Casebook — SIEM & SOC Case Studies

Hands-on lab with Splunk, Linux and Threat Intel

Carlos Alexandre Menezes · Junior Cybersecurity Analyst

Overview

This Casebook collects the main practical cases developed in my home lab: a small SOC built with Splunk, Suricata, UFW and public threat intelligence feeds. The goal is not only to show dashboards, but to explain the architecture decisions and the SPL behind each detection.

All scenarios were built in a real environment (my own Linux server at home), using real SSH brute-force attempts from the Internet, HTTP traffic, MySQL logs and internal Splunk events.

Case 1 — SSH Brute-force Detection + Auto-ban (UFW)

Goal: detect IPs that fail SSH authentication multiple times in a short window and automatically block them in UFW through a Splunk alert.

Detection SPL

```
index=* sourcetype=linux_secure "sshd" (Failed OR failure OR "Invalid user")
| rex field=_raw "from (?<src_ip>\d{1,3}(?:\.\d{1,3}){3})"
| where isnotnull(src_ip)
| streamstats time_window=1m count as consecutive_failures by src_ip
| where consecutive_failures >= 5
| fields src_ip
| dedup src_ip
| outputlookup ips_a_banir.csv
```

This search runs as a **scheduled alert** every minute. The result (list of IPs) is written into a lookup file (`ips_a_banir.csv`), which works as a “vault” of IPs to be banned.

Ban script (ban_ip.sh)

```
#!/usr/bin/env bash
... (same script as PT version, comments can remain in PT/EN)
```

With this, the home SOC **reacts automatically** to SSH brute-force attacks, logs history and allows later auditing directly from Splunk.

Case 2 — Threat Intel: Overlapping IPs

A Threat Intelligence add-on was integrated to compare observed IPs with public malicious IP lists. The *"IPs Matching Threat Intel"* panel highlights IPs that appear both in our logs and in known C&C, botnet or scanner feeds.

Correlation SPL

```
| multisearch
    [ search index=* sourcetype=ban_script
      | rex max_match=1 "(?<banned_ip>\d{1,3}(?:\.\d{1,3}){3})"
      | eval via="ban_script" ]
    [ search index=* sourcetype=syslog BAN_SCRIPT
      | rex max_match=1 "ip=(?<banned_ip>\d{1,3}(?:\.\d{1,3}){3})"
      | eval via="syslog" ]
| where isnotnull(banned_ip)
| stats latest(_time) as when values(via) as via by banned_ip
| iplocation banned_ip
| lookup threatintel_by_ip banned_ip as ip OUTPUTNEW ti_desc ti_feed threat_key
| eval score=30 + if(isnotnull(ti_feed),50,0) + if(coalesce(Country,"")!="PT",10,0)
| where score>=70
| convert ctime(when) as when
| table when banned_ip Country City ti_feed ti_desc threat_key via score
```

This combines **reactive defense** (UFW bans) with **threat intelligence**, focusing on IPs that are:

- seen attacking our server; and
- already known to be malicious by external feeds.

Case 3 — "Advanced Center" Dashboard & Auditing

The "Advanced Center" dashboard centralizes Splunk health metrics, MySQL error levels, event volume and banned IPs. It also includes **quick audit** buttons (last 24h) and **15/30-day audits** that open specific SPL searches in new tabs.

This Casebook shows how the project grew from a simple SSH brute-force panel into a mini-SOC with:

- detection + automatic response;
- threat reputation matching;
- geoIP and risk scoring;
- audit panels and historical views.

For further technical details, see also the SOC Playbook and Portfolio Book.