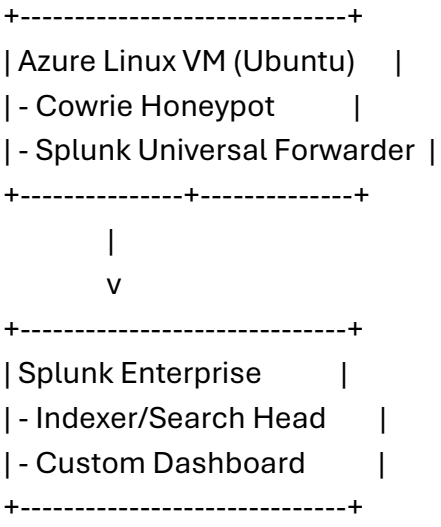
Cowrie Honeypot with Splunk SIEM Integration

Author: Omar Al-Fayyadh

Executive Summary

This report documents an end-to-end honeypot-to-SIEM pipeline using the Cowrie SSH/Telnet honeypot and Splunk Enterprise. The environment is deployed on Microsoft Azure, where Cowrie captures live attacker activity from the public internet. Logs are shipped with Splunk Universal Forwarder (UF) to Splunk Enterprise for parsing, indexing, enrichment, and visualization via a custom dashboard. The goal is to provide practical experience with threat monitoring, log ingestion, and security analytics while generating a reusable blueprint for SOC workflows and cloud lab demonstrations.

Architecture



Environment Details

Component	Details
Cloud	Microsoft Azure
VM Size	Standard B1s/B2s (lab)
OS	Ubuntu 22.04 LTS
Public Exposure	NSG allows TCP 22, 2222, 2223
Honeypot	Cowrie (Docker or venv install)
Log Shipper	Splunk Universal Forwarder
SIEM	Splunk Enterprise (local or remote indexer)
Index	cowrie (custom)

Deployment Guide (Condensed)

1. Provision an Ubuntu VM in Azure with a public IP. Attach a Network Security Group allowing inbound TCP 22, 2222, 2223 (or as desired).
2. Install prerequisites and Cowrie (Docker method shown below).
3. Install Splunk Universal Forwarder and configure inputs/outputs.
4. On Splunk Enterprise, create index 'cowrie', source type mappings, and dashboard panels.
5. Harden VM and monitor using scheduled health checks and resource limits.

Cowrie (Docker) Quick Start

 # Install Docker & dependencies

```
sudo apt-get update && sudo apt-get install -y docker.io docker-compose
sudo systemctl enable --now docker
```

 # Deploy Cowrie honeypot container

```
sudo docker run -d --name cowrie \
 -p 2222:2222 -p 2223:2223 \
 -e COWRIE_USER=cowrie \
 -e COWRIE_GROUP=cowrie \
 -v /opt/cowrie/var:/cowrie/var \
 -v /opt/cowrie/etc:/cowrie/etc \
 cowrie/cowrie:latest
```

Splunk Universal Forwarder Setup (on Cowrie VM)

wget -O splunkforwarder.tgz 'https://download.splunk.com/products/universalforwarder/releases/9.2.0/linux/splunkforwarder-9.2.0-Linux-x86_64.tgz'
sudo tar -xzvf splunkforwarder.tgz -C /opt
sudo /opt/splunkforwarder/bin/splunk start --accept-license
sudo /opt/splunkforwarder/bin/splunk enable boot-start
sudo /opt/splunkforwarder/bin/splunk restart

Configuration Samples

[monitor:///opt/cowrie/var/log/cowrie/cowrie.json]
sourcetype = cowrie:json
index = cowrie
disabled = false

