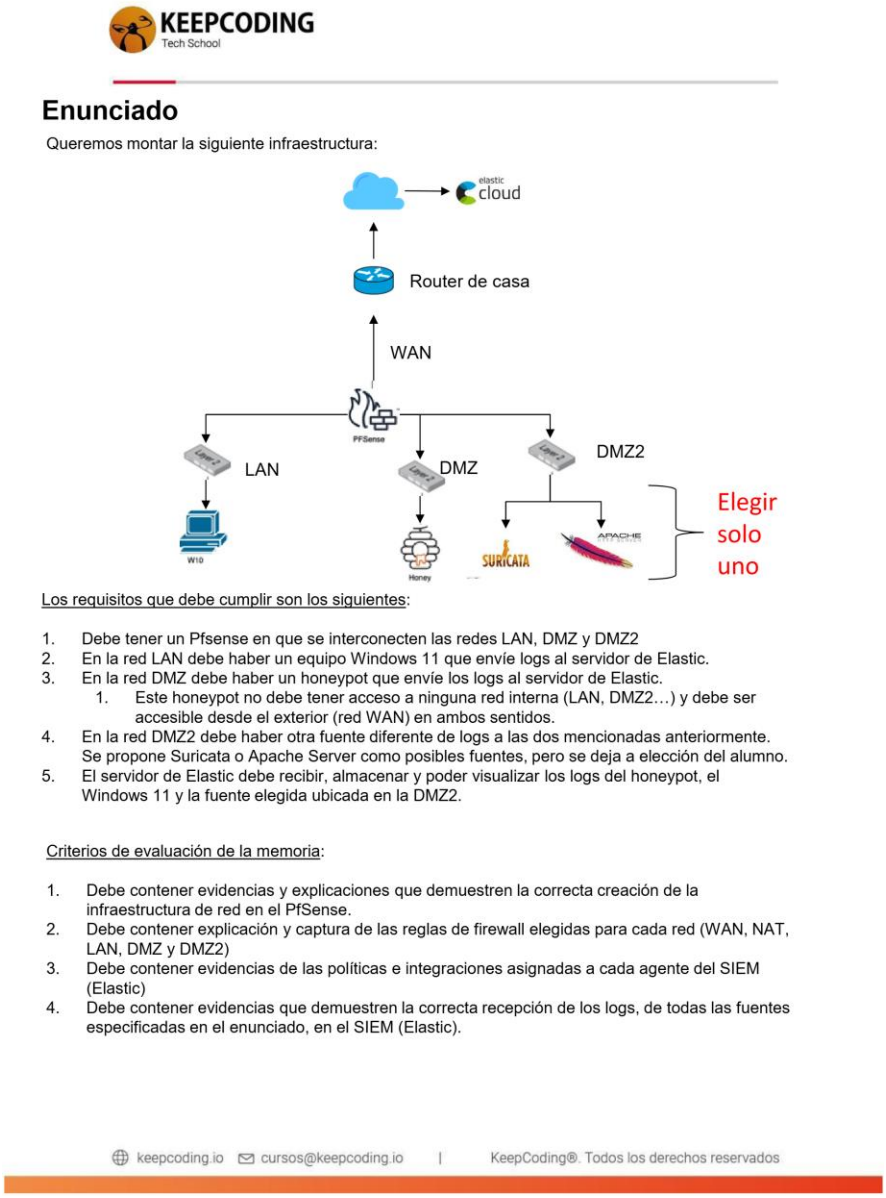
Alain Gonzalez

Cybersecurity Blue Team Report



KEEPCODING
Tech School

Enunciado

Queremos montar la siguiente infraestructura:

Los requisitos que debe cumplir son los siguientes:

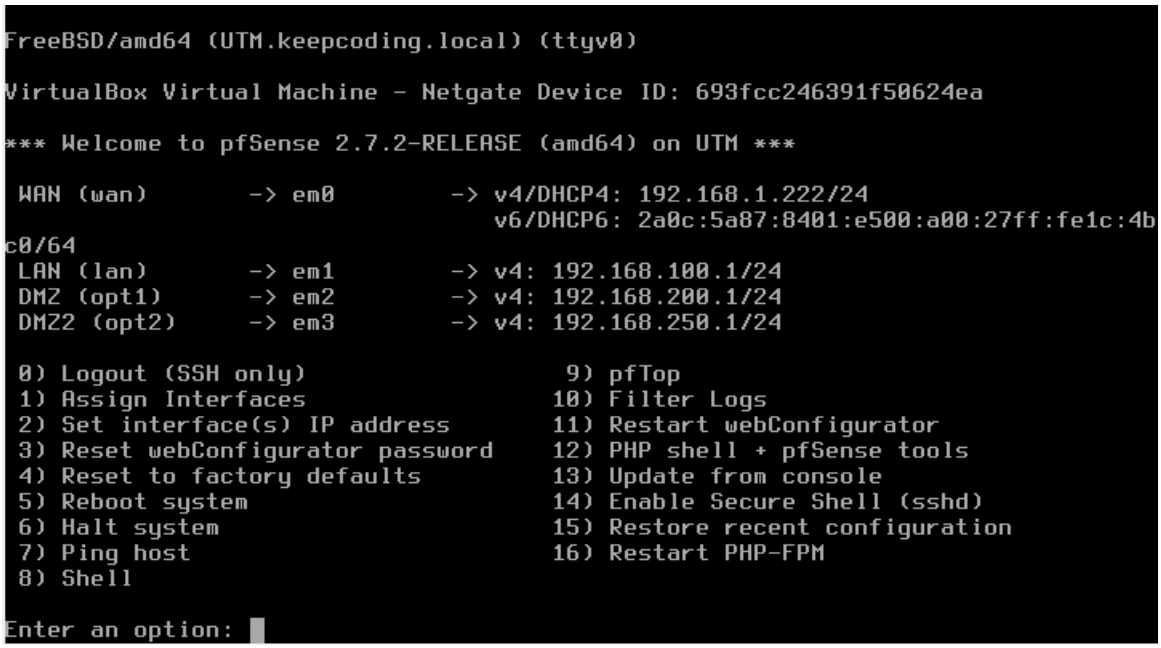
1. Debe tener un Pfsense en que se interconecten las redes LAN, DMZ y DMZ2
2. En la red LAN debe haber un equipo Windows 11 que envíe logs al servidor de Elastic.
3. En la red DMZ debe haber un honeypot que envíe los logs al servidor de Elastic.
    1. Este honeypot no debe tener acceso a ninguna red interna (LAN, DMZ2…) y debe ser accesible desde el exterior (red WAN) en ambos sentidos.
4. En la red DMZ2 debe haber otra fuente diferente de logs a las dos mencionadas anteriormente. Se propone Suricata o Apache Server como posibles fuentes, pero se deja a elección del alumno.
5. El servidor de Elastic debe recibir, almacenar y poder visualizar los logs del honeypot, el Windows 11 y la fuente elegida ubicada en la DMZ2.

Criterios de evaluación de la memoria:

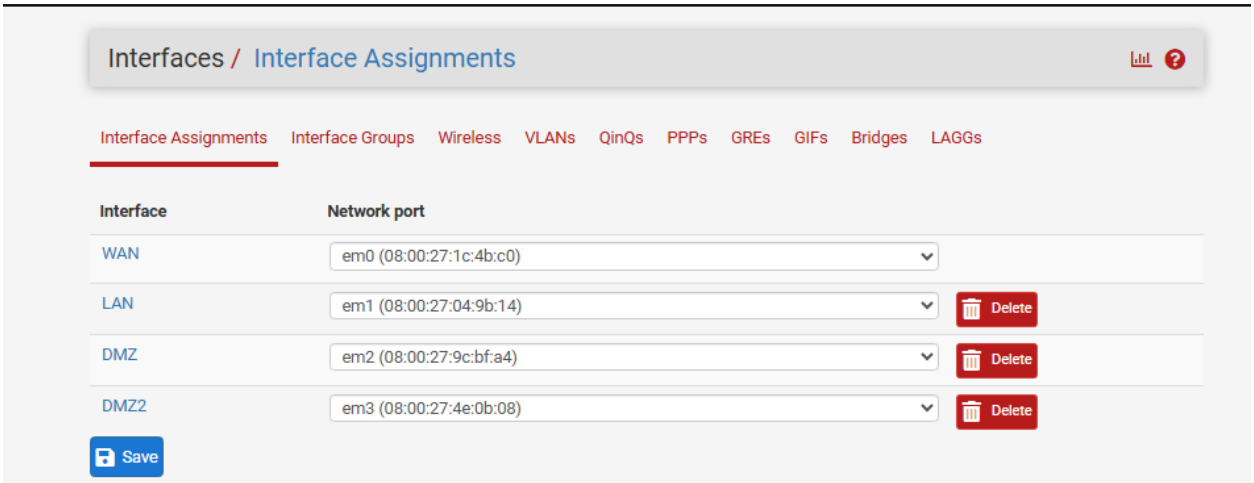
1. Debe contener evidencias y explicaciones que demuestren la correcta creación de la infraestructura de red en el PfSense.
2. Debe contener explicación y captura de las reglas de firewall elegidas para cada red (WAN, NAT, LAN, DMZ y DMZ2)
3. Debe contener evidencias de las políticas e integraciones asignadas a cada agente del SIEM (Elastic)
4. Debe contener evidencias que demuestren la correcta recepción de los logs, de todas las fuentes especificadas en el enunciado, en el SIEM (Elastic).

