LABORATÓRIO PESSOAL – 2
Análise de tráfego utilizando Wireshark e Zeek

Autor: Lucas Vieira Areal

Data: 07/01/2026

Repositório: https://github.com/sacullaera/Labs

Este laboratório prático foi desenvolvido para simular e analisar atividades de reconhecimento de rede, uma das primeiras etapas de um ataque cibernético. Utilizando a combinação de Wireshark para inspeção de pacotes em baixo nível e Zeek para geração de logs estruturados, o projeto demonstra as técnicas e ferramentas fundamentais que um Analista de SOC utiliza para detectar e investigar ameaças em redes corporativas.

As motivações do projeto foram:

- Construção de um laboratório de segurança local e isolado;
- Capturar tráfego de rede gerado por atividades benignas e maliciosas simuladas;
- Analisar pacotes de rede em baixo nível com Wireshark;
- Gerar e interpretar logs estruturados com Zeek (Bro);
- Comparar as abordagens de análise em tempo real (Wireshark) e pós-incidente (Zeek);
- Desenvolver habilidades fundamentais para atuação em equipes de SOC.

A ambientação do laboratório se dá conforme apresentado no quadro abaixo:

COMPONENTE	DETALHE
Máquina Host	**Windows 11**
Hypervisor	**VMware Workstation Pro**
VM Atacante	**Kali Linux 2025.3**
VM Vítima	**Metasploitable2**
Rede	**Modo “Host-only” no VMware**
Ferramentas	**Wireshark (Windows), Zeek + Nmap (Kali)**

Tabela 1: Componentes do laboratório

Metodologia

Criação do Ambiente

Foi configurado um ambiente de laboratório isolado utilizando o VMware Workstation Pro. Foram criadas duas Máquinas Virtuais (VMs): uma com Kali Linux 2025.3 atuando como 'Atacante' e outra com o sistema Metasploitable2 atuando como 'Vítima'. A rede foi configurada no modo 'Host-only', garantindo que todo o tráfego gerado permanecesse confinado ao ambiente local, sem risco de extravasamento para a rede residencial.

Captura com Wireshark

- Interface monitorada: VMware network Adapter VMnet

- Atividades simuladas:

  - ping do Kali para Metasploitable2 (tráfego ICMP)

  - nmap -sS -p 21,22,23,80 do Kali para Metasploitable2 (scan TCP SYN)

Primeiro vamos identificar as redes de cada máquina – alvo e atacante – com a finalidade de registrarmos e facilitar nosso ataque com uso do nmap. O uso do comando para visualização das redes é o *ifconfig* que irá mostrar as redes das máquinas, identificando dessa forma a atacante com endereço **192.168.182.133/24** e a alvo **192.168.182.129/24**. Após a identificação das redes, iremos agora executar o zeek para capturar o tráfego assim que como fizemos com o wireshark com o comando *sudo /opt/zeek/bin/zeek -i eth0*. Iremos abrir outra aba no terminal de comando na máquina atacante e iremos pingar a alvo com o comando *ping 192.168.182.129*.

Após sete pings eu encerrei o comando e encerrei a análise de tráfego do wireshark na máquina hospedeira para analisar o tráfego gerado, porém devido a alta quantidade de tráfego gerado devemos filtrar esse tráfego de forma que facilite nossa visualização, então irei filtrar pelo protocolo ICMP que é o protocolo de camada 3 (camada de rede) utilizado pelos dispositivos de rede para comunicação de problemas com a transmissão dos dados. Selecionei um ping *request* que para visualizar as informações do envio dos dados e outro *reply* para visualizar a resposta do alvo. Após analisarmos vamos agora realizar um segundo teste e visualizar o tráfego coletado com o uso do comando *sudo nmap -sS -p 21,22,23,80 192.168.182.129* para visualizar de formar rápida as portas FTP utilizada para transferência de arquivos sem criptografia, SSH para acesso remoto criptografado, Telnet para acesso remoto com plaintext e HTTP para transferência de dados pela internet sem segurança.

