

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 forma 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.

The screenshot shows a terminal window titled "Sessão Ações Editar Exibir Ajuda". The command \$ ifconfig is run, displaying network interface information. The output shows two interfaces: eth0 (Ethernet) and lo (Loopback). The eth0 interface has an IP of 192.168.182.133 and a broadcast address of 192.168.182.255. The lo interface has an IP of 127.0.0.1. The terminal prompt is shown again at the bottom.

```
lucas@kali: ~
Sessão Ações Editar Exibir Ajuda
[(lucas@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.182.133 netmask 255.255.255.0 broadcast 192.168.182.255
      ether 00:0c:29:3f:a5:d4 txqueuelen 1000 (Ethernet)
        RX packets 93 bytes 12095 (11.8 KiB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 633 bytes 55724 (54.4 KiB)
        TX errors 0 dropped 2 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Loopback Local)
        RX packets 10 bytes 580 (580.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 10 bytes 580 (580.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[(lucas@kali)-[~]
$
```

Figura 1 - identificação da rede atacante

The screenshot shows a terminal window titled "msfadmin@metasploitable:~\$". The command \$ ifconfig is run, displaying network interface information. The eth0 interface has an IP of 192.168.182.129 and a broadcast address of 192.168.182.255. The lo interface has an IP of 127.0.0.1. The terminal prompt is shown again at the bottom.

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:0b:15:d3
          inet addr:192.168.182.129 Bcast:192.168.182.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe0b:15d3/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:9 errors:0 dropped:0 overruns:0 frame:0
            TX packets:76 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:1388 (1.3 KB) TX bytes:10363 (10.1 KB)
            Interrupt:17 Base address:0x2000

lo      Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:176 errors:0 dropped:0 overruns:0 frame:0
            TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:60641 (59.2 KB) TX bytes:60641 (59.2 KB)

msfadmin@metasploitable:~$ _
```

Figura 2 - identificação da rede alvo

A screenshot of a terminal window titled "Sessão Ações Editar Exibir Ajuda". It shows two tabs: "lucas@kali: ~" (selected) and "lucas@kali: ~". The selected tab contains the following text:

```
$ ping 192.168.182.129
PING 192.168.182.129 (192.168.182.129) 56(84) bytes of data.
64 bytes from 192.168.182.129: icmp_seq=1 ttl=64 time=1.09 ms
64 bytes from 192.168.182.129: icmp_seq=2 ttl=64 time=0.427 ms
64 bytes from 192.168.182.129: icmp_seq=3 ttl=64 time=1.07 ms
64 bytes from 192.168.182.129: icmp_seq=4 ttl=64 time=0.495 ms
64 bytes from 192.168.182.129: icmp_seq=5 ttl=64 time=0.943 ms
64 bytes from 192.168.182.129: icmp_seq=6 ttl=64 time=0.765 ms
64 bytes from 192.168.182.129: icmp_seq=7 ttl=64 time=1.35 ms
^C
--- 192.168.182.129 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6048ms
rtt min/avg/max/mdev = 0.427/0.876/1.346/0.308 ms
```

The bottom of the terminal window shows the prompt: \$

Figura 3 - comando de ping na máquina alvo

A screenshot of a NetworkMiner tool interface. The main pane displays the following details for a captured ICMP packet:

- Frame 169: Packet, 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface \Device\NPF_{B68A1854-96FF-473A-A1D5-A375B3617216}, id 0
- Ethernet II, Src: VMWare_3f:a5:d4 (00:0c:29:3f:a5:d4), Dst: VMWare_0b:15:d3 (00:0c:29:0b:15:d3)
- Internet Protocol Version 4, Src: 192.168.182.133, Dst: 192.168.182.129
- Internet Control Message Protocol
 - Type: Echo (ping) request (8)
 - Code: 0
 - Checksum: 0x6a3e [correct]
 - [Checksum Status: Good]
 - Identifier (BE): 2 (0x0002)
 - Identifier (LE): 512 (0x0200)
 - Sequence Number (BE): 1 (0x0001)
 - Sequence Number (LE): 256 (0x0100)
 - [Response frame: 170]
- ICMP Data: e0dc5e6900000000083a50c0000000000101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637

Figura 4 - demonstração do tráfego ICMP capturado

```
lucas@kali: ~ [x] lucas@kali: ~ [x]
[~] $ sudo nmap -sS -p 21,22,23,80 192.168.182.129
[sudo] senha para lucas:
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-07 19:56 -03
Nmap scan report for 192.168.182.129
Host is up (0.0053s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
MAC Address: 00:0C:29:0B:15:D3 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.32 seconds