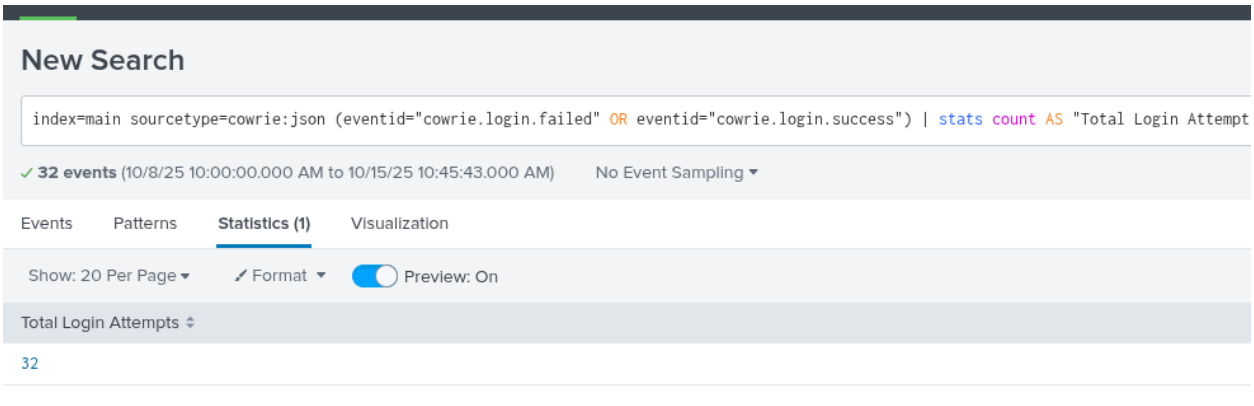
[tcpout]
defaultGroup = default-autolb-group
[tcpout:default-autolb-group]
server = SPLUNK_INDEXER_IP:9997
[indexAndForward]
index = false

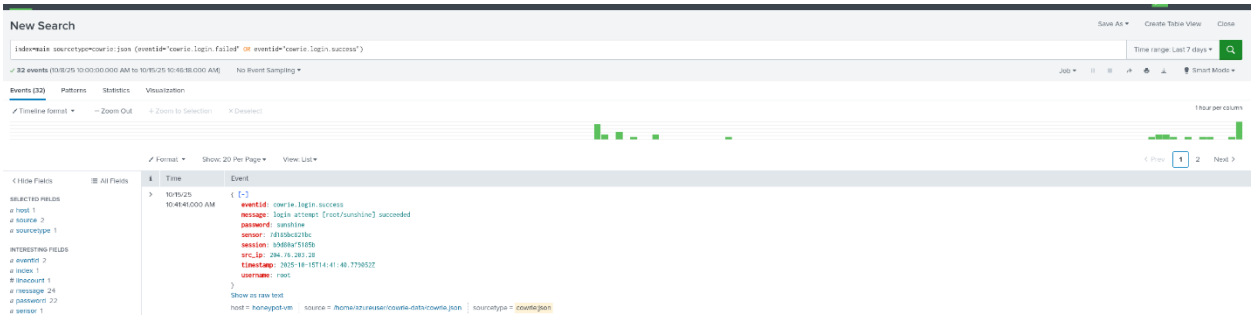
[cowrie:json]
DATETIME_CONFIG =
KV_MODE = json
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
TRUNCATE = 0
TIME_FORMAT = %Y-%m-%dT%H:%M:%S
TIME_PREFIX = "timestamp":\s*"

Splunk Dashboard Queries (SPL)

Total login attempts:

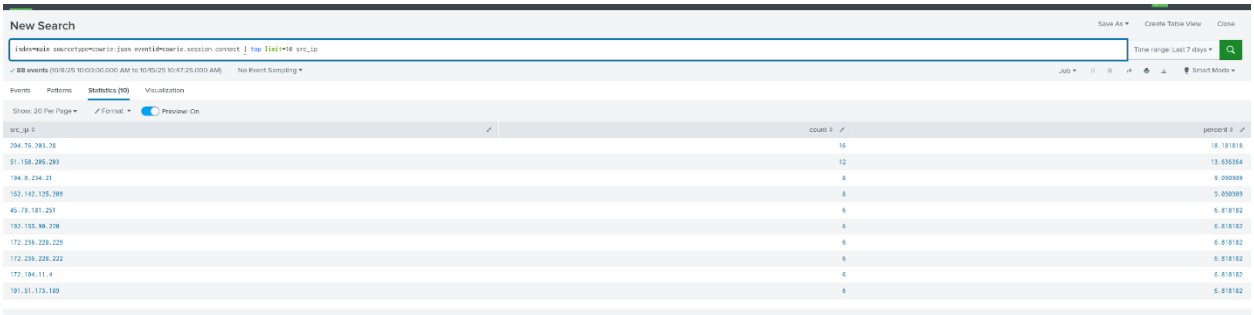
index=main sourcetype=cowrie:json (eventid="cowrie.login.failed" OR eventid="cowrie.login.success") | stats count AS "Total Login Attempts"