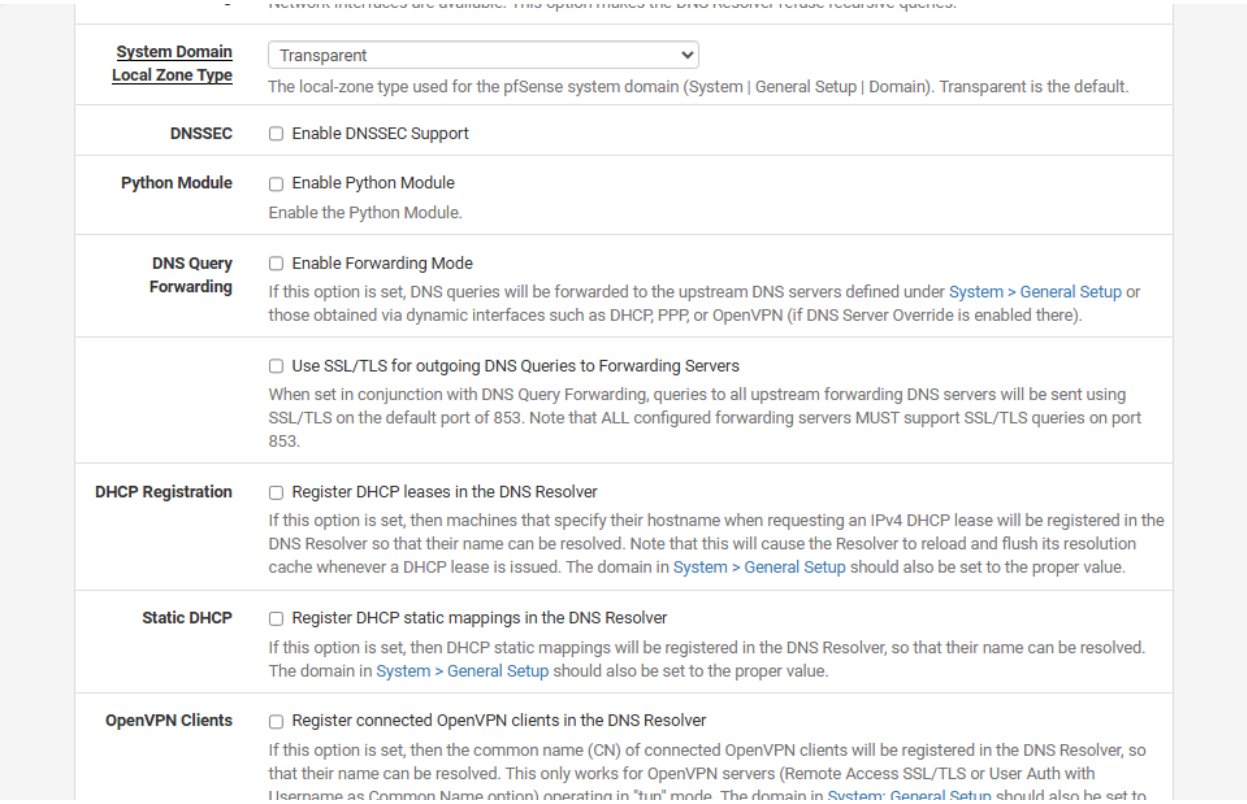
First off, I'm going to demonstrate my pfSense configurations and rules for all internal networks (LAN, DMZ, DMZ2).

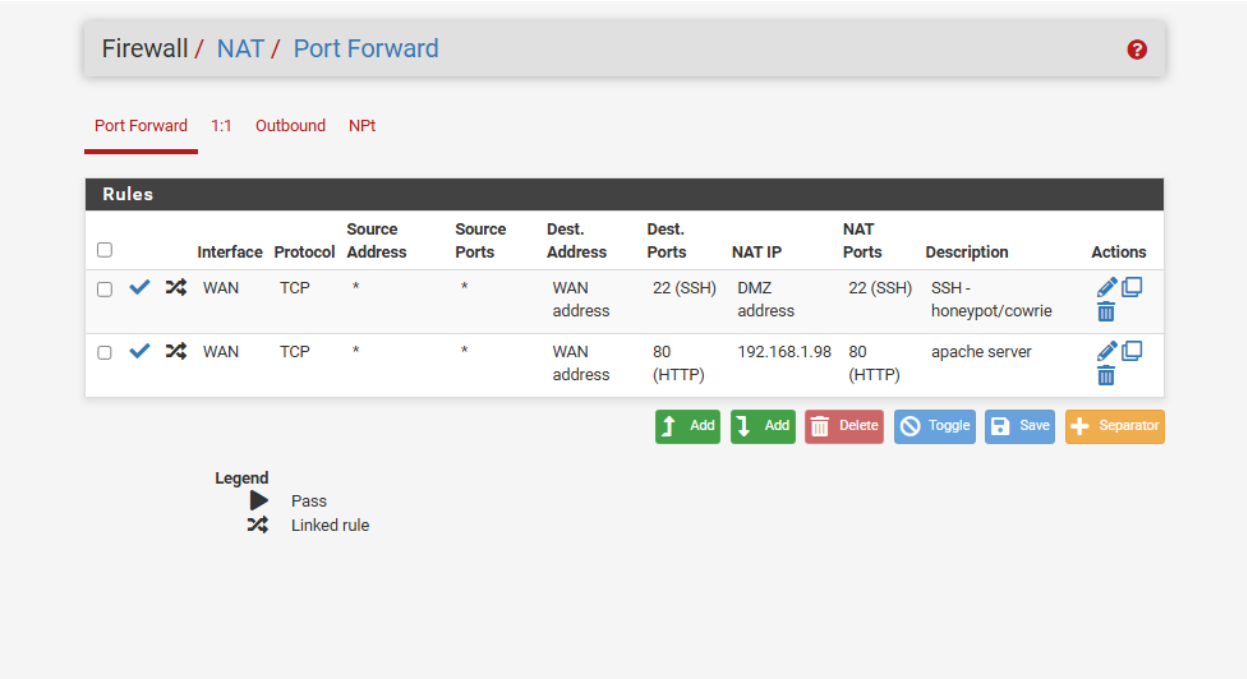
1. pfSense



Here’s the pfSense machine displaying all the internal networks and their assigned IPs. Below this image are the individual configurations for the internal networks, as well as their rules (including port forwarding). The settings “Enable DNSSEC Support” and “Enable Python Module” must be disabled.