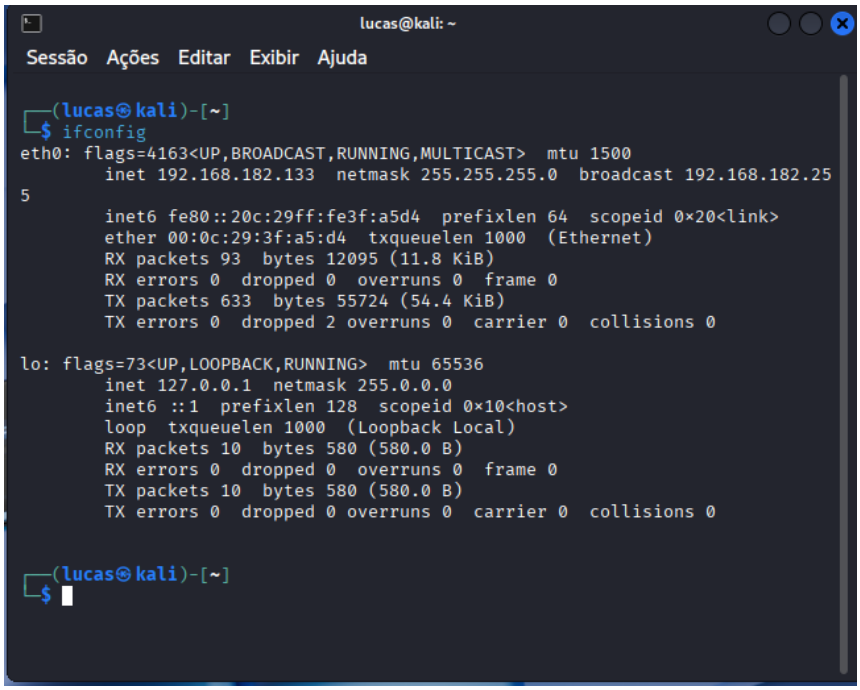

Figura 1 - identificação da rede atacante

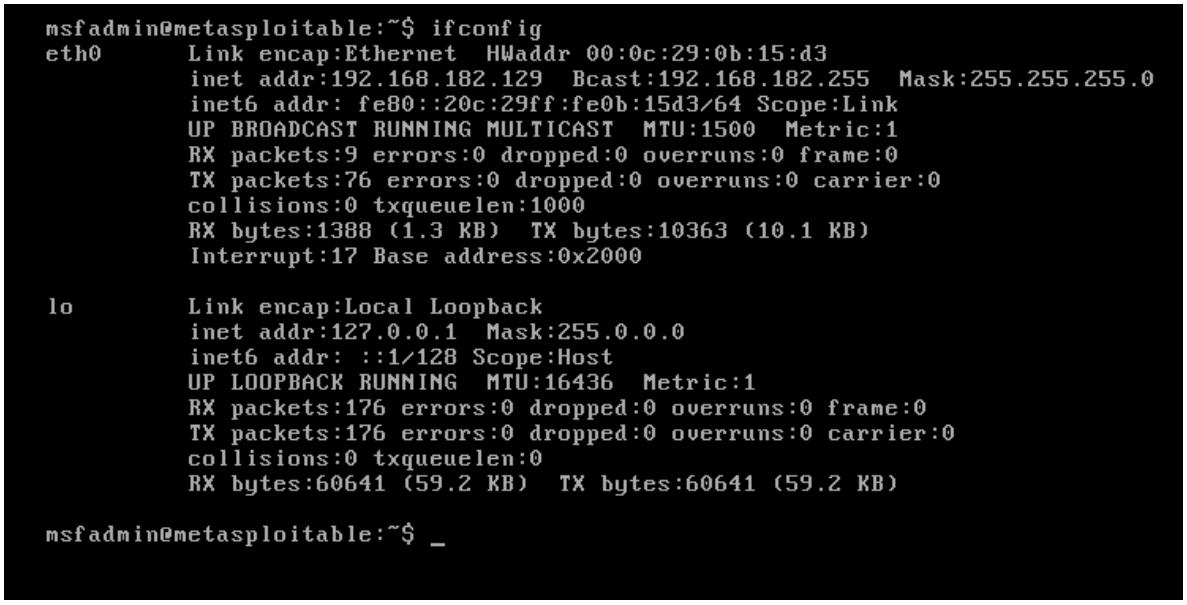

Figura 2 - identificação da rede alvo

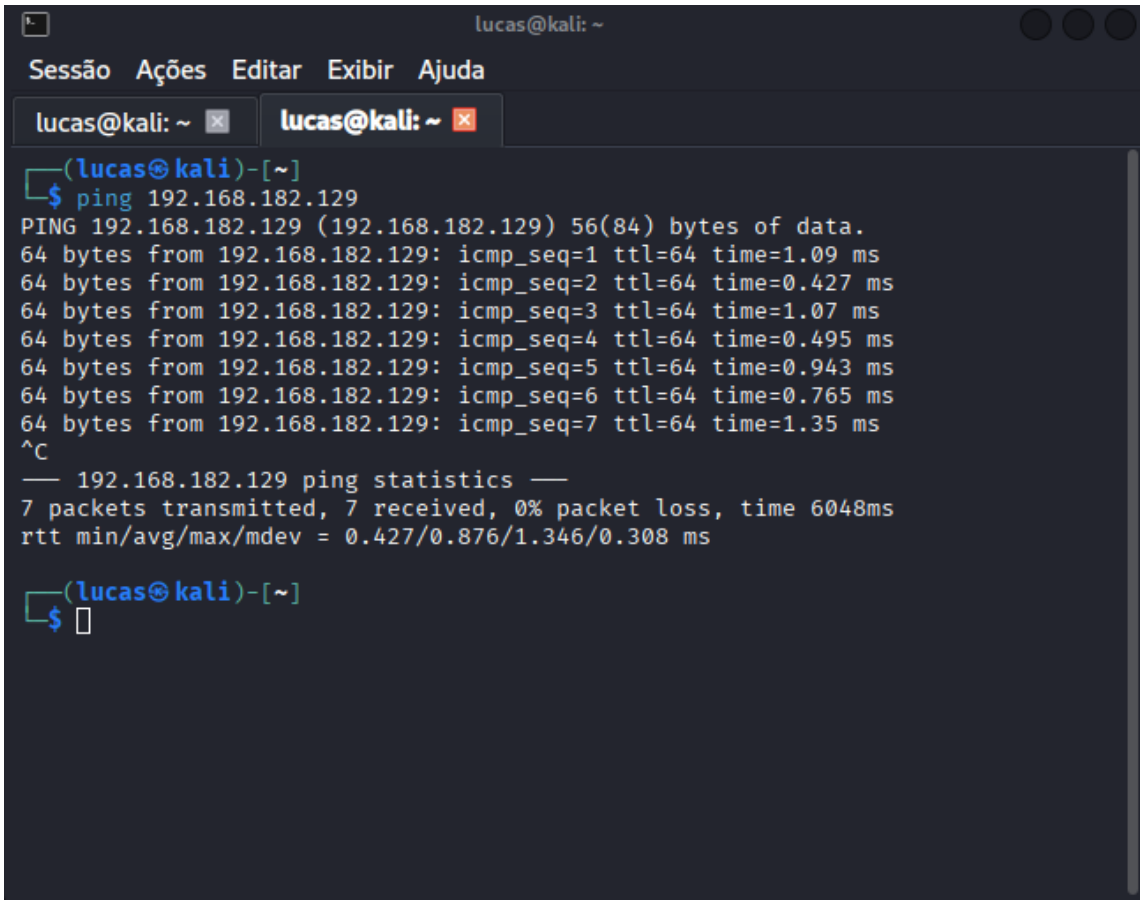

Figura 3 - comando de ping na máquina alvo

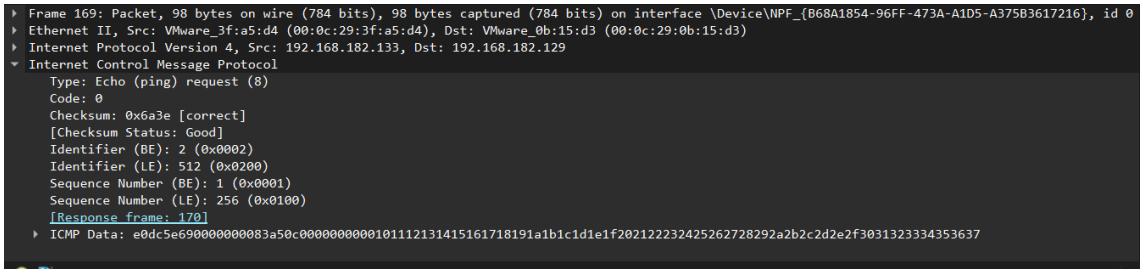

Figura 4 - demonstração do tráfego ICMP capturado

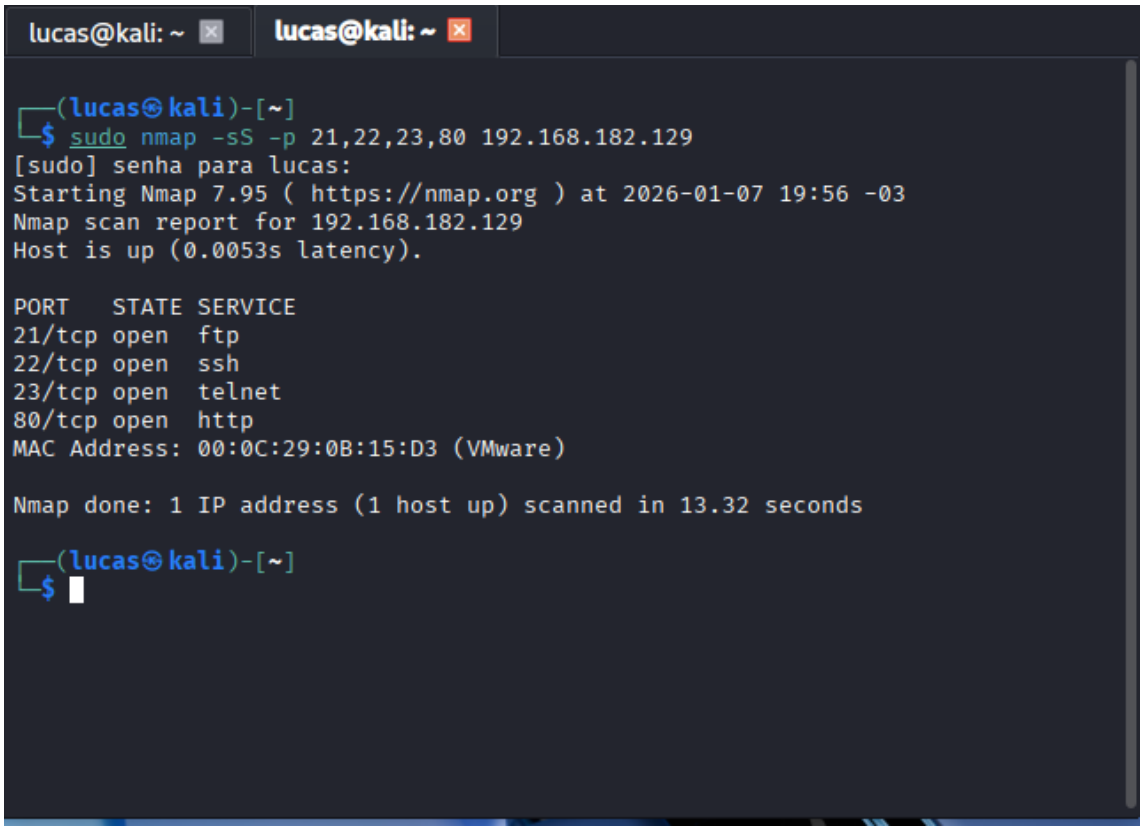

Figura 5 - nmap para rápida visualização das portas especificadas

