Part 1 Project Goal

Project Goal

The goal of this project was to deploy a public-facing SSH honeypot in a controlled cloud environment to observe and collect real attacker activity. The honeypot was designed to safely emulate an exposed SSH service while preventing access to any real production systems or internal network resources.

Cowrie was used as the SSH honeypot platform to capture attacker interactions, including connection attempts, login attempts, submitted credentials, executed commands, session activity, and uploaded files. These logs were then forwarded into Wazuh, allowing the honeypot activity to be centrally collected, parsed, and investigated through a SIEM workflow.

The project also involved building a dedicated Cowrie-focused SIEM dashboard in Wazuh. This dashboard provides visibility into key attacker behaviours, including successful and failed login attempts, source IP activity, command execution, uploaded files, and recent raw Cowrie events. The final objective was to create a practical investigation environment that demonstrates cloud deployment, log forwarding, SIEM integration, detection engineering, and security monitoring skills.

In summary, this project aimed to:

- Deploy a public-facing SSH honeypot.
- Collect real-world attacker activity.
- Forward Cowrie logs into Wazuh.
- Build a SIEM dashboard for investigation and analysis.

Part 2 Architecture

Architecture

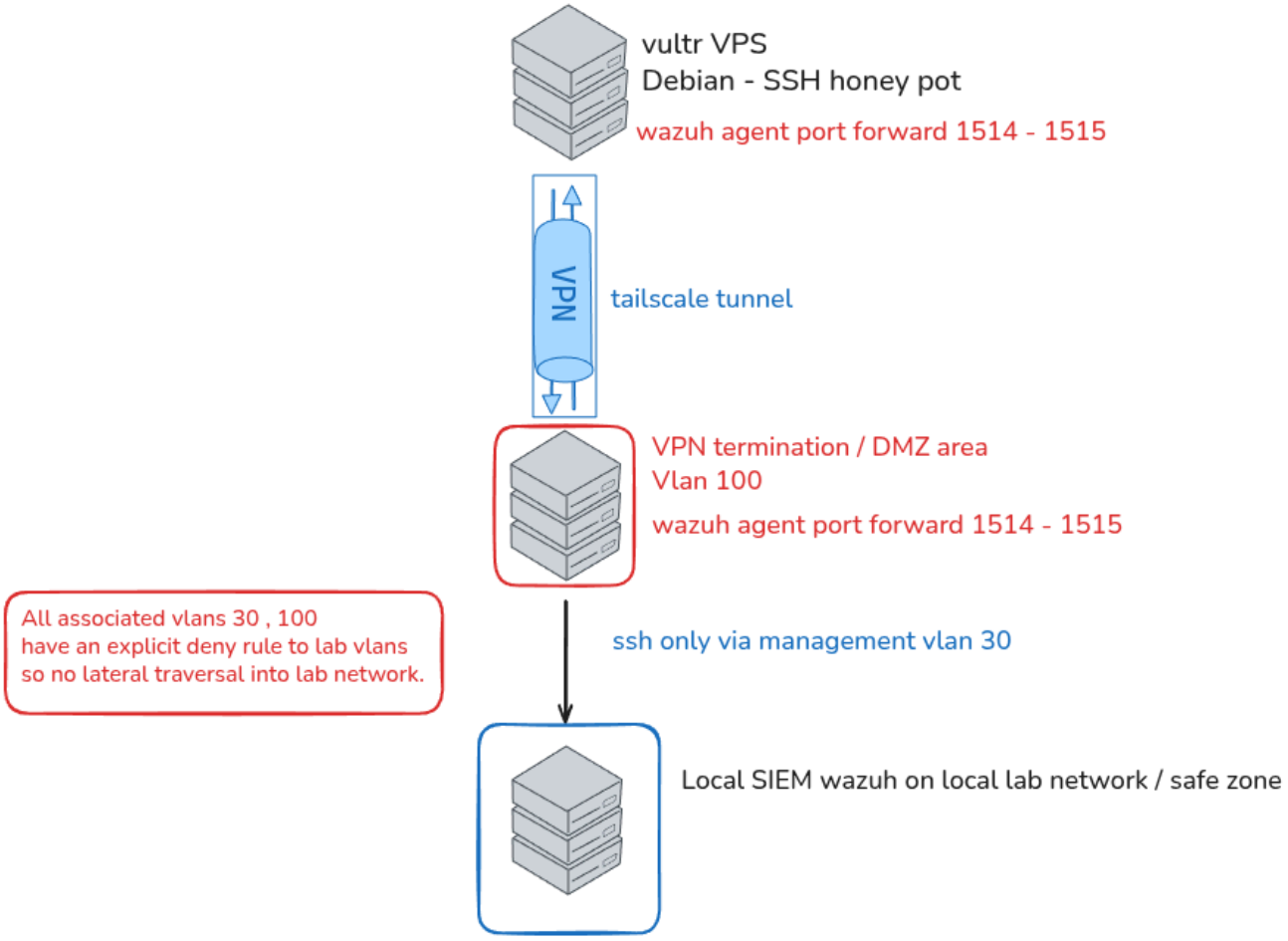
The project was designed around a hybrid cloud and on-premise security monitoring architecture. A public-facing Vultr VPS was used to host the SSH honeypot, while the Wazuh manager and SIEM dashboard were hosted on-premise inside a segmented lab environment. This allowed attacker activity to be collected from the internet while keeping the central monitoring infrastructure isolated from direct public exposure.

