# Segurança de Redes: Tópicos Essenciais

Pedroso

TE354 Redes de Computadores

11 de junho de 2025

# Sumário

# Ataques de Camada 2

- **O que é Yersinia?**
  - ✔ Ferramenta de linha de comando (e antiga interface gráfica) para testes e exploração de vulnerabilidades em protocolos de Camada 2.
  - ✔ Permite simular diferentes tipos de ataques e cenários para entender a robustez de uma rede.
- **Para que é Usado?**
  - ✔ Testes de Penetrabilidade (Pentests) em ambientes controlados.
  - ✔ Pesquisa e Desenvolvimento de Segurança de Rede.
  - ✔ Educação sobre vulnerabilidades de Camada 2.
  - ✔ Validação de configurações de segurança (ex: BPDU Guard, Root Guard).
- **Protocolos Suportados (entre outros):**
  - ✔ STP (Spanning Tree Protocol)
  - ✔ CDP (Cisco Discovery Protocol)
  - ✔ DTP (Dynamic Trunking Protocol)
  - ✔ VTP (VLAN Trunking Protocol)
  - ✔ HSRP (Hot Standby Router Protocol)
  - ✔ DHCP (Dynamic Host Control Protocol)

# AVISO DE SEGURANÇA E ÉTICA

## CUIDADO!

- Este tipo de ataque pode causar interrupções graves na rede.
- Seu uso indevido (em redes de produção ou sem autorização) é antiético, ilegal e altamente prejudicial.
- **Utilize o Yersinia EXCLUSIVAMENTE em:**
  - ✔ **Ambientes de Laboratório Isolados:** Máquinas virtuais, redes de teste dedicadas.
  - ✔ **Com Permissão Explícita:** **Não faça isso sem permissão.**
- Objetivo: Compreender vulnerabilidades e aprender a proteger as redes, não causar danos.

# Exemplo Prático: Ataque de Root Bridge STP (CLI)

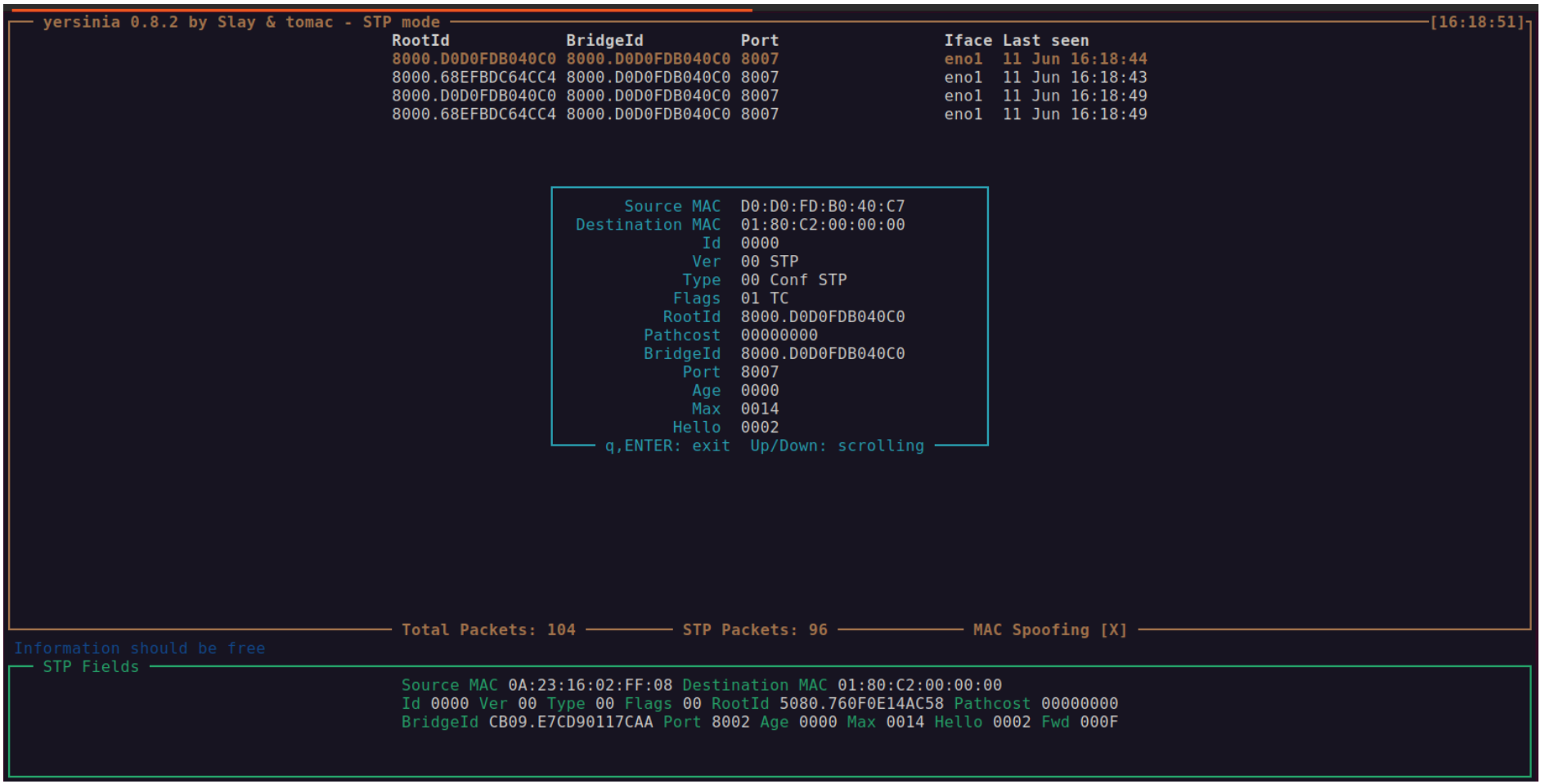
- **Objetivo do Ataque:** Fazer com que sua máquina se torne o *Root Bridge* da rede, forçando os outros switches a reconfigurarem a topologia STP.
- **Mecanismo:** Envia STP-BPDUs (Bridge Protocol Data Units) falsificados com uma prioridade de bridge muito baixa (inferior aos demais switches).
- **Comando para Iniciar o Ataque:**

```
$ sudo yersinia -I <interface> -D stp -t 1
```