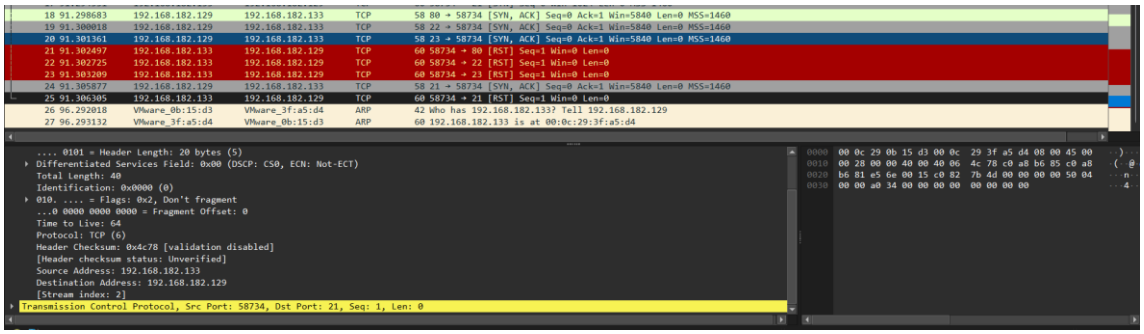

Figura 6 - captura do wireshark

Geração de logs com Zeek (Kali Linux)

- Comando usado: sudo zeek -i eth0 local

- Mesmas atividades simuladas durante a execução do Zeek.

- Logs gerados: conn.log, packet_filter.log

Após coletar o tráfego pelo wireshark vou agora coletar o tráfego coletado pelo zeek na máquina alvo através do arquivo “conn.log” o qual é o documento gerado após encerrar a coleta de tráfego. Vamos agora fazer o comparativo das duas coletas (wireshark e zeek) e após uma breve análise podemos visualizar o tráfego do ping na última linha que informa o endereço de onde veio a solicitação (máquina atacante),

para onde foi enviado (máquina alvo), a contagem, além das portas de envio dos pings de ambas as máquinas.

Vamos agora visualizar o ataque de sniff de portas e serviços com uso do nmap com o uso do comando *sudo nmap -sS -p 21,22,23,80 192.168.182.129* e aguardar 20s para finalizar o ataque.

Análise dos resultados

Análise do Tráfego ICMP (Ping)

Wireshark

Os pacotes ICMP Echo Request (type=8) e Echo Reply (type=0) são visíveis com timestamps precisos. Cada ping gera um par de pacotes e podemos também identificar o endereço de origem e destino além das portas de origem e destino.