High-Level Architecture

The main components of the architecture were:

- Vultr VPS
- Cowrie SSH honeypot
- Tailscale/private tunnel
- On-premise Wazuh manager
- Segmented VLAN and firewall rules
- Wazuh dashboards

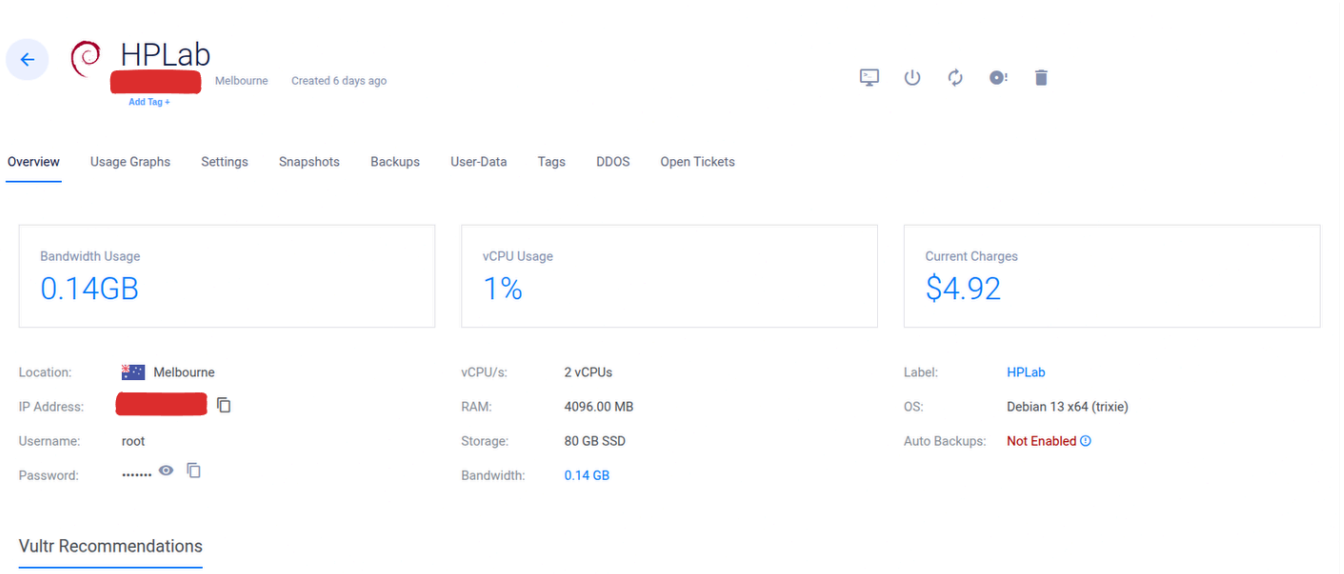
The Vultr VPS acted as the exposed cloud endpoint. Cowrie was deployed on this system to emulate an SSH service and capture attacker behaviour. Instead of exposing the Wazuh manager directly to the internet, log forwarding occurred over a private tunnel, reducing the attack surface of the monitoring environment.