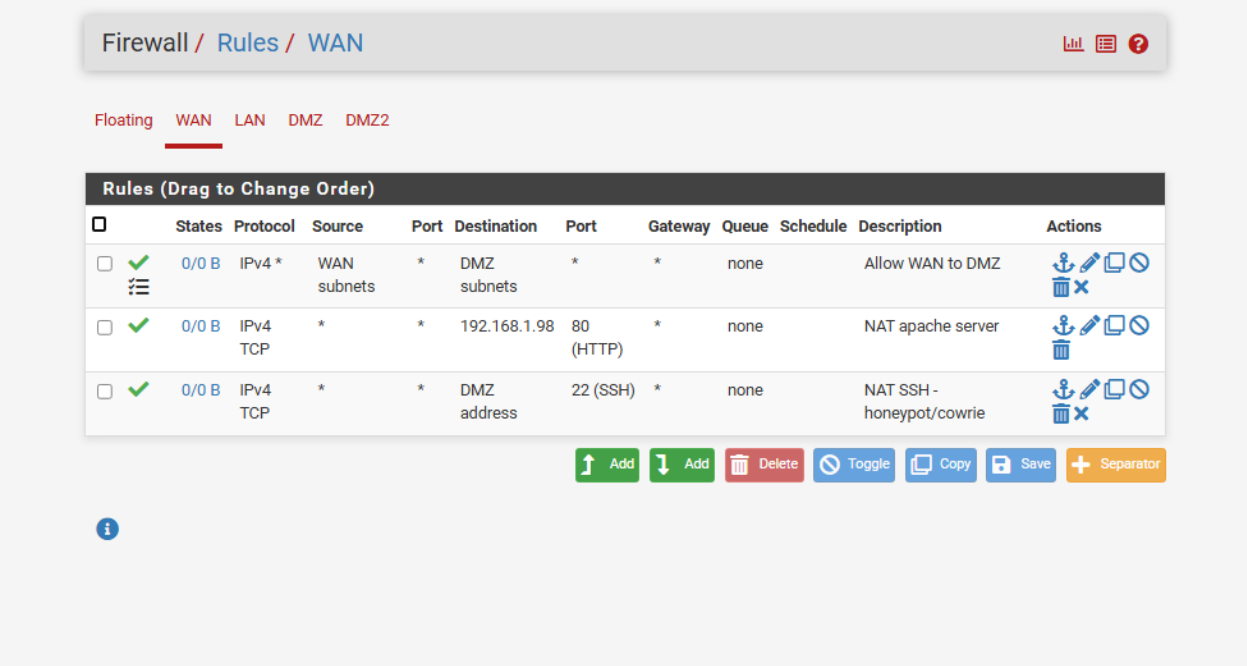


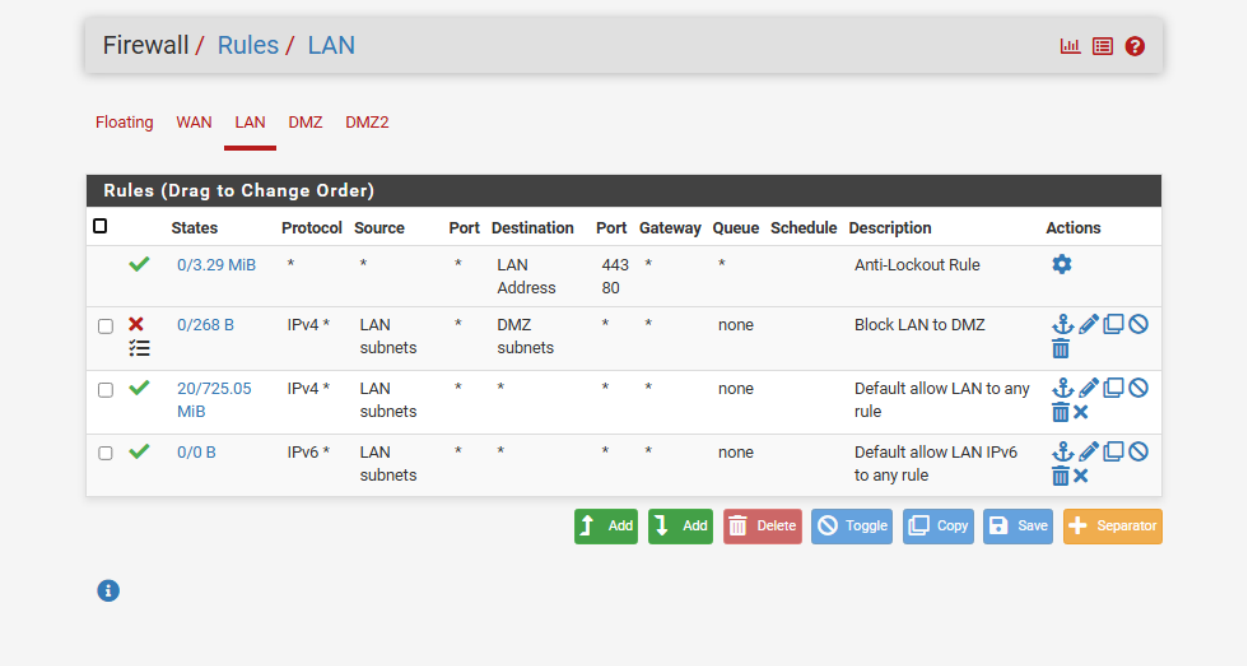


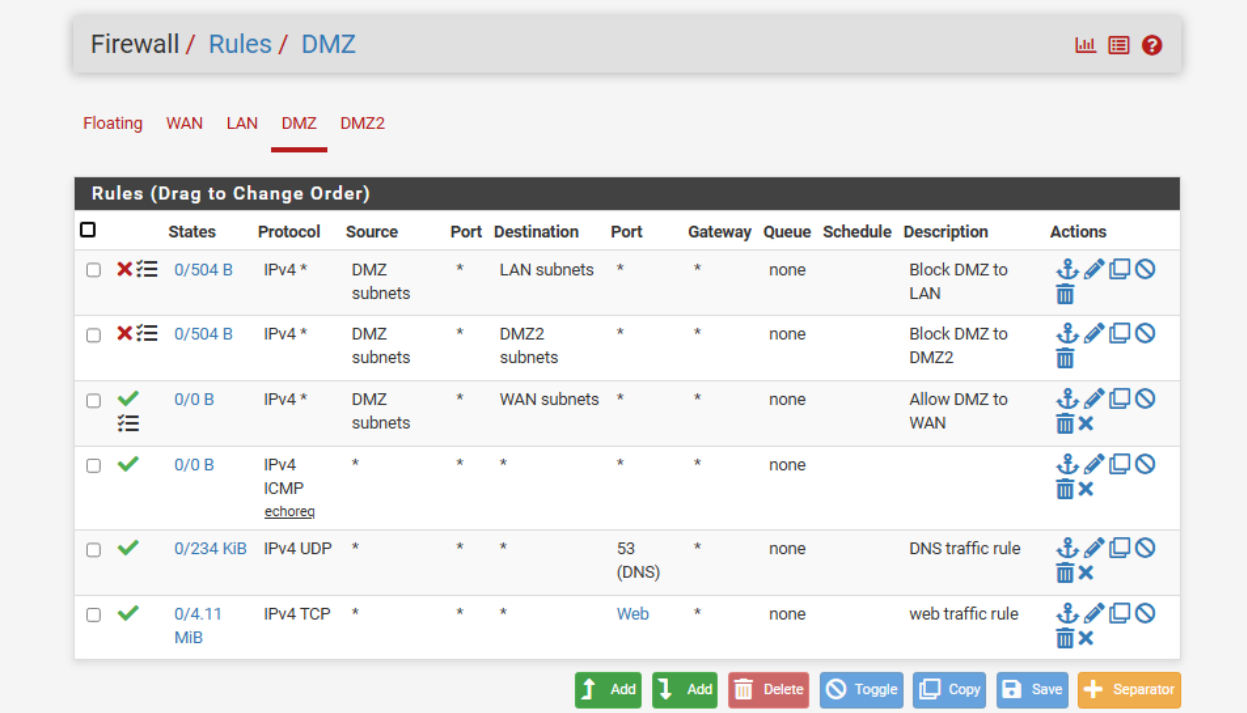


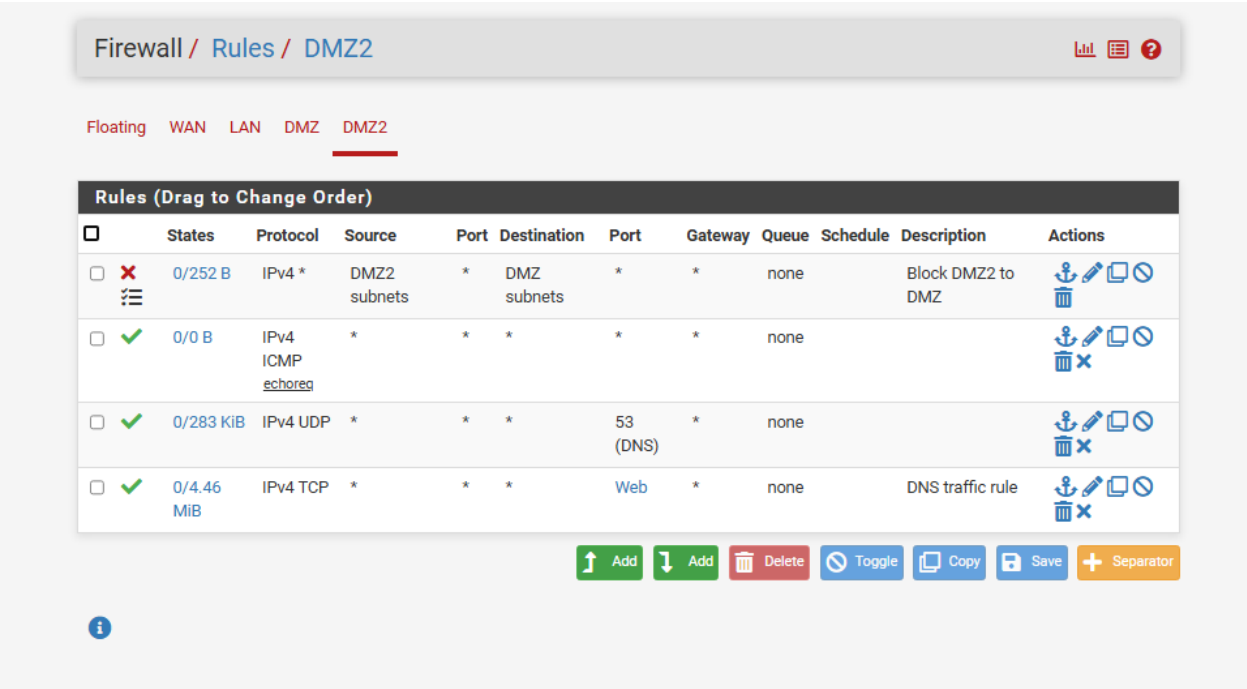
The port ”22” must be open for cowrie to be able to connect via SSH to the outside.