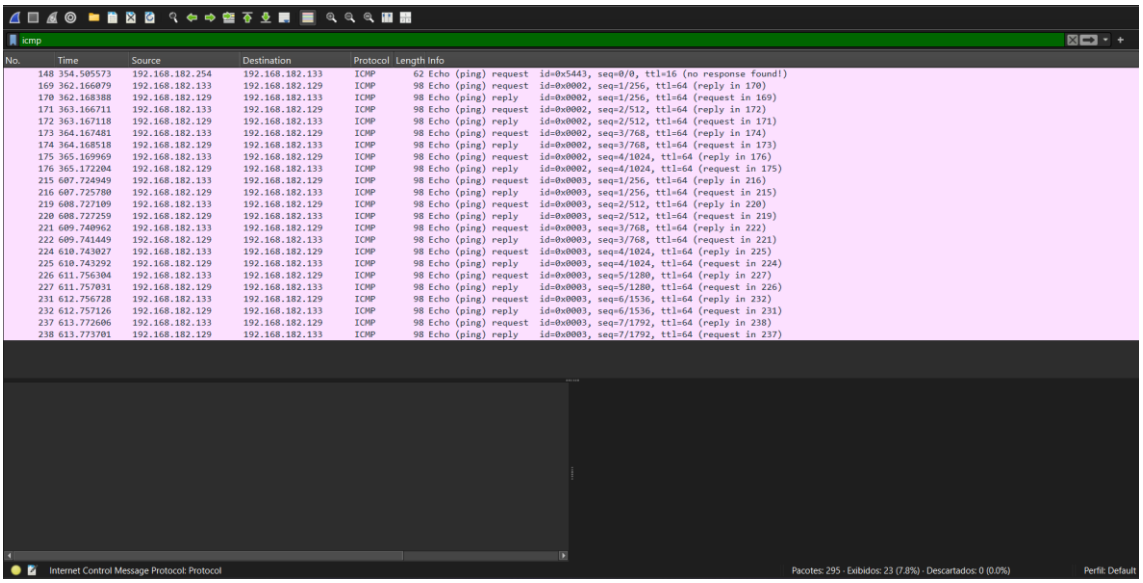

Figura 7 - captura de tráfego filtrado por ICMP pelo wireshark

Zeek

O Zeek não registra o tráfego ICMP no conn.log por padrão, pois este log foca em conexões com estado (TCP/UDP). Isso mostra uma limitação importante a ser conhecida por um analista: nem todo tráfego é visível nos logs principais.

Análise do Scan SYN com Nmap

Wireshark

É possível notar que o wireshark capturou conexões com serviços distintos, sendo estes FTP, telnet, SSH e HTTP, onde todos possuem características em comum:

- Tempo de duração extremamente breve;
- Não finalizou o processo do three-way-handshake com flag RST enviada pelo atacante;
- Durante a comunicação não houve transmissão de dados.

Com essas informações podemos ter uma forte comprovação de que se trata de um tipo de ataque de análise de portas para identificação de serviços, portas e protocolos vulneráveis abertos de forma *stealth* (escondida). Apesar de visualizarmos ao final do tráfego analisado as flags RST, é importante notar que houve a tentativa de comunicação inicial (SYN, SYN/ACK), o que é outro indício forte de ataque de mapeamento.