Vultr VPS

A Vultr VPS was deployed as the public-facing component of the lab. This server was intentionally exposed to the internet on SSH in order to attract automated scans, brute-force attempts, and attacker interactions.

The VPS was not treated as a trusted internal system. It was separated from the on-premise environment and only allowed to communicate back to the Wazuh infrastructure through controlled private connectivity. This design reduced the risk of a compromised honeypot being used as a pivot point into the internal network.



Cowrie Honeypot

Cowrie was installed on the Vultr VPS to emulate an SSH server. The honeypot captured attacker activity such as:

- SSH connection attempts
- Failed login attempts
- Successful honeypot logins
- Usernames and passwords submitted by attackers
- Commands entered during interactive sessions
- Session IDs
- Source IP addresses
- Uploaded files

Cowrie generated structured JSON logs, which made the activity suitable for forwarding into Wazuh and parsing into searchable SIEM fields.

```
jono@HPLab:~$ sudo supervisorctl status cowrie
[sudo] password for jono:
cowrie                           RUNNING   pid 1183, uptime 1 day, 21:39:28
```

Tailscale / Private Tunnel

A private tunnel was used to connect the public VPS back to the on-premise monitoring environment. This avoided exposing the Wazuh manager directly to the public internet.

The tunnel allowed the Wazuh agent on the VPS to communicate with the Wazuh manager over private addressing. This was important because the honeypot needed to forward logs reliably, but the SIEM itself needed to remain protected behind firewall rules and network segmentation.

**1 to 1 Tailscale VPN tunnel - VPS to on prem

```
jono@HPLab:~$ tailscale status
[redacted]  hplab   JonathonReyesPortfolio@  linux   -
            debian  JonathonReyesPortfolio@  linux   active; relay "syd", tx 42476612 rx 11030132
```

**

We used netcat to test port forward rules

```
jono@HPLab:~$ nc -nvz -w 3 192.168.1.200 1514
Connection to 192.168.1.200 1514 port [tcp/*] succeeded!
jono@HPLab:~$ nc -nvz -w 3 192.168.1.200 1515
Connection to 192.168.1.200 1515 port [tcp/*] succeeded!
```

Ensured lateral movement disabled

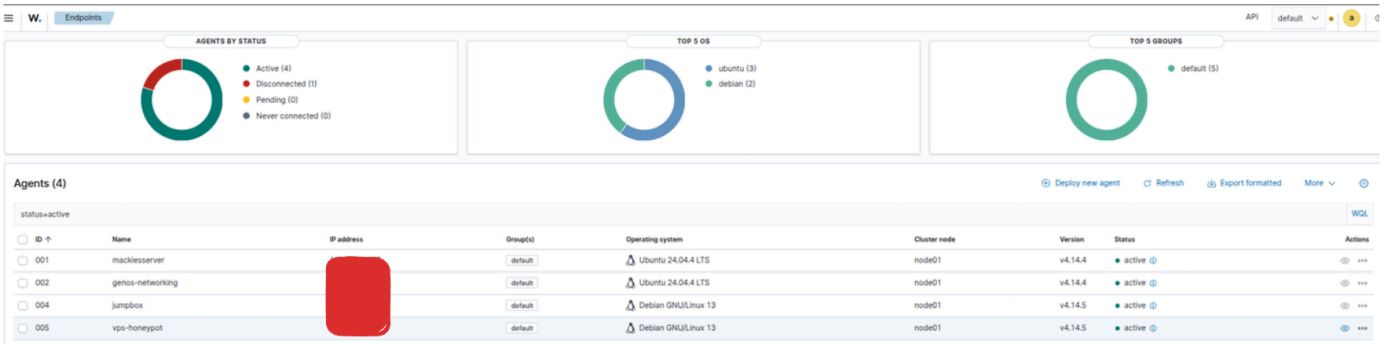
```
jono@HPLab:~$ ping 192.168.1.200
PING 192.168.1.200 (192.168.1.200) 56(84) bytes of data.
```