[~] $
```

Figura 5 - nmap para rápida visualização das portas especificadas

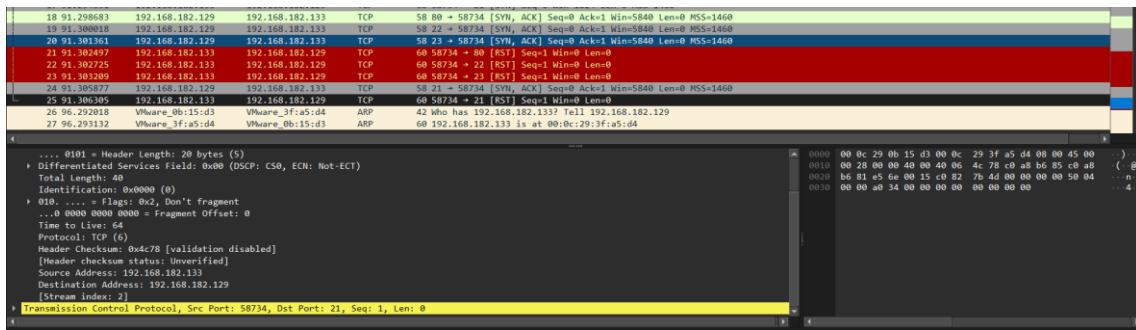


Figura 6 - captura do wireshark

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.

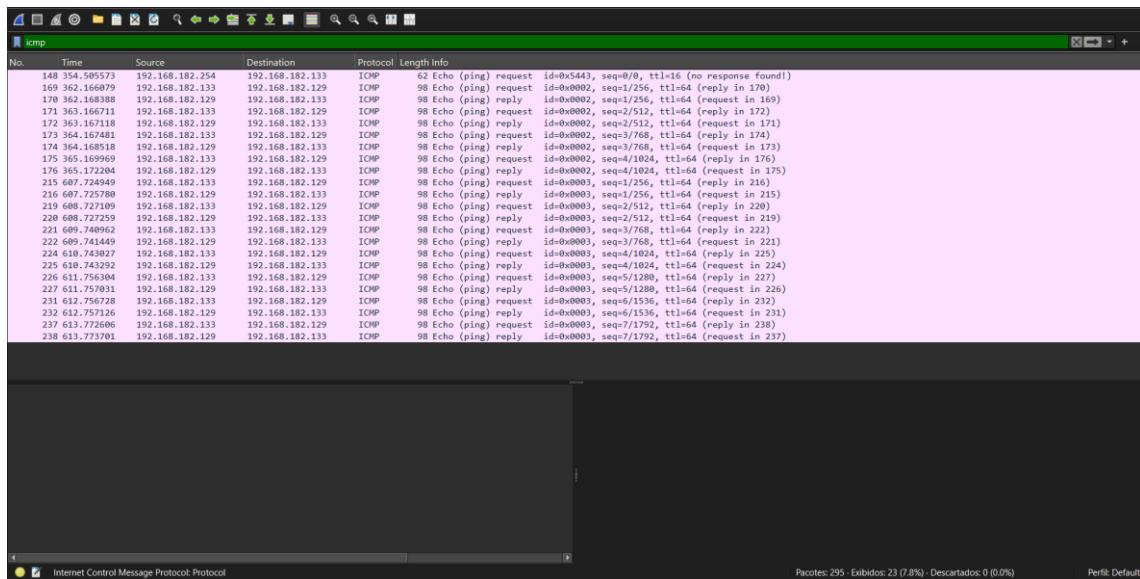


Figura 7 - captura de tráfego filtrado por ICMP pelo wireshark

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.