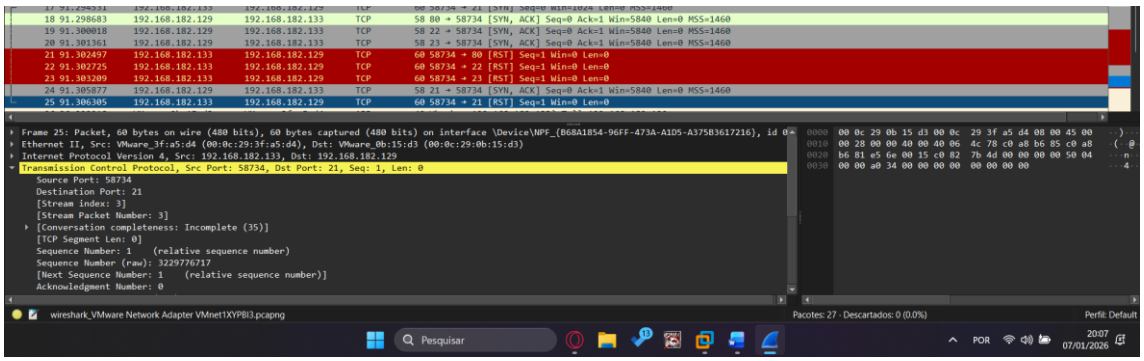

Figura 8 - captura do tráfego nmap - FTP

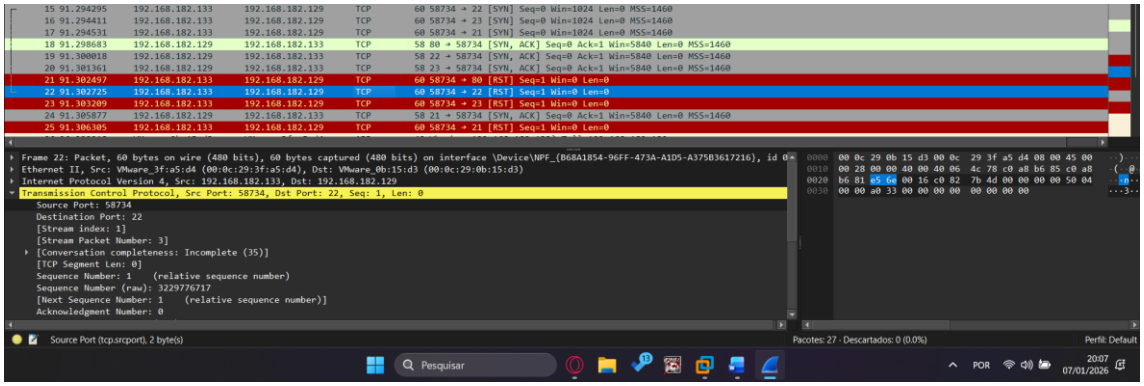

Figura 9 - captura do tráfego nmap - SSH

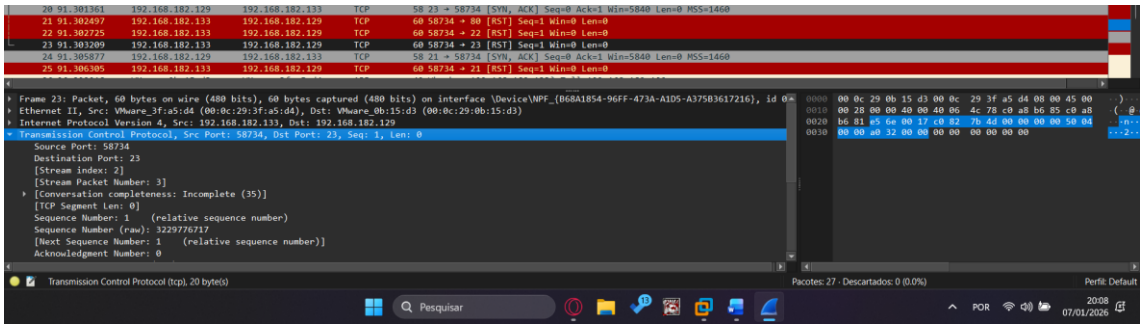

Figura 10 - captura do tráfego wireshark – telnet

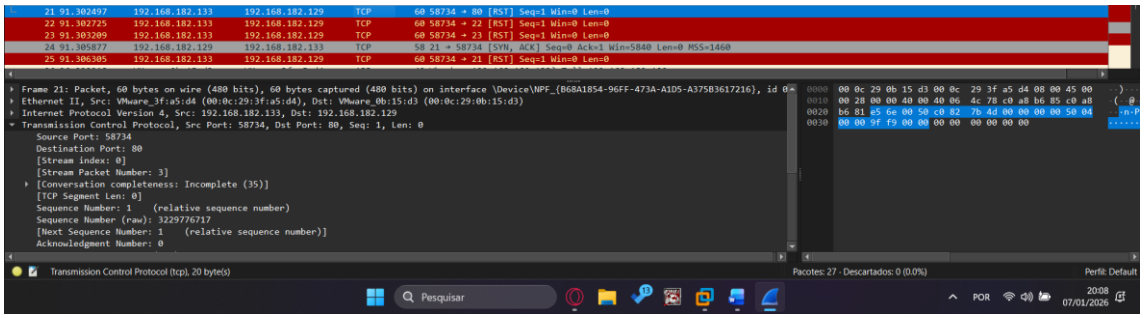

Figura 11 - captura do tráfego wireshark – HTTP

Zeek (conn.log)

Durante a execução do comando nmap -sS -p 21,22,23,80 192.168.182.129, o Zeek registrou quatro conexões distintas no arquivo conn.log. Todas apresentam as seguintes características em comum:

conn_state = "RSTO": Indica que a conexão foi iniciada pelo atacante, mas foi este mesmo quem a encerrou com um pacote RST de forma abrupta, sem finalizar o handshake.

duration < 0.012 segundos: A conexão foi extremamente breve.

orig_bytes = 0 e resp_bytes = 0: Nenhum dado da aplicação foi trocado.