On-Premise Wazuh Manager

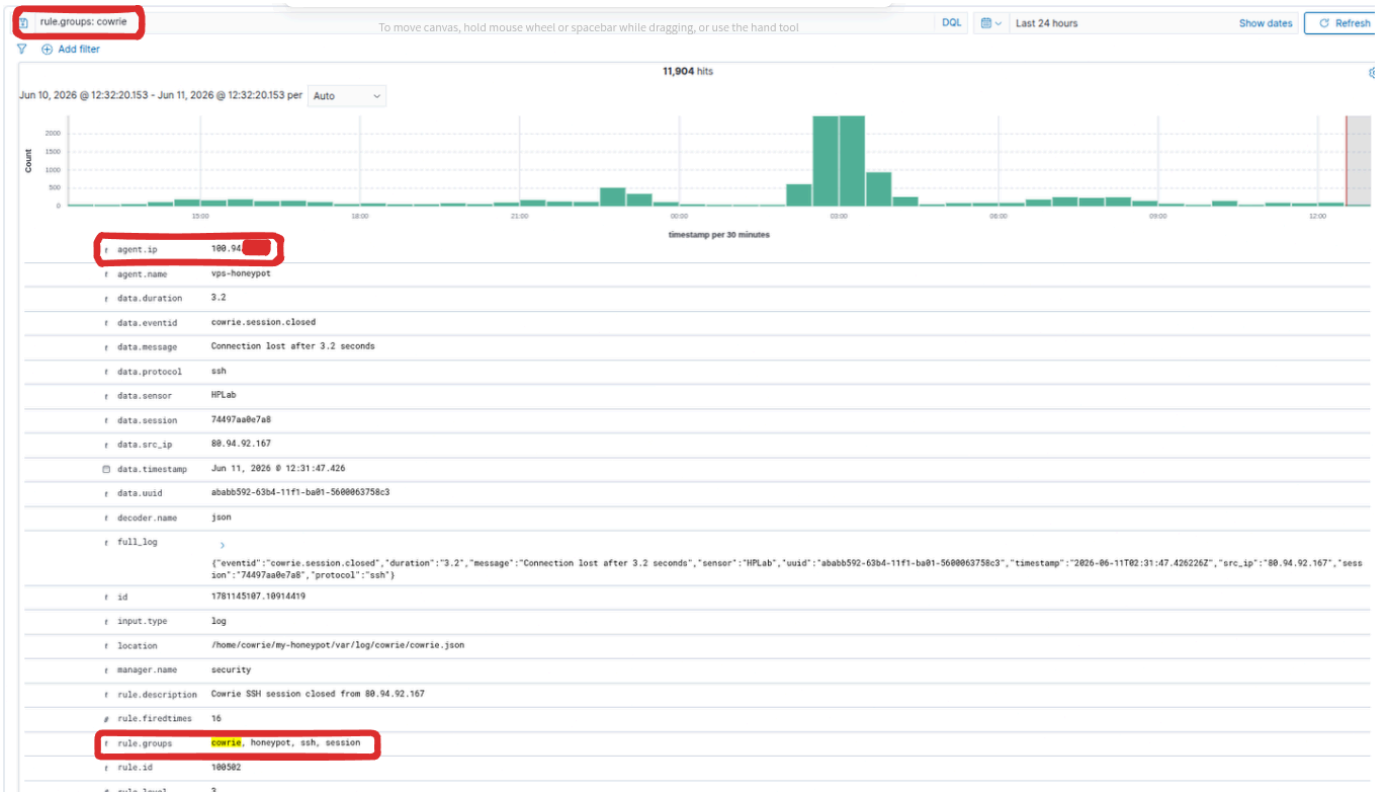
The Wazuh manager was hosted on-premise and used as the central point for log collection, alerting, and investigation. The Wazuh agent on the Vultr VPS forwarded Cowrie logs to the Wazuh manager, where they were decoded, indexed, and made available in the dashboard.

