
jordan@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0e:a8:9b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
       valid_lft 85657sec preferred_lft 85657sec
    inet6 fe80::2816:2c4a:53b1:3f0e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
jordan@ubuntu:~$
```

My IP is 192.168.1.50/24



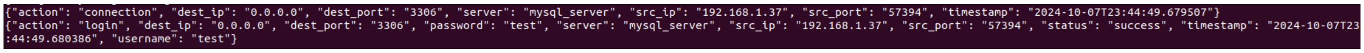
```
jordan@ubuntu:~$ python3 -m honeypots --setup mysql:3306
/home/jordan/.local/lib/python3.10/site-packages/paramiko/pkey.py:82: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be remo
ved from this module in 48.0.0.
  "cipher": algorithms.TripleDES,
/home/jordan/.local/lib/python3.10/site-packages/paramiko/transport.py:256: CryptographyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will b
e removed from this module in 48.0.0.
  "class": algorithms.TripleDES,
[INFO] For updates, check https://github.com/qeeqbox/honeypots
[WARNING] Using system or well-known ports requires higher privileges (E.g. sudo -E)
[INFO] Use [Enter] to exit or python3 -m honeypots --kill
[INFO] Parsing honeypot [normal]
{"action": "process", "dest_ip": "0.0.0.0", "dest_port": "3306", "server": "mysql_server", "src_ip": "0.0.0.0", "src_port": "3306", "status": "success", "timestamp": "2024-10-07T23:37:55.167230"}
[INFO] servers mysql running...
[INFO] Everything looks good!
```

I have setup the mysql honeypot on my ubuntu machine



```
┌──(jordan㉿kali)-[~]
└─$ mysql -h 192.168.1.50 -u test -ptest

ERROR 1040 (08004): Too many connections

┌──(jordan㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.37  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:fec3:3e30  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:c3:3e:30  txqueuelen 1000  (Ethernet)
        RX packets 895  bytes 90840 (88.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 117  bytes 10400 (10.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

I got an error when running the test and verified my ip on kali as 192.168.1.37



```
{"action": "connection", "dest_ip": "0.0.0.0", "dest_port": "3306", "server": "mysql_server", "src_ip": "192.168.1.37", "src_port": "57394", "timestamp": "2024-10-07T23:44:49.679507"}
{"action": "login", "dest_ip": "0.0.0.0", "dest_port": "3306", "password": "test", "server": "mysql_server", "src_ip": "192.168.1.37", "src_port": "57394", "status": "success", "timestamp": "2024-10-07T23
:44:49.680386", "username": "test"}
```

Going back to ubuntu we see the attack was logged and ubuntu caught the source of the connection