DMZ (Honeypot) must not be able to access LAN or DMZ2, while still having access to WAN both ways. The following screenshots are the configurations for each interface's firewall rules to make sure this works.



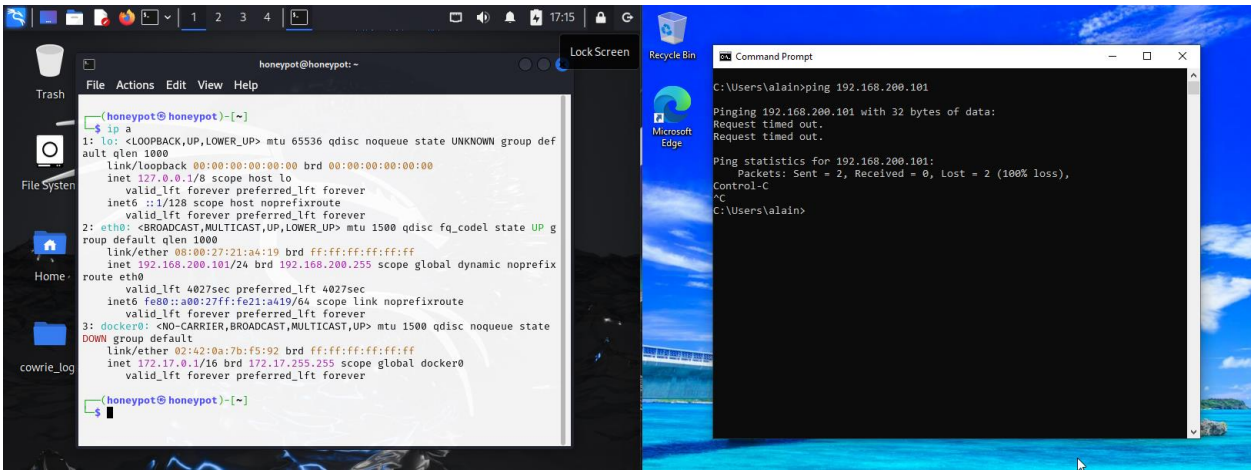


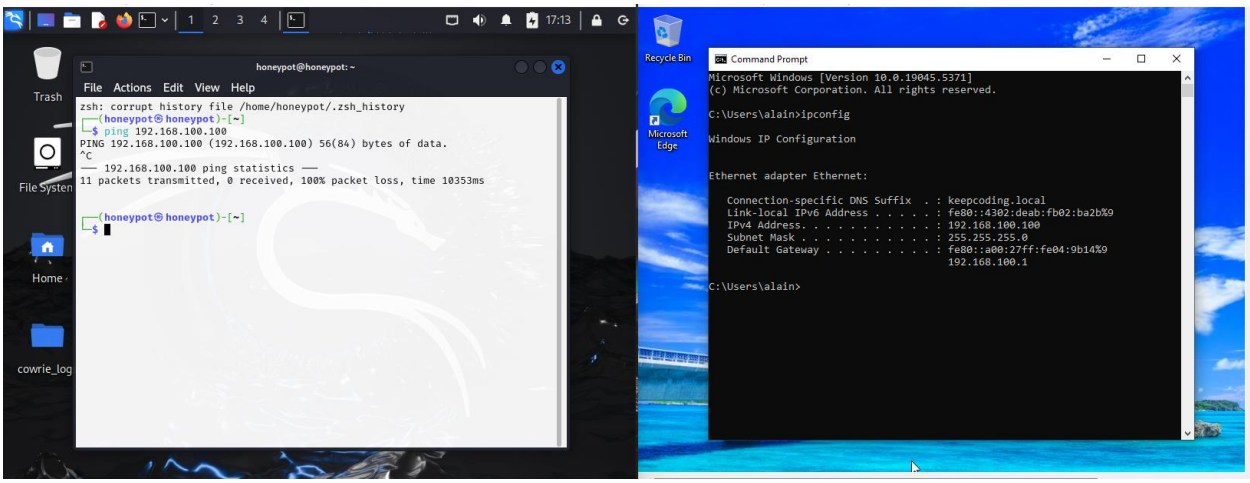




These are the firewall rules for each interface. DMZ and DMZ2 will have rules that block each other using their subnets. Same thing goes for DMZ and LAN. DMZ and WAN both share settings that allow communication to each other back and forth.

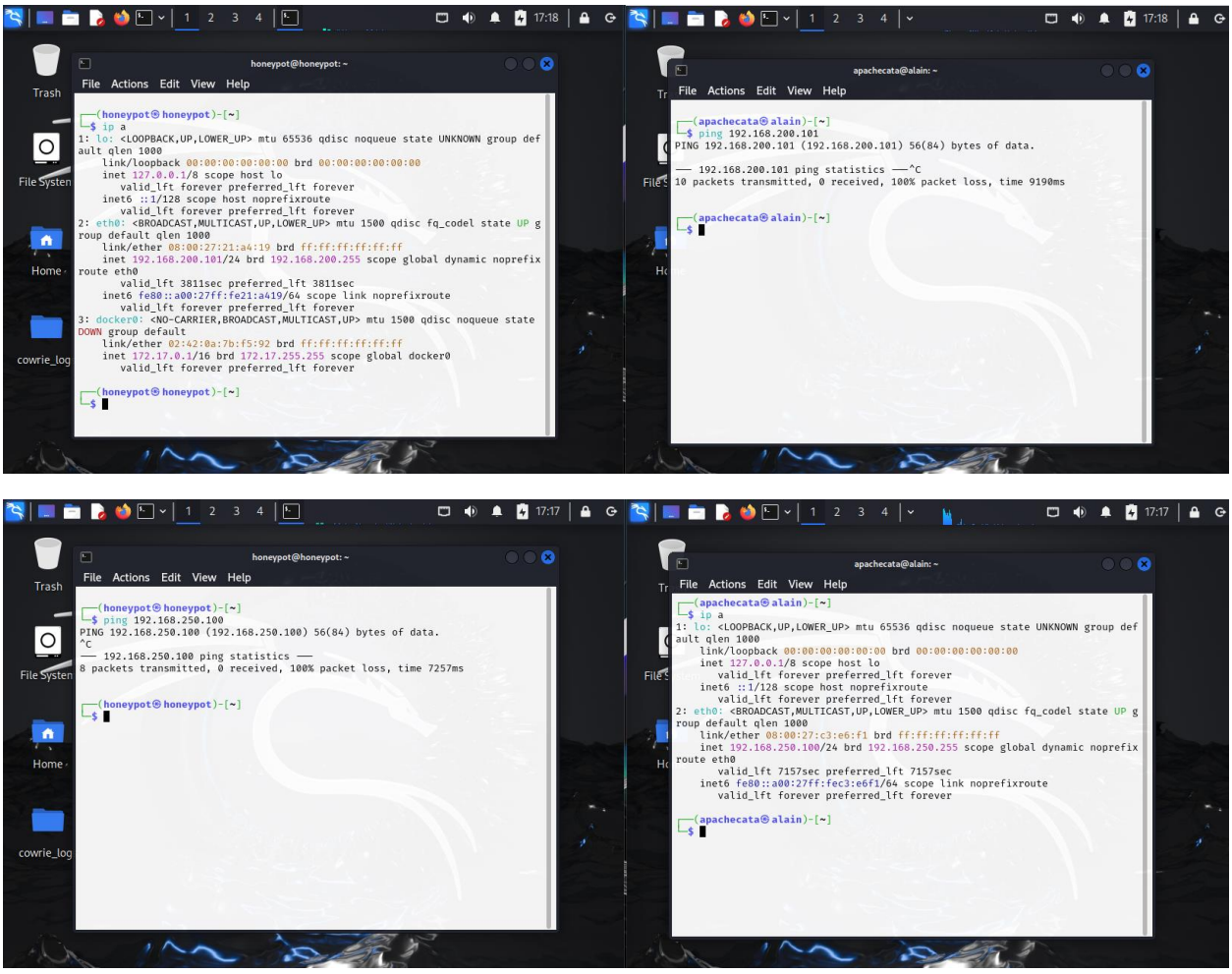
Here are screenshots displaying the rules in practice, showing that their communication between each other is indeed blocked by the firewall rules put in place. I used ping commands to present this example.