This setup demonstrated a realistic monitoring pattern: internet-facing endpoint in the cloud, private log forwarding, and centralised SIEM infrastructure kept away from direct internet exposure.

```
jono@HPLab:~$ sudo systemctl status wazuh-agent --no-pager
● wazuh-agent.service - Wazuh agent
     Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service; enabled; preset: enabled)
     Active: active (running) since Wed 2026-06-10 00:53:30 UTC; 1 day 1h ago
```



Rules visible from agent installed on VPS

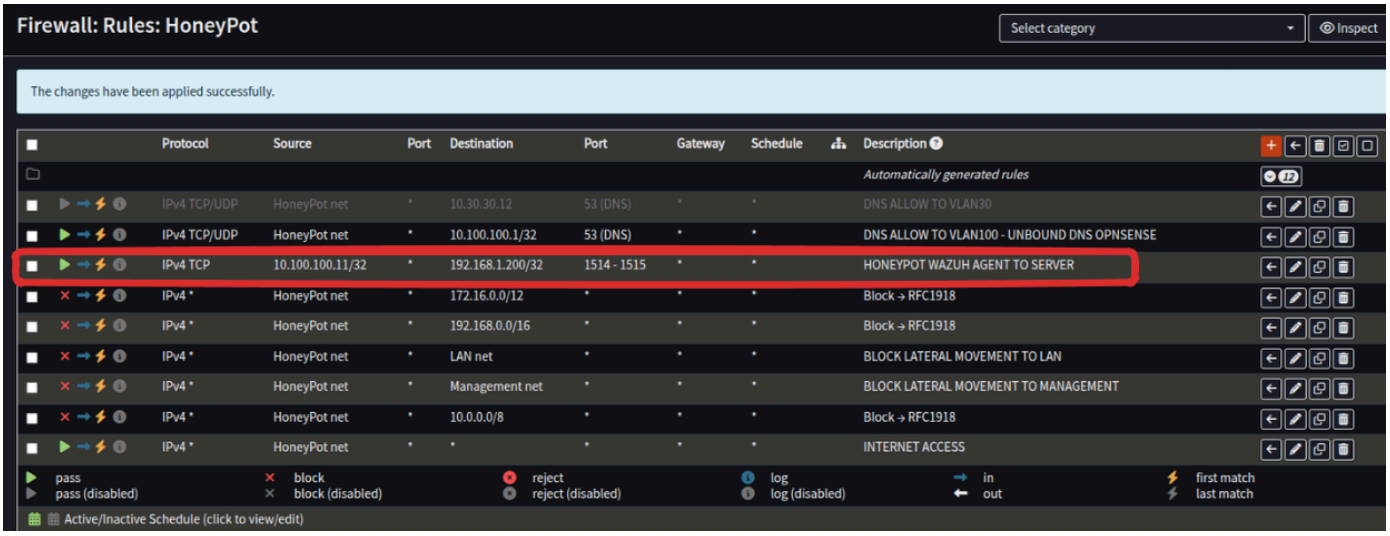