  - ✔ -D stp: Indica o protocolo STP.
  - ✔ -t 1: Especifica o tipo de ataque *Root Confluence*.
- **Para Parar o Ataque:**
  - ✔ Pressione Ctrl+C no terminal onde o Yersinia está rodando.
  - ✔ A rede se reestabilizará (voltando ao Root Bridge original) após alguns segundos.

# Interface

```
$ sudo yersinia -I <interface>
```



# O que Acontece Durante o Ataque?

- **Reconfiguração da Topologia:**
  - ✔ Switches legítimos na rede recebem seus BPDUs falsificados.
  - ✔ Eles acreditam que sua máquina é o novo Root Bridge (devido à prioridade mais baixa).
  - ✔ Recalculam a árvore de abrangência e ajustam o estado de suas portas.
- **Impacto na Rede:**
  - ✔ Pode haver uma **breve interrupção** no tráfego enquanto os switches convergem para a nova topologia.
  - ✔ Dependendo da configuração e das proteções STP (ex: BPDU Guard, Root Guard), o ataque pode ser **bloqueado** ou causar o **desligamento da porta** afetada no switch.
  - ✔ Em cenários sem proteção, o tráfego que passaria pelo Root Bridge original pode ser **redirecionado através da sua máquina**, permitindo interceptação (Man-in-the-Middle).
- **Monitoramento (em ambiente de teste):**
  - ✔ Use ferramentas como **Wireshark** para capturar BPDUs e observar as mensagens de reconfiguração STP durante o ataque.
  - ✔ Monitore os logs dos switches gerenciáveis para ver alertas.

# Netscan: Introdução

- **Netscan**: Descobrir os hosts ativos em uma rede.
  - ✔ O Netscan é o processo de identificar hosts, serviços e vulnerabilidades em uma rede.
  - ✔ É uma etapa crucial tanto para defensores (auditoria) quanto para atacantes (reconhecimento).
  - ✔ Utiliza diferentes protocolos e técnicas para mapear a topologia e os recursos da rede.
- Uma vez de posse da lista de hosts e roteadores ativos na rede o atacante pode selecionar alvos.
- É muito difícil não ser descoberto ... principalmente na rede local ...