These two screenshots above are showing DMZ's inability to communicate with LAN (the Windows machine), and vice versa. When DMZ (Honeypot) attempts to ping 192.168.100.100 (the LAN machine's assigned IP), the

connection times out. The same thing occurs when the LAN machine attempts to ping the DMZ machine back.

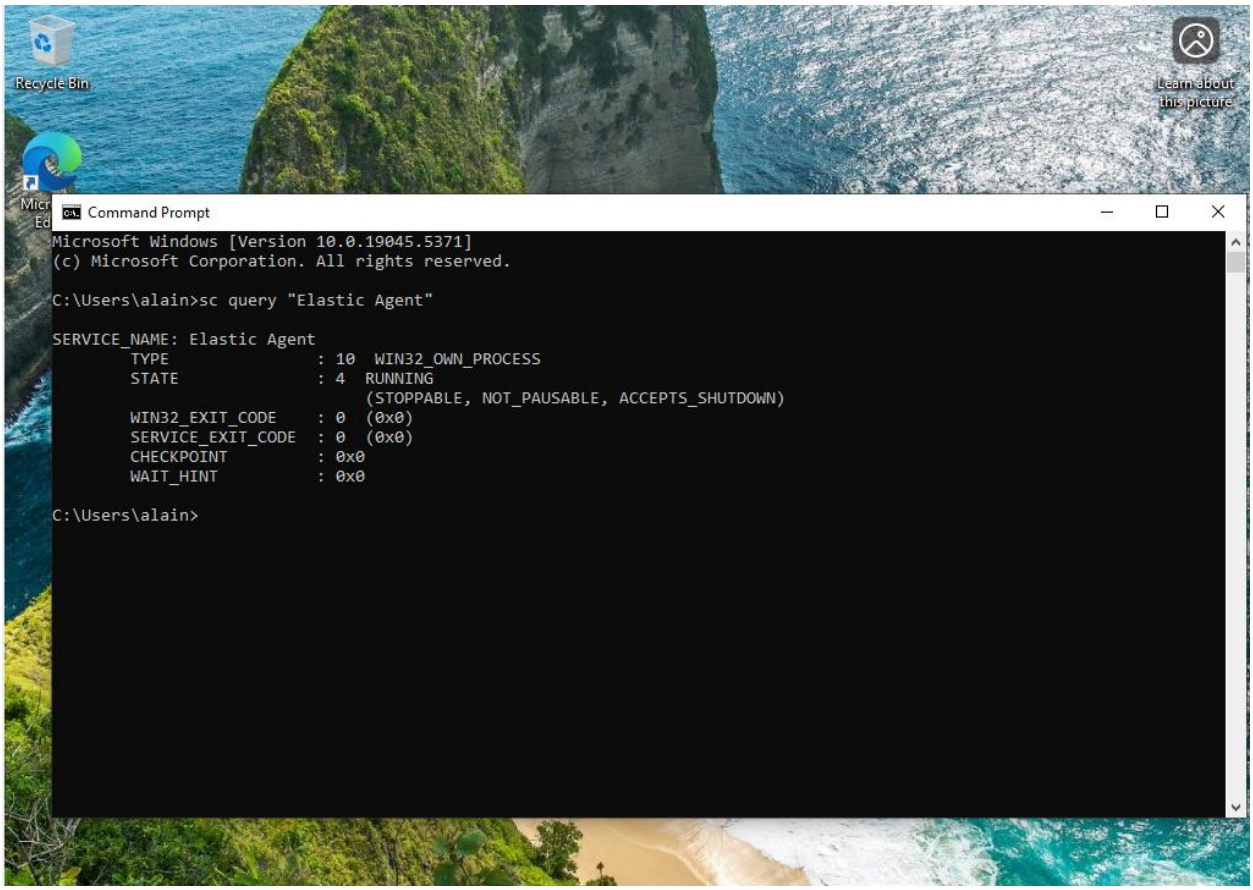


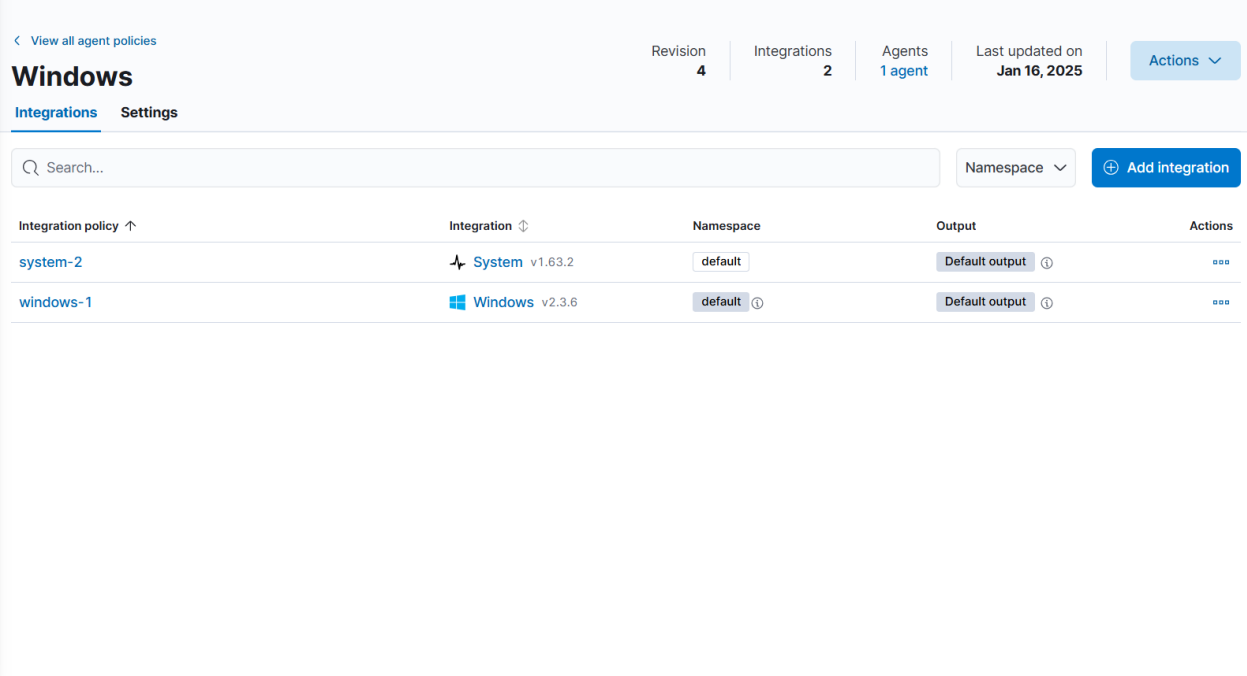
These two machines (DMZ and DMZ2) which are both running on Kali-Linux, are running the Honeypot and Suricata. The machine on the left is the honeypot, while the one on the right is Suricata. You can see that when DMZ attempts to ping DMZ2's address (192.168.250.100) the connection times out due to the firewall rules put in place. The same thing happens in reverse.

2. LAN (Windows)

After installing Windows 10 on my virtual machine and assigning it its IP from pfSense, I created the agent policy for the machine on Elastic Cloud, added the Windows integration, and installed Elastic Agent on the Windows machine to connect it to my Elastic Cloud. This was done with the

integration’s given command to run in Windows PowerShell with Administrator permissions. Below is a screenshot displaying the existence of Elastic Agent on the Windows machine, and its status as “Running”.

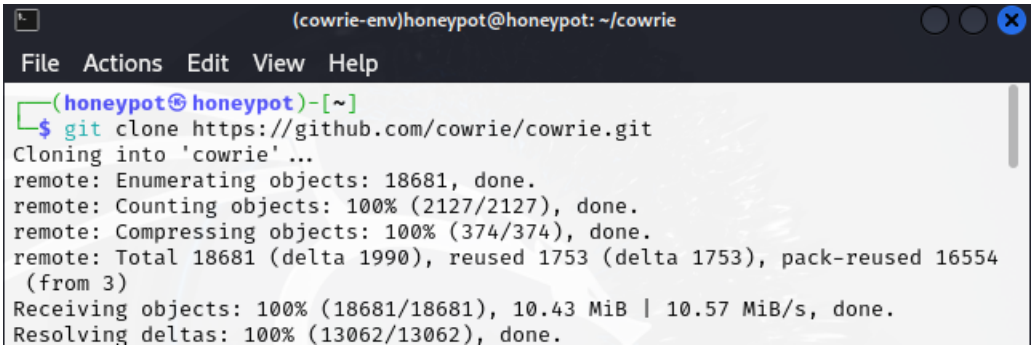




All logs and excerpts successfully ingested and transported to Elastic Cloud will be displayed at the end of the report underneath the Elastic section.