Segmented VLAN and Firewall Rules

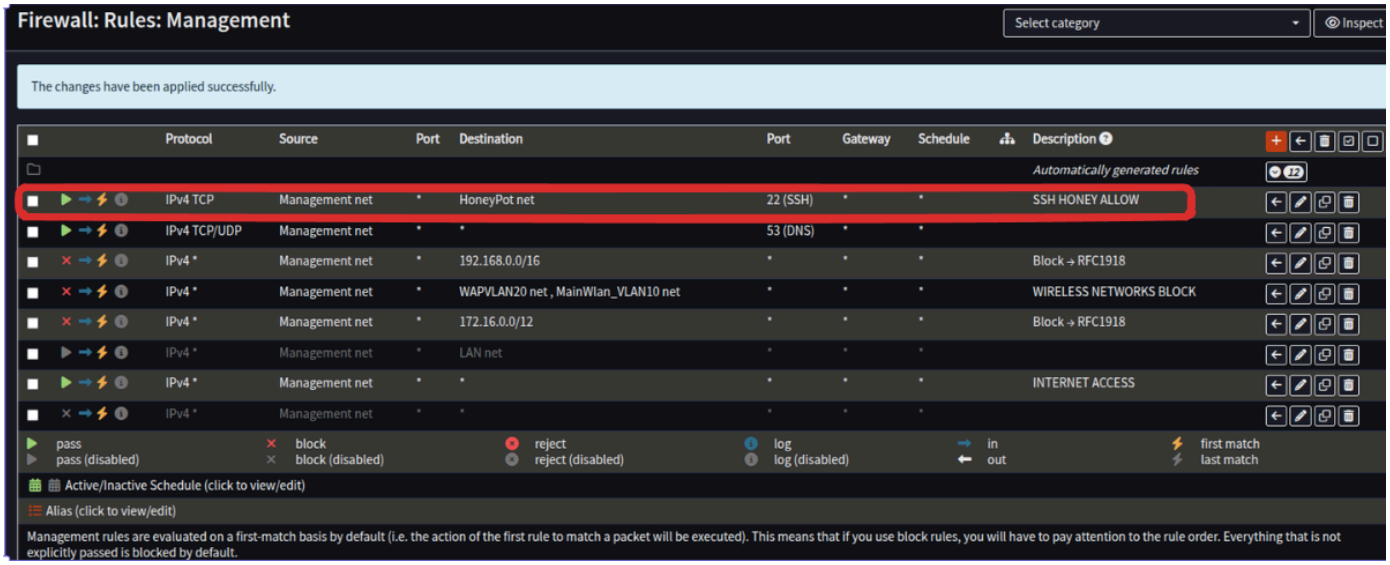
The on-premise side of the architecture used VLAN segmentation and firewall rules to limit access between systems. The Wazuh manager was placed in a controlled network segment, while access from the honeypot path was restricted to only the required Wazuh communication ports.

The key principle was least privilege: the VPS only needed to send logs to Wazuh, so unnecessary access to internal systems was blocked. This helped maintain separation between the exposed honeypot and the rest of the lab environment.

