


Trabalho dia 24/06


1. Ambiente preparado

- Máquina virtual criada (Ubuntu Server)
- Sistema atualizado
- Conectividade testada (ex: ping entre máquinas)

2. Snort instalado e configurado

- Versão: Snort 2.9.20
- snort.conf ajustado corretamente
- Criado e usado o ficheiro de regras personalizadas: local.rules
- Snort a correr com sucesso (ex: snort -A console -q -c /etc/snort/snort.conf -i <interface>)
- Teste feito com ping → Alerta detetado com sucesso 

3. Elasticsearch instalado e operacional

- Corrigido erro inicial relacionado com Java 21. A versão do snort que esta a ser utilizada não trabalhava com Java 21, então foi necessário mudar para Java 11
- Instalado e ativado Java 11
- Heap ajustado para máquinas com pouca RAM (512m)
- Elasticsearch iniciado e verificado com:
 - curl -X GET "localhost:9200/"
- Elasticsearch a funcionar sem erros 

Trabalho dia 25/06

1. Configuração de Rede na Máquina Virtual

Para permitir que o Snort monitorize tráfego externo (ex: pings de uma máquina Windows host), foi necessário configurar dois adaptadores de rede na máquina virtual:

Adaptador 1: NAT

- Usado para dar acesso à internet na VM (atualizações, pacotes, etc.)
- Permite que a VM tenha saída para a internet, mas não é útil para tráfego local do host (ex: ping da máquina real)

Adaptador 2: Host-only Adapter

- Configurado Host-only Adapter
- Permite que o host Windows comunique com a VM
- Essencial para simular ataques e tráfego real do host para a VM

 Teste feito:

Depois de configurar os dois adaptadores, foi possível executar:

```
ping <IP_da_VM>
```

...a partir do Windows, e o Snort passou a capturar esse tráfego ICMP, gerando alertas no ficheiro snort.alert.fast. Foi também testado a captura do tráfego ICMP pelo tcpdump (sudo tcpdump -i).

2. Configuração do Snort

Snort configurado para gerar alertas no ficheiro /var/log/snort/snort.alert.fast.

Foi necessário fazer a verificação de que existia de facto o diretório /var/log/snort/ (caso não existisse teríamos que criar pelo comando sudo mkdir -p /var/log/snort) e também verificar se snort tinha permissões para escrever no diretório (sudo chown snort:snort /var/log/snort).

3. Configuração do Logstash

Criado ficheiro `/etc/logstash/conf.d/snort.conf`

Foi reiniciado o Logstash para aplicar configuração.

4. Validação do Logstash e Elasticsearch

Foi verificado que o Logstash está a processar logs e a enviar para Elasticsearch.

Foi usado o comando `curl -X GET "localhost:9200/_cat/indices?v"`.

Este comando permitiu listar todos os ativos no Elasticsearch e confirmou que o índice `snort-alerts-*` existia e que estava a receber documentos (`docs.count > 0`).

5. Configuração do Logstash

Foi criado Data View (Index Pattern) `snort-alerts-*` no Kibana.

Confirmada a visualização dos logs no Discover do Kibana.

Trabalho dia 28/06

1. Simulação de Ataques Reais com Kali Linux

- Foi instalado o Kali Linux no Windows via WSL, permitindo realizar ataques internos controlados.
- Foi utilizada a ferramenta Hydra para realizar ataques SSH brute force a partir do Kali para a VM Ubuntu.

2. Criação e Implementação de Regras Snort Personalizadas

- Foi criada regras específicas no `local.rules`, como:

Deteção de brute force SSH:

```
alert tcp any any -> <IP_VM> 22 (msg:"Possible SSH brute force attack"; flags:S; threshold:type threshold, track by_src, count 5, seconds 60; sid:1000002; rev:1;)
```

- Foi identificado e resolvido conflitos de `SID` duplicado no ficheiro de regras.

3. Correção e Teste de Permissões no Logstash

- O Logstash estava a falhar por erro de permissões no ficheiro de alertas.
- Foi corrigido com sucesso os acessos e o Logstash começou a ingerir os logs corretamente.