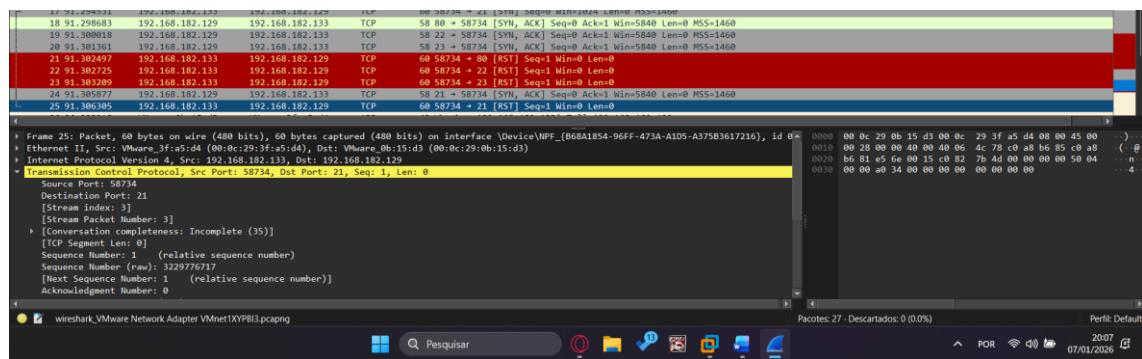


Figura 8 - captura do tráfego nmap - FTP

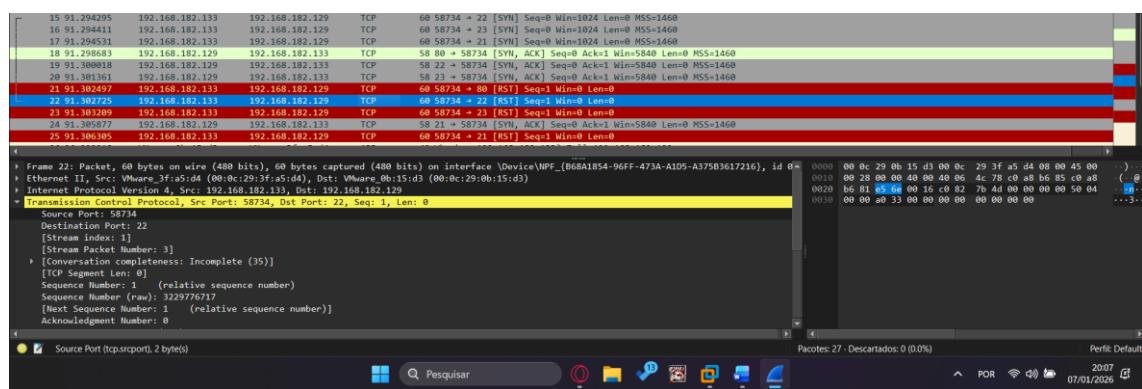


Figura 9 - captura do tráfego nmap - SSH

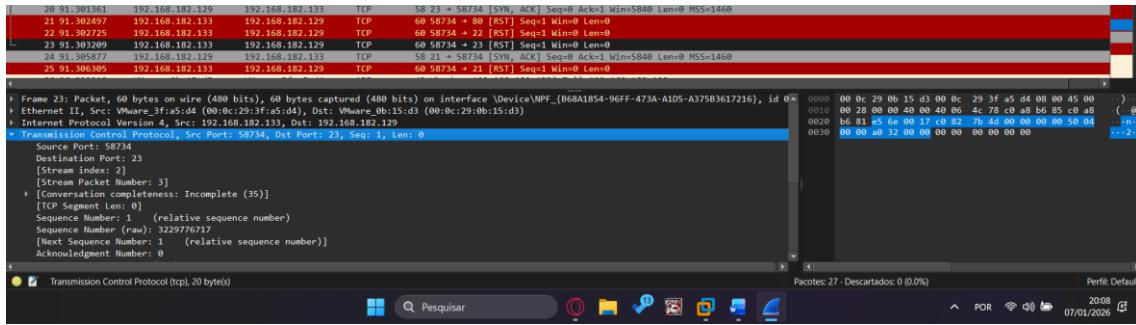


Figura 10 - captura do tráfego wireshark – telnet

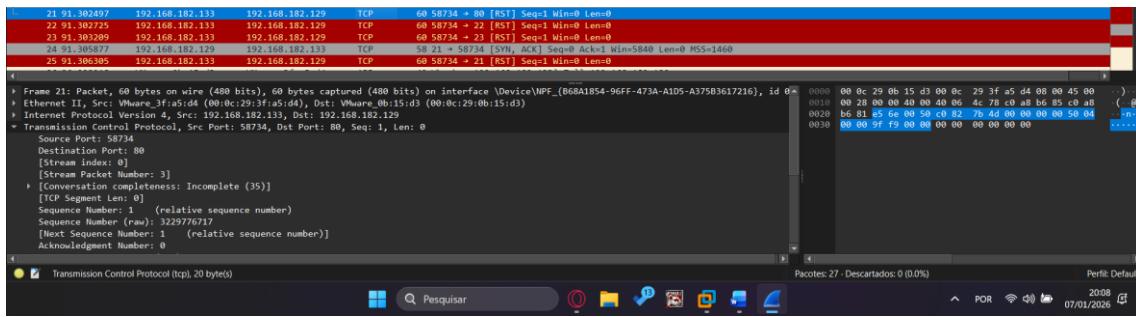


Figura 11 - captura do tráfego wireshark – HTTP

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/