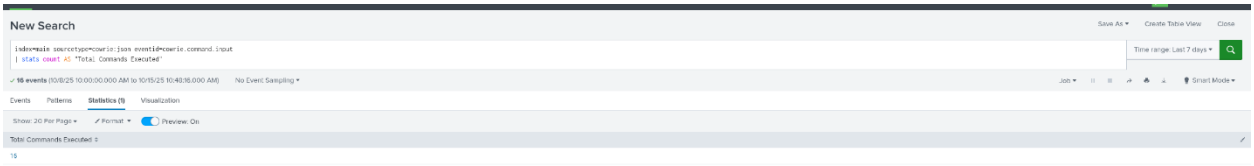
Attack by IP address:

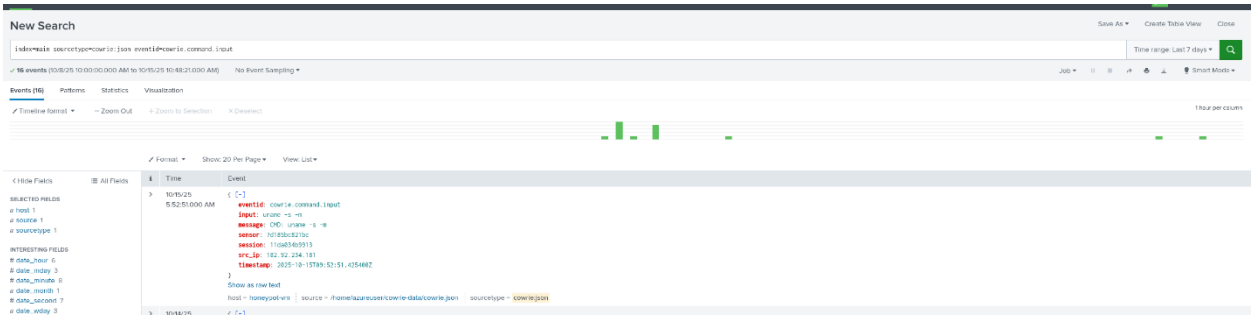
index=main sourcetype=cowrie:json eventid=cowrie.session.connect | top limit=10 src_ip



Total Commands Executed:

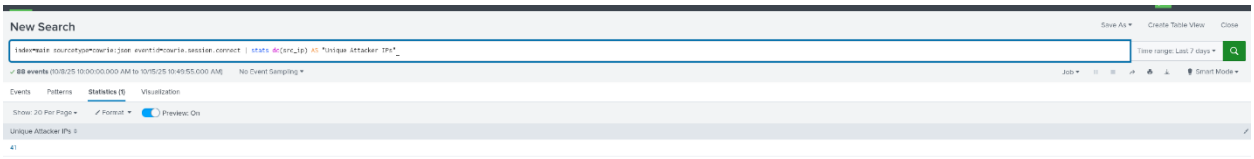
index=main sourcetype=cowrie:json eventid=cowrie.command.input | stats count AS "Total Commands Executed"