3. DMZ (Honeypot)

The DMZ internal network is on a Kali-Linux machine running the Honeypot. In order to achieve this, I installed Cowrie using GitHub and ran it with a virtual environment. This method is far more consistent for being able to access its logs, as while as being able to successfully transfer them to Elastic Cloud using Elastic Cloud's "Custom Logs" integration. I installed Filebeat as well in order for the logs to be properly transferred to Elastic Cloud. The screenshots below are the installation process, as well as starting the Cowrie service.



```
┌──(honeypot㉿honeypot)-[~/cowrie]
└─$ python3 -m venv cowrie-env

┌──(honeypot㉿honeypot)-[~/cowrie]
└─$ source cowrie-env/bin/activate
```

```
  GNU nano 8.3                    cowrie.cfg
# (default: not specified)
#sensor_name=myhostname

# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment
#
# (default: svr04)
hostname = svr04


# Directory where to save log files in.
#
# (default: log)
log_path = /var/log/cowrie


# Directory where to save downloaded artifacts in.
#
# (default: downloads)
download_path = ${honeypot:state_path}/downloads


# Directory for static data files

^G Help        ^O Write Out   ^F Where Is    ^K Cut         ^T Execute
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify
```

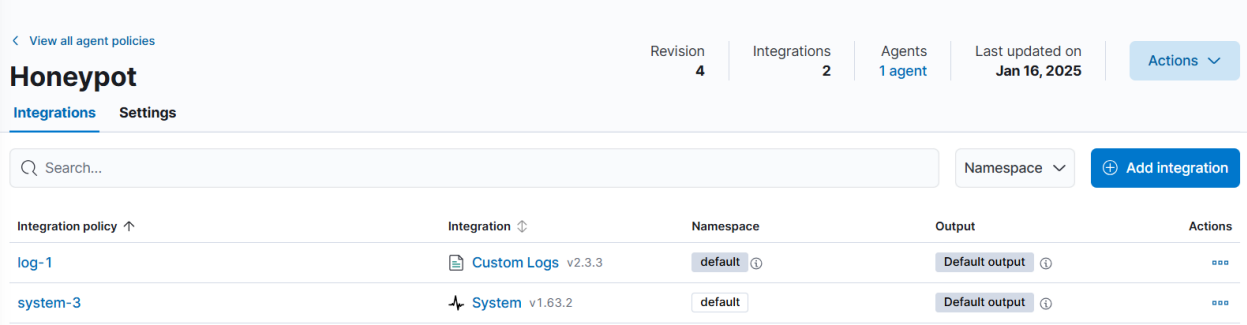
```
┌──(honeypot㉿honeypot)-[~/cowrie]
└─$ ./bin/cowrie start

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

Using default Python virtual environment "/home/honeypot/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger
cowrie.python.logfile.logger cowrie ]...
/home/honeypot/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/s
sh/transport.py:105: CryptographyDeprecationWarning: TripleDES has been moved
 to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be rem
oved from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/home/honeypot/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/s
sh/transport.py:112: CryptographyDeprecationWarning: TripleDES has been moved
 to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be rem
oved from cryptography.hazmat.primitives.ciphers.algorithms in 48.0.0.
  b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
```

Here are screenshots for the integration to Elastic Cloud, as well as the installation process for Filebeat and configuration for the log path.

```
┌──(honeypot㉿honeypot)-[~]
└─$ curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat
-8.0.0-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Curre
nt
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--
  0 34.2M    0  202k    0     0   201k      0  0:02:54  0:00:01  0:02:53  201
 59 34.2M   59 20.4M    0     0  10.4M      0  0:00:03  0:00:01  0:00:02 10.4
100 34.2M  100 34.2M    0     0  13.9M      0  0:00:02  0:00:02 --:--:-- 14.0
M

┌──(honeypot㉿honeypot)-[~]
└─$ sudo dpkg -i filebeat-8.0.0-amd64.deb
Selecting previously unselected package filebeat.
(Reading database ... 418214 files and directories currently installed.)
Preparing to unpack filebeat-8.0.0-amd64.deb ...
Unpacking filebeat (8.0.0) ...
Setting up filebeat (8.0.0) ...
Processing triggers for kali-menu (2024.4.0) ...
```

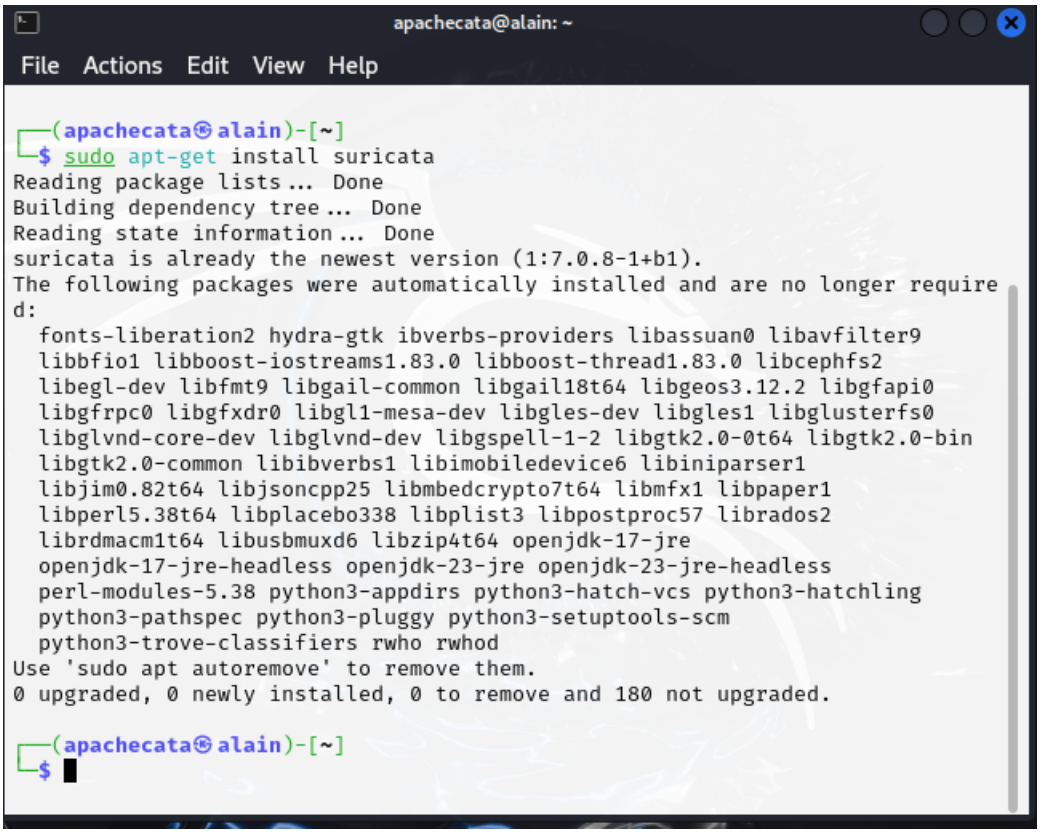
```
# ============================== Filebeat inputs ===============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Change to true to enable this input configuration.
  enabled: false

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/cowrie/*.log
    #- c:\programdata\elasticsearch\logs\*
```

The log path in Filebeat's config file is set to *.log to make sure that any file ending in .log is read for Elastic Cloud. The folder is Cowrie's dedicated folder for Cowrie's logs, so despite the path being set to *.log to read any log file in the folder, it will only read Cowrie's logs due to that being the only log file in the folder. Here's a screenshot showing Elastic Cloud's integration with Honeypot.