Firewall rules vlan100



**Firewall rule vlan 30 (ssh)



**

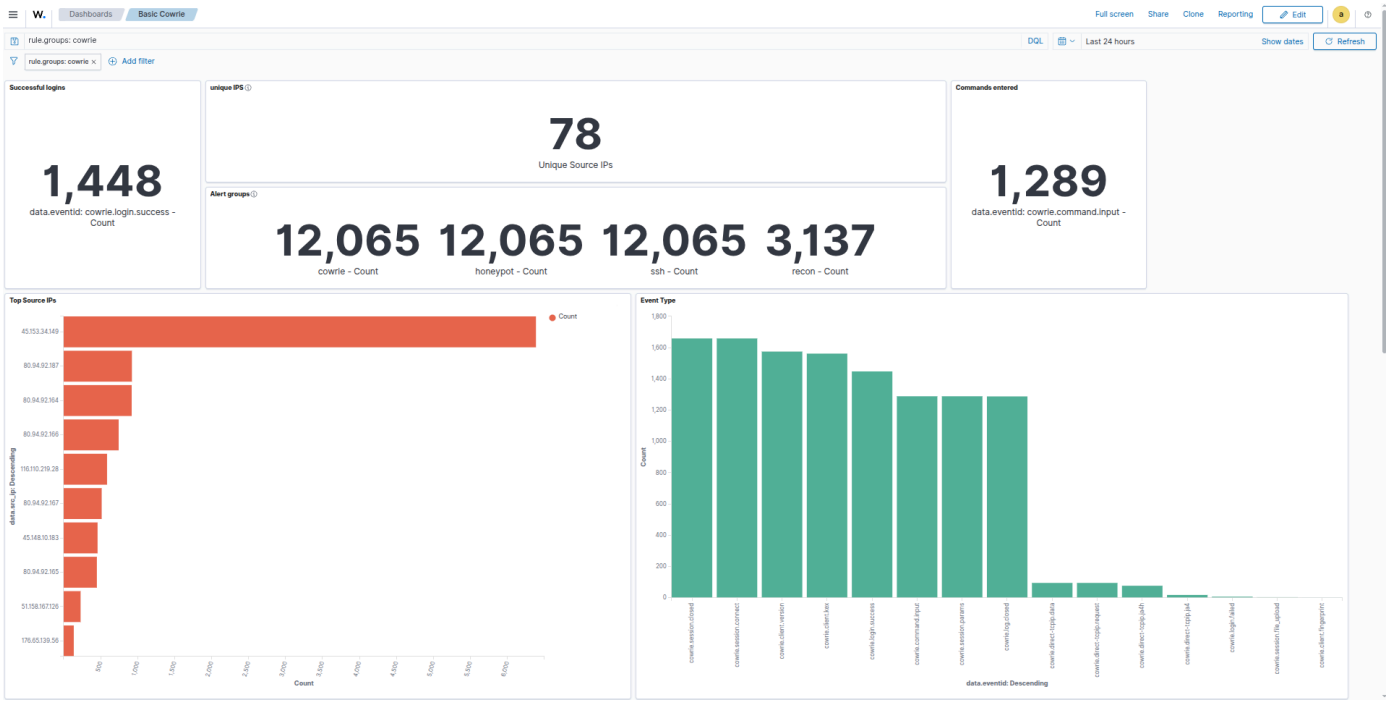
Wazuh Dashboards

A dedicated Wazuh dashboard was created to investigate Cowrie honeypot activity. The dashboard focused on attacker behaviour rather than generic system metrics.

The dashboard included panels for:

- Cowrie event volume
- Source IP activity
- Successful and failed login attempts
- Common usernames
- Common passwords
- Commands executed by attackers
- File upload events
- Recent Cowrie events for analyst investigation

The most important panel was the recent events table, which provided a raw investigation view of attacker sessions. This allowed suspicious spikes, source IPs, credentials, commands, and session IDs to be inspected quickly.