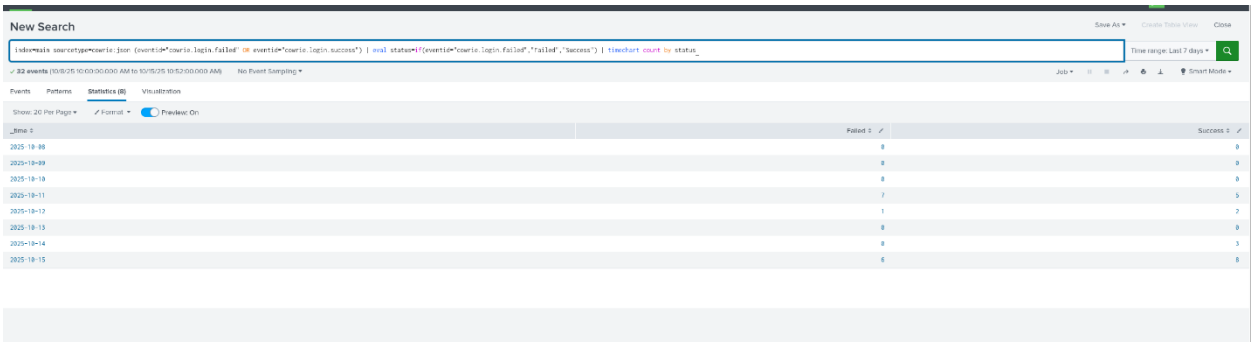
Unique Attacker IPs:

index=main sourcetype=cowrie:json eventid=cowrie.session.connect | stats dc(src_ip) AS "Unique Attacker IPs"



Failed vs Successful Logins Over Time:

index=main sourcetype=cowrie:json (eventid="cowrie.login.failed" OR eventid="cowrie.login.success") | eval status=if(eventid="cowrie.login.failed","Failed","Success") | timechart count by status



Findings & Observations

During the observation window, the honeypot recorded continuous automated attacks dominated by credential stuffing and brute-force attempts. Common usernames included 'root', 'admin', and 'test'. Attack sources clustered around known cloud provider ranges. Command input analysis showed reconnaissance commands like 'uname -a', 'ls', and 'cat /proc/cpuinfo'. Average session duration remained short (<60s), consistent with automated scanners.

Security Considerations & Hardening

- Isolate the honeypot from production networks; use separate VNet/subnet/resource group.
- Restrict outbound traffic where possible; deny egress to internal/private ranges.
- Rotate VM credentials and disable real SSH service on port 22 when Cowrie is in use.
- Limit data retention and ensure no real secrets exist on the VM.
- Monitor UF/Splunk connectivity; alert on gaps in data delivery.

Automation: Splunk Health Check Cron

*/10 * * * * root /opt/splunkforwarder/bin/splunk status || /opt/splunkforwarder/bin/splunk start

Appendix A: Example Dashboard JSON (excerpt)

SourceURL:file:///home/brownskull/Downloads/Cowrie_Honeypot_Splunk_Report_Omar_Alfayyadh_Updated.docx

```
{
  "description": "Cowrie Dashboard",
  "label": "Cowrie Overview",
  "visualizations": [
   {"title": "Total Login Attempts", "type": "singlevalue", "search": "index=main sourcetype=cowrie:json (eventid=\"cowrie.login.failed\" OR eventid=\"cowrie.login.success\") | stats count AS \"Total Login Attempts\""},
   {"title": "Attack by IP Address", "type": "table", "search": "index=main sourcetype=cowrie:json eventid=cowrie.session.connect | top limit=10 src_ip"},
   {"title": "Total Commands Executed", "type": "singlevalue", "search": "index=main sourcetype=cowrie:json eventid=cowrie.command.input | stats count AS \"Total Commands Executed\""},
   {"title": "Unique Attacker IPs", "type": "singlevalue", "search": "index=main sourcetype=cowrie:json eventid=cowrie.session.connect | stats dc(src_ip) AS \"Unique Attacker IPs\""},
   {"title": "Failed vs Successful Logins Over Time", "type": "timechart", "search": "index=main sourcetype=cowrie:json (eventid=\"cowrie.login.failed\" OR eventid=\"cowrie.login.success\") | eval status=if(eventid=\"cowrie.login.failed\",\"Failed\",\"Success\") | timechart count by status"}
  ]
}
```

Appendix B: Sample NSG Rules

Priority	Name	Port	Protocol	Direction	Action	Source	Destination
1000	allow-ssh	22	TCP	Inbound	Allow	Internet	VM Public IP
1001	allow-cowrie1	2222	TCP	Inbound	Allow	Internet	VM Public IP
1002	allow-cowrie2	2223	TCP	Inbound	Allow	Internet	VM Public IP
2000	default-deny	*	*	Inbound	Deny	Internet	VM Public IP