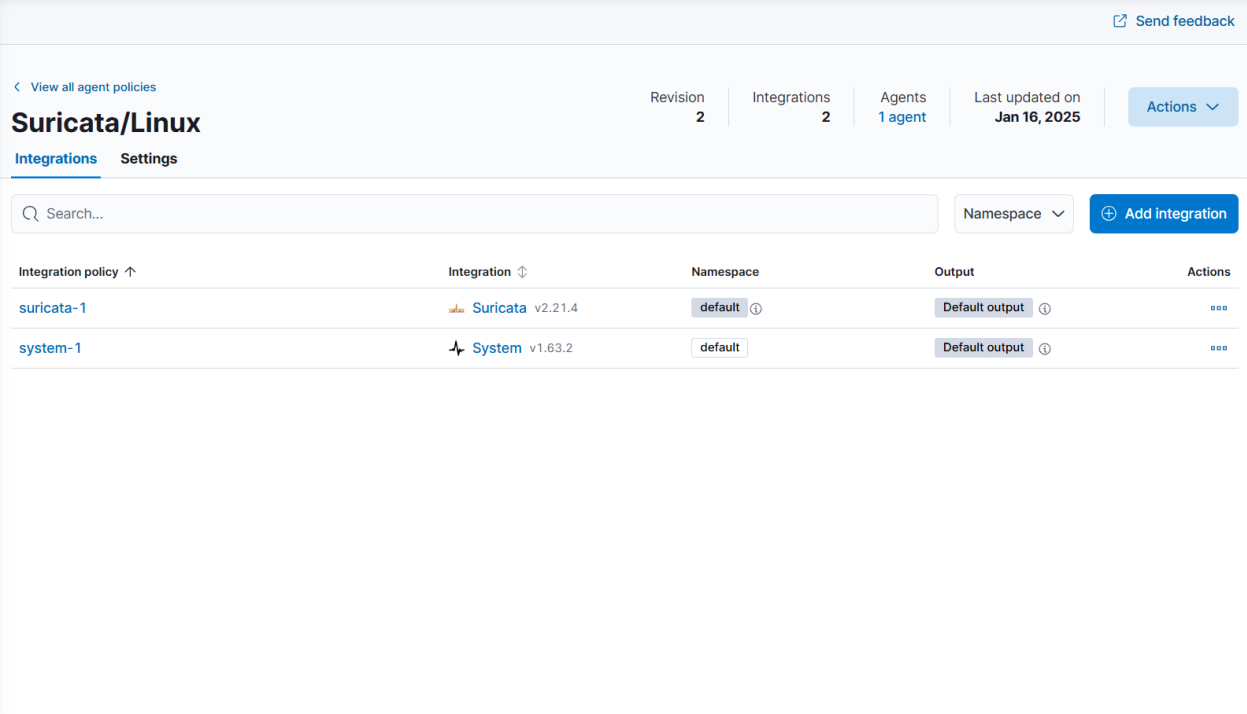


4. DMZ2 (Suricata)

DMZ2 is also a Kali-Linux machine, but running Suricata instead. The installation was very straight forward. Here's the installation process and proof of Suricata running successfully. The integration to Elastic Cloud was also very straight forward, due to Elastic Cloud having a native integration for Suricata.

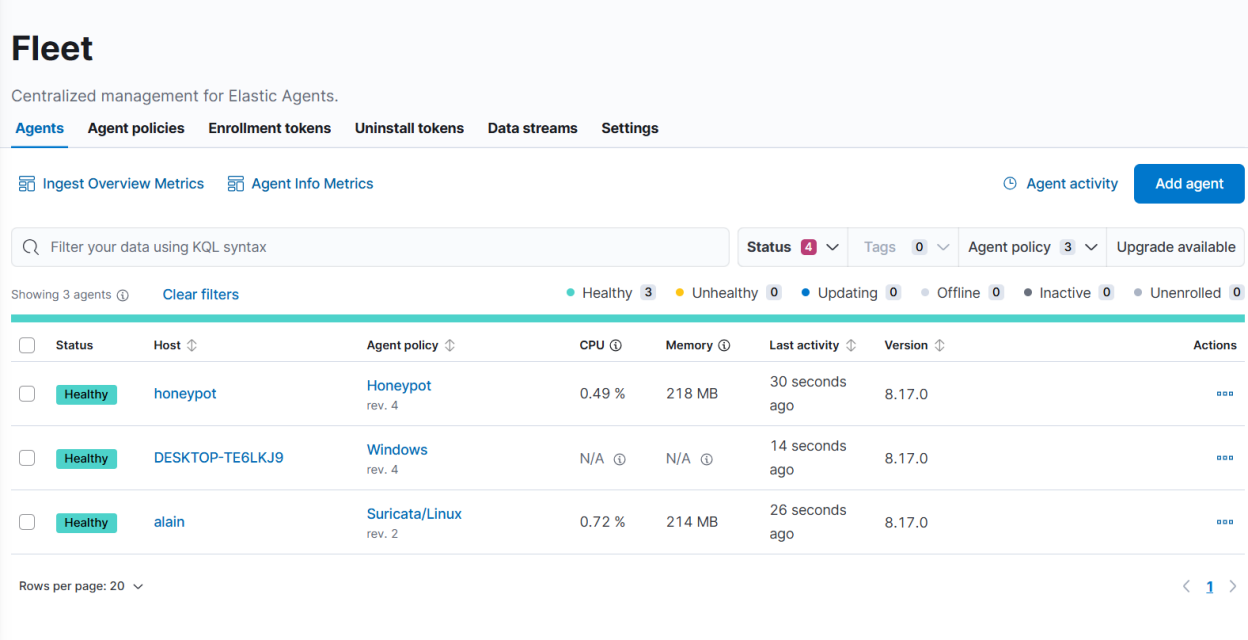

apachecata@alain: ~
File Actions Edit View Help
┌──(apachecata㉿alain)-[~]
└─$ sudo apt-get install suricata
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
suricata is already the newest version (1:7.0.8-1+b1).
The following packages were automatically installed and are no longer require
d:
fonts-liberation2 hydra-gtk ibverbs-providers libassuan0 libavfilter9
libbfio1 libboost-iostreams1.83.0 libboost-thread1.83.0 libcephfs2
libegl-dev libfmt9 libgail-common libgail18t64 libgeos3.12.2 libgfapi0
libgfrpc0 libgfxdr0 libgl1-mesa-dev libgles-dev libgles1 libglusterfs0
libglvnd-core-dev libglvnd-dev libgspell-1-2 libgtk2.0-0t64 libgtk2.0-bin
libgtk2.0-common libibverbs1 libimobiledevice6 libiniparser1
libjim0.82t64 libjsoncpp25 libmbedcrypto7t64 libmfx1 libpaper1
libperl5.38t64 libplacebo338 libplist3 libpostproc57 librados2
librdmacm1t64 libusbmuxd6 libzip4t64 openjdk-17-jre
openjdk-17-jre-headless openjdk-23-jre openjdk-23-jre-headless
perl-modules-5.38 python3-appdirs python3-hatch-vcs python3-hatchling
python3-pathspec python3-pluggy python3-setuptools-scm
python3-trove-classifiers rwho rwhod
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 180 not upgraded.
┌──(apachecata㉿alain)-[~]
└─$


┌──(apachecata㉿alain)-[~]
└─$ sudo systemctl status suricata
● suricata.service - Suricata IDS/IDP daemon
Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; pres>
Active: active (running) since Fri 2025-01-17 16:14:23 CET; 2min 4s ago
Invocation: b3946c8fd97141d3b62be269ff5668fc
Docs: man:suricata(8)
man:suricatasc(8)
https://suricata.io/documentation/
Process: 791 ExecStart=/usr/bin/suricata -D --af-packet -c /etc/suricata>
Main PID: 793 (Suricata-Main)
Tasks: 10 (limit: 4557)
Memory: 85.4M (peak: 87M)
CPU: 770ms
CGroup: /system.slice/suricata.service
└─793 /usr/bin/suricata -D --af-packet -c /etc/suricata/suricat>
Jan 17 16:14:23 alain systemd[1]: Starting suricata.service - Suricata IDS/I>
Jan 17 16:14:23 alain suricata[791]: i: suricata: This is Suricata version 7>
Jan 17 16:14:23 alain systemd[1]: Started suricata.service - Suricata IDS/ID>
lines 1-18/18 (END)