O estado RSTO, especialmente quando associado a uma duração de conexão extremamente curta (< 0.012s) e ausência de troca de dados (orig_bytes = 0), é uma assinatura de um scan de portas do tipo SYN. Isso permite que equipes de SOC criem regras de detecção precisas para identificar reconhecimento de rede malicioso, mesmo quando as portas-alvo estão abertas.

Figura 12 - tráfego ping capturado pelo zeek

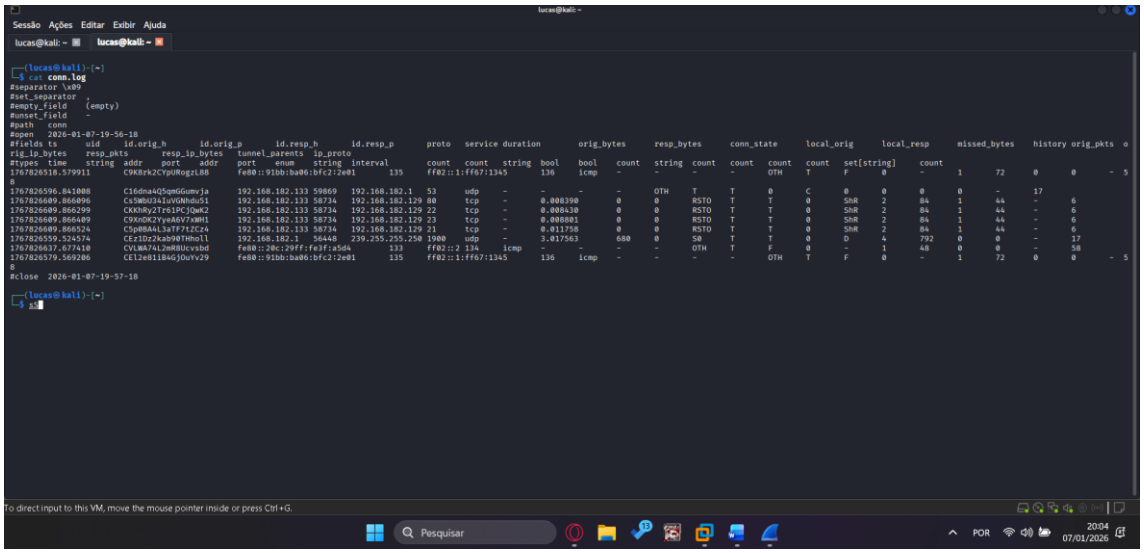

Figura 13 - tráfego nmap capturado pelo zeek

Lições Aprendidas e Conclusão

Este laboratório foi fundamental para consolidar a compreensão de como diferentes ferramentas de segurança fornecem perspectivas complementares sobre o mesmo evento de rede. O Wireshark permite a inspeção forense do tráfego bruto, enquanto o Zeek agrega e estrutura essas informações, tornando-as prontas para análise em larga escala. A habilidade de correlacionar essas duas fontes de dados é essencial para um Analista de SOC, pois permite validar alertas, reduzir falsos positivos e compreender a tática do atacante com precisão.

Próximos passos

Como evolução deste projeto, pretendo integrar os logs do Zeek a um stack de visualização (como o Grafana ou Kibana) para criar dashboards de monitoramento em tempo real. Além disso, explorarei a escrita de scripts personalizados no Zeek para gerar alertas automáticos sempre que for detectado o padrão de conexão RSTO associado a múltiplas portas em um curto intervalo de tempo, simulando uma regra de detecção real de um SOC.

Referências

- Zeek Official Site. Disponível em: https://zeek.org

- Nmap Reference Guide. Disponível em: https://nmap.org/book/man.html

- Wireshark User’s Guide. Disponível em: https://www.wireshark.org/docs/wsug_html_chunked/