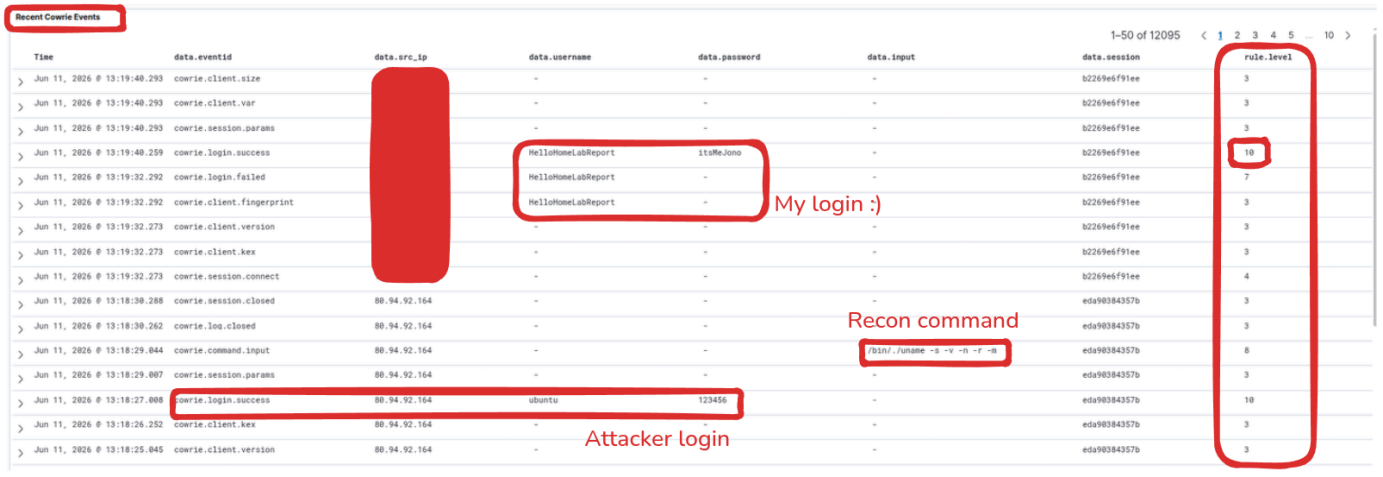


(Dashboard continues below)

Custom event search for dashboard

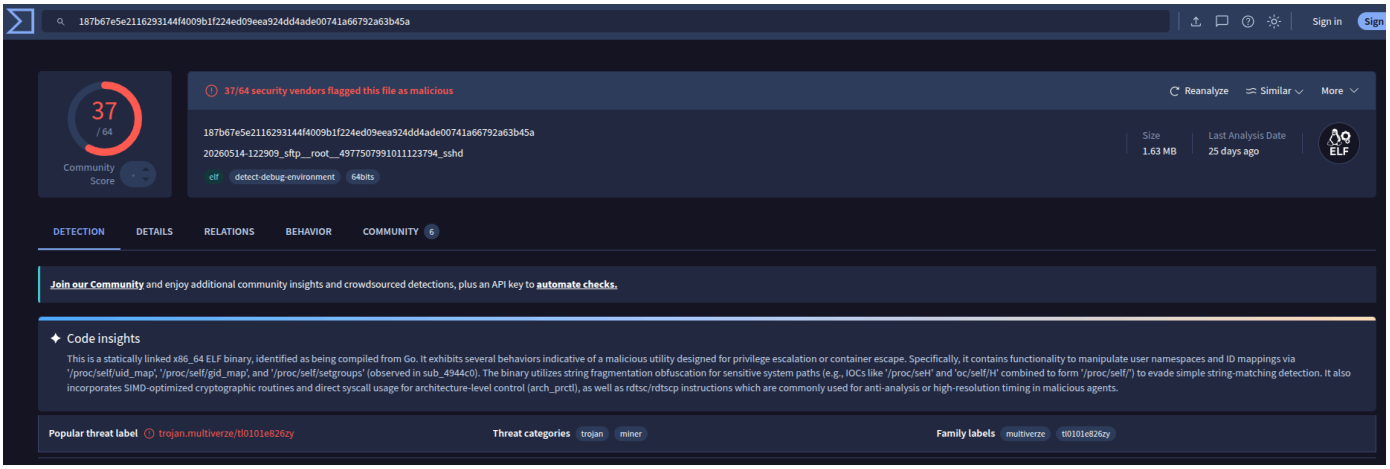
Severity level 12 onwards - ie SFTP connections - file uploads to honeypot

Lower severity - but interacting with said honeypot



Inaction view

As we can see the SHA256 hash is included in the file upload - we can use this artefact to further investigate file on virustotal.



The malware detection `trojan.multiverze/tl0101e826zy` was observed in relation to a Cowrie file upload event. Based on the available telemetry, this activity is mapped to **MITRE ATT&CK T1105 — Ingress Tool Transfer**, because the attacker transferred a suspected malicious payload into the honeypot environment.

Architecture Summary

This architecture provided a safe and practical way to observe real attacker behaviour against a public SSH service while keeping the monitoring infrastructure protected. The design combined cloud exposure, honeypot telemetry, private tunnelling, on-premise SIEM ingestion, network segmentation, and dashboard-based investigation.

Overall, the architecture demonstrated the following security engineering concepts:

- Public cloud honeypot deployment
- Controlled exposure of an intentionally vulnerable service
- Private log forwarding to an internal SIEM
- Network segmentation and firewall control
- Wazuh agent and manager integration
- Custom dashboard creation for attacker activity investigation
- Practical incident analysis using real honeypot telemetry