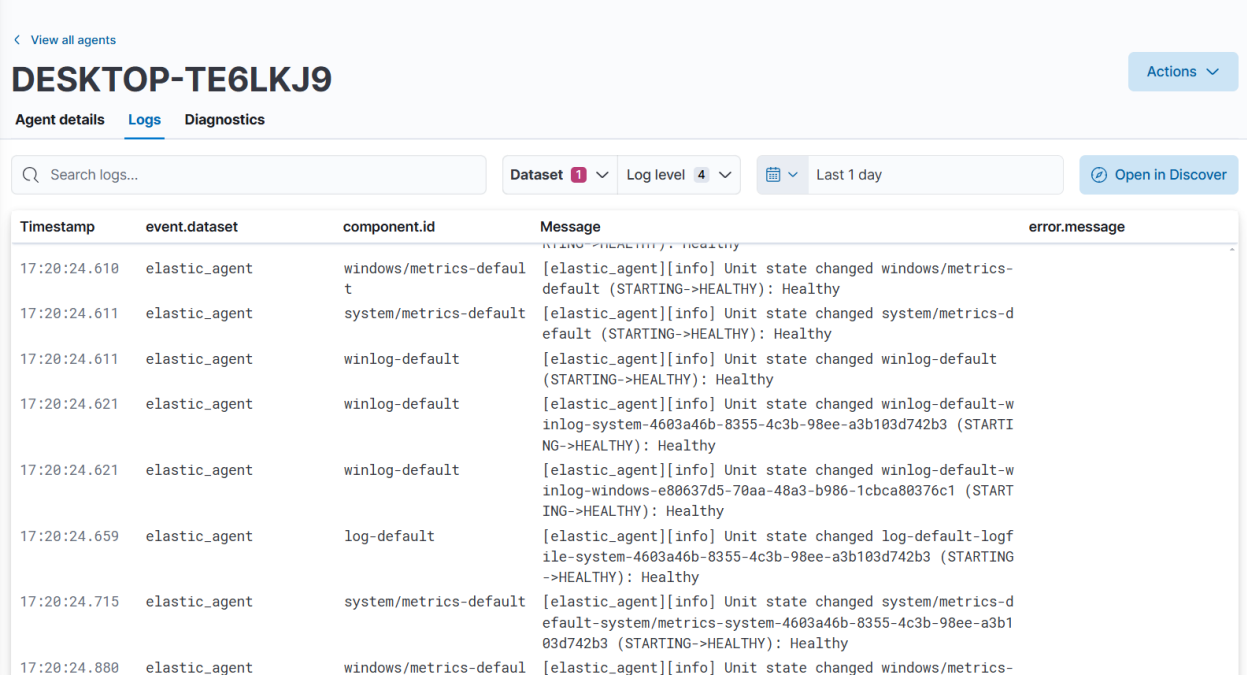
5. Elastic Cloud

The process for Elastic Cloud and connecting each internal network and their services successfully to the site was very efficient and linear. Both Windows and Suricata had their own native integrations to use, and retrieving their logs was a matter of a couple clicks and 1 command each. For Honeypot, since there is no native integration available, the Custom Logs integration is the best option. It was simply giving Elastic Cloud the log file path, and installing Elastic Agent onto the host.



This screenshot displays all the machines connected to the Elastic Cloud. Below this will be screenshots of the logs being successfully retrieved, as well as individual logs retrieved from Elastic Cloud to show more in detail.

Windows:



`host.os.name.text`

Windows 10 Home
@timestamp

Jan 18, 2025 @ 17:15:17.648
agent.ephemeral_id

fa1b7867-cdb4-4653-aacf-245b911bb471
agent.id

fd50d10d-29ce-434a-a0f9-de2f033a7d9d
agent.name

DESKTOP-TE6LKJ9
agent.type

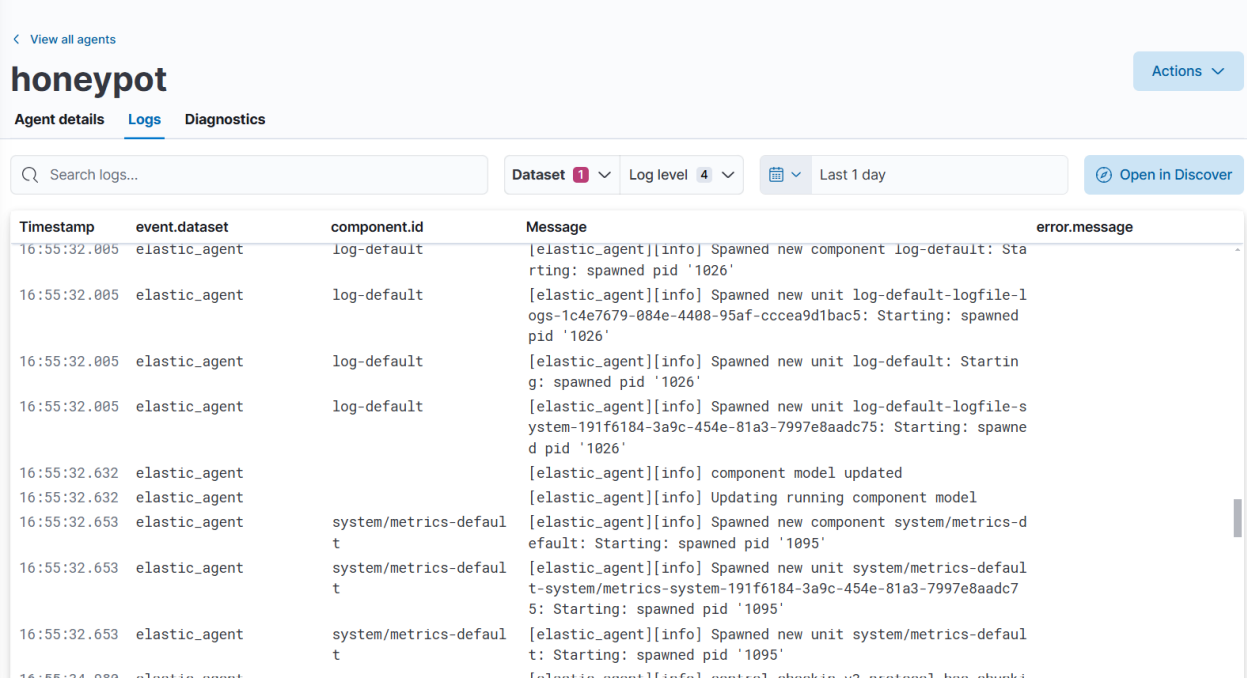
filebeat
agent.version

8.17.0
component.binary

metricbeat
component.dataset

Honeypot:

In order to check if Elastic Cloud was receiving and ingesting Cowrie logs through Filebeat properly, I ran an SSH connection and checked if Elastic Cloud received a log entry related to Honeypot SSH, rather than only reading Filebeat's process. Below is an individual log referencing Honeypot SSH, showing that Elastic Cloud is receiving Cowrie's logs due to Filebeat successfully ingesting Cowrie's logs through the path that was put in place in the configuration file.



message

2025-01-18T13:21:42.926428Z [-] Ready to accept SSH connections

@timestamp

Jan 18, 2025 @ 13:21:43.147

agent.ephemeral_id

2fdc759d-e0c9-475d-b535-41870574483f

agent.id

0157a5f1-e573-4d50-a5ec-743f5f9ff3d9

agent.name

honeypot

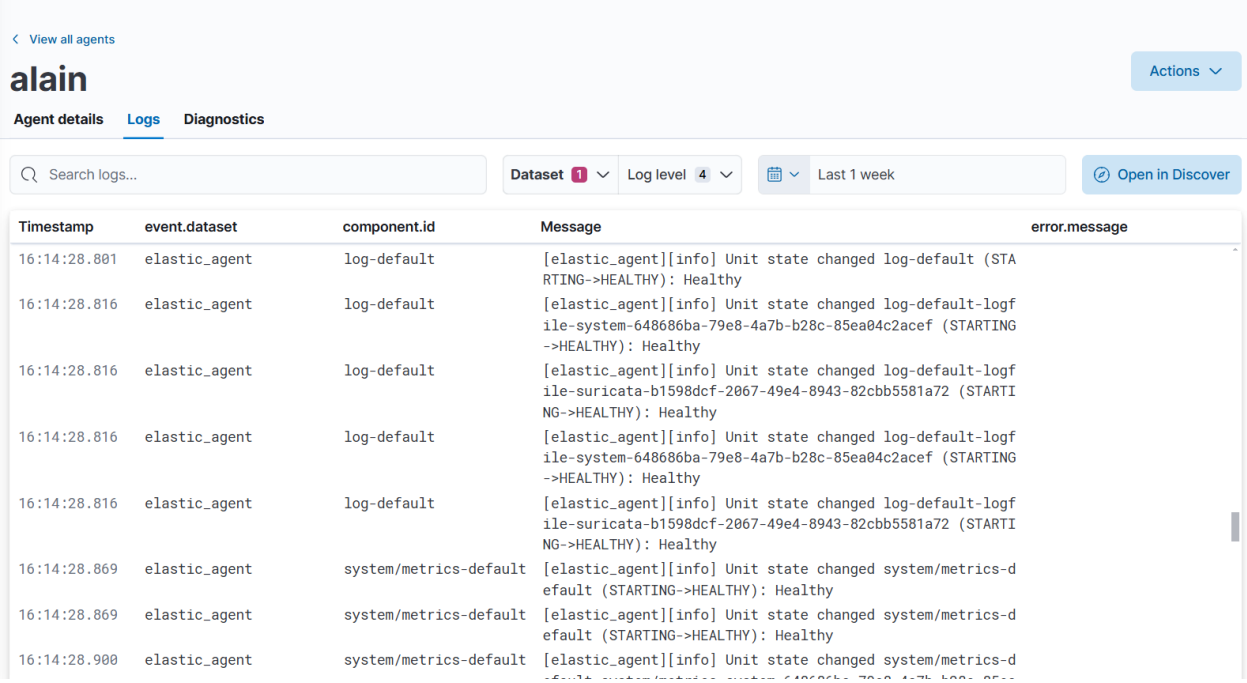
agent.type

filebeat

agent.version

8.17

Suricata:



@timestamp

Jan 17, 2025 @ 16:17:16.350
agent.ephemeral_id

cf184e04-68c8-4bf8-a49d-5594a8e160a0
agent.id

7bcc064f-4827-4937-9ee9-4f9af5a71b01
agent.name

alain
agent.type

filebeat
agent.version

8.17.0
data_stream.dataset

suricata.eve
data_stream.namespace

End of Report