Casebook — SIEM & SOC Case Studies

Casos práticos com Splunk, Linux e Threat Intel

Carlos Alexandre Menezes · Junior Cybersecurity Analyst

Visão Geral

Este Casebook reúne os principais casos práticos desenvolvidos no meu laboratório pessoal: configuração de um SOC caseiro com Splunk, Suricata, UFW e integração com listas de ameaça (Threat Intelligence). O objetivo é mostrar não apenas os dashboards, mas o raciocínio por trás de cada decisão de arquitetura e cada SPL.

Todos os cenários foram construídos em ambiente real (meu servidor Linux em casa), com ataques de força bruta SSH reais vindos da Internet, tráfego HTTP, logs MySQL e eventos internos do próprio Splunk.

Caso 1 — Detecção de Força Bruta SSH + Auto-ban via UFW

Objetivo: detectar IPs que falham autenticação SSH várias vezes num curto intervalo e bloqueá-los automaticamente no firewall (UFW) através de um alerta do Splunk.

Pesquisa SPL de deteção

```
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
```

Esta pesquisa é usada num **alerta agendado** a cada minuto. O resultado (lista de IPs) é escrito para um ficheiro de lookup (`ips_a_banir.csv`), que funciona como uma "caixa-forte" de IPs a serem banidos.

Script de banimento (ban_ip.sh)

```
#!/usr/bin/env bash
set -euo pipefail
IFS=$'\n\t'

DEBUG_LOG="/opt/splunk/var/log/splunk/ban_script.log"
LOOKUP_FILE="/opt/splunk/etc/apps/search/lookups/ips_a_banir.csv"
WHITELIST="/opt/splunk/etc/apps/search/lookups/ip_whitelist.csv"
HISTORY="/opt/splunk/etc/apps/search/lookups/banned_history.csv"
UFW="/usr/sbin/ufw"
LOGGER="/usr/bin/logger"

exec 2>>"$DEBUG_LOG"

echo "--------------------------------------------------" >> "$DEBUG_LOG"
```

```
echo "$(date): Início ban_ip.sh. Lendo: $LOOKUP_FILE" >> "$DEBUG_LOG"

mapfile -t WL < <(awk -F',' 'NR>1{print $1}' "$WHITELIST" | tr -d '\r"')

awk -F',' 'NR>1{print $1}' "$LOOKUP_FILE" | tr -d '\r"' | while read -r IP; do
  IP="$(echo "$IP" | xargs)"
  [[ -z "${IP:-}" ]] && continue
  [[ ! "$IP" =~ ^([0-9]{1,3}\.){3}[0-9]{1,3}$ ]] && { echo "$(date): Ignorado (IPv4
inválido): '$IP'" >> "$DEBUG_LOG"; continue; }

  for W in "${WL[@]:-}"; do
    if [[ "$IP" == "$W" ]]; then
      echo "$(date): Ignorado (whitelist): $IP" >> "$DEBUG_LOG"
      continue 2
    fi
  done

  if /usr/bin/sudo $UFW status numbered | grep -qE "DENY IN +$IP"; then
    echo "$(date): Já está banido: $IP" >> "$DEBUG_LOG"
    continue
  fi

  echo "$(date): A banir $IP via ufw..." >> "$DEBUG_LOG"
  /usr/bin/sudo $UFW insert 1 deny from "$IP" to any comment 'Splunk-Auto-Ban-SSH-Brute-
Force' || {
    echo "$(date): Falha ao banir $IP" >> "$DEBUG_LOG"
    continue
  }

  $LOGGER -t BAN_SCRIPT "action=ip_banned ip=$IP reason=ssh_bruteforce"
  echo "$(date): $IP banido." >> "$DEBUG_LOG"
  echo "$(date +%F\ %T),$IP,ban,ban_script" >> "$HISTORY"
done

printf 'src_ip\n' > "$LOOKUP_FILE"
chown splunk:splunk "$LOOKUP_FILE" "$HISTORY"
echo "$(date): Fim ban_ip.sh. Lookup limpo." >> "$DEBUG_LOG"
echo "--------------------------------------------------" >> "$DEBUG_LOG"
exit 0
```

Com isto, o SOC caseiro passa a **reagir automaticamente** a ataques de força bruta SSH, registando histórico e permitindo auditoria posterior via Splunk.

Caso 2 — Threat Intel: IPs Coincidentes com Feeds Maliciosos

Integrámos um Add-on de Threat Intelligence para comparar IPs observados nos logs com listas de IPs maliciosos públicos. O painel *"IPs Coincidentes com Threat Intel"* mostra quando um IP visto na nossa infraestrutura também aparece em feeds de C&C, botnets ou scanners conhecidos.

SPL de correlação

```
| 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
```

Aqui combinamos **defesa reativa** (banimentos via UFW) com **inteligência de ameaças**, destacando os IPs que são, ao mesmo tempo:

- observados a atacar o nosso servidor;
- e conhecidos em listas de ameaças (feeds externos).

Caso 3 — Painel "Centro Avançado" & Auditoria

O dashboard "Centro Avançado" centraliza métricas de saúde do Splunk, MySQL, volume de eventos e status de IPs banidos. Inclui também botões de **auditoria rápida** (últimas 24h) e auditorias de **15 e 30 dias**, que abrem pesquisas SPL dedicadas em novas abas.

Este Casebook mostra como o projeto evoluiu de um simples painel de brute force SSH para um mini-SOC com:

- detecção + reação automática;
- ameaças com reputação externa;
- geoIP e score de risco;
- painéis de auditoria e histórico.

Para mais detalhes técnicos, consulte também o SOC Playbook e o Portfolio Book.