# Netscan: ARP e ICMP

- **Netscan**. As abordagens mais comuns são o uso de:
  - **ARP (Address Resolution Protocol):** Utilizado para mapear endereços IP a endereços MAC em redes locais (camada 2).

```
$ arp -a
? (200.17.220.114) at d0:94:66:c4:7d:7e [ether] on eno1
? (200.17.220.45) at 70:62:b8:c0:5a:f9 [ether] on eno1
? (200.17.220.103) at 00:e0:4c:68:00:ec [ether] on eno1
? (200.17.220.104) at d0:94:66:c4:90:8d [ether] on eno1
? (200.17.220.77) at d0:94:66:c4:8c:b1 [ether] on eno1
```

  - **ICMP (Internet Control Message Protocol):** Protocolo usado para enviar mensagens de controle e erro. O *ping* é um exemplo clássico de uso do ICMP para verificar a acessibilidade de um host.

```
$ ping 200.17.203.78
PING 200.17.203.78 (200.17.203.78) 56(84) bytes of data.
64 bytes from 200.17.203.78: icmp_seq=1 ttl=59 time=1.13 ms
64 bytes from 200.17.203.78: icmp_seq=2 ttl=59 time=1.23 ms
64 bytes from 200.17.203.78: icmp_seq=3 ttl=59 time=0.793 ms
64 bytes from 200.17.203.78: icmp_seq=4 ttl=59 time=0.956 ms
```

# Exemplo de Netscan com Nmap (ARP e ICMP)

- O Nmap (Network Mapper) é uma ferramenta poderosa para descoberta de rede e auditoria de segurança.
- Exemplos:
  - ✔ Ping Scan utilizando exclusivamente requisições ARP:

    ```
    $ nmap -sn -PR 192.168.1.0/24
    ```
  - ✔ Ping Scan utilizando requisições de eco ICMP:

    ```
    $ nmap -sn -PE 192.168.1.0/24
    ```
    - ✓ -sn: Desabilita varredura de portas (apenas host discovery).
    - ✓ -PR: Especifica o uso de ARP para descoberta de hosts (funciona apenas na rede local!).
    - ✓ -PE: Envia ICMP Echo Request (equivalente ao ping clássico).

# Mapeamento da Topologia da Rede com Traceroute

- **O que é Traceroute?**
  - ✔ Ferramenta de diagnóstico de rede que rastreia o caminho que um pacote IP percorre de um host de origem até um host de destino.
  - ✔ Revela os saltos (roteadores) intermediários e o tempo que leva para cada salto.
  - ✔ Essencial para entender a conectividade e a estrutura da rede.
- **Para que serve?**
  - ✔ Identificar rotas.
  - ✔ Diagnosticar problemas de latência ou perda de pacotes.
  - ✔ Descobrir a topologia de redes desconhecidas.
  - ✔ Auxiliar na solução de problemas de conectividade.

# Como o Traceroute Funciona?

- **O Princípio do TTL (Time To Live):**
  - ✔ O traceroute envia uma série de pacotes (geralmente UDP, mas também ICMP ou TCP).
  - ✔ Cada pacote tem um valor de TTL (Time To Live) inicial, que é um contador.
  - ✔ Cada roteador que encaminha o pacote decrementa o TTL em 1.
  - ✔ Quando o TTL de um pacote chega a 0, o roteador que o recebeu descarta o pacote e envia uma mensagem ICMP *Time Exceeded* (Tempo Excedido) de volta ao remetente.
- **Descoberta Sequencial dos Saltos:**
  - ✔ O traceroute começa enviando um pacote com TTL=1. O primeiro roteador responde com *Time Exceeded*.
  - ✔ Em seguida, envia um pacote com TTL=2. O segundo roteador responde.
  - ✔ O processo se repete, incrementando o TTL, até que o pacote chegue ao destino final ou um limite de saltos seja atingido.
- **Informações Coletadas:**
  - ✔ Endereço IP de cada roteador no caminho.
  - ✔ Tempo de ida e volta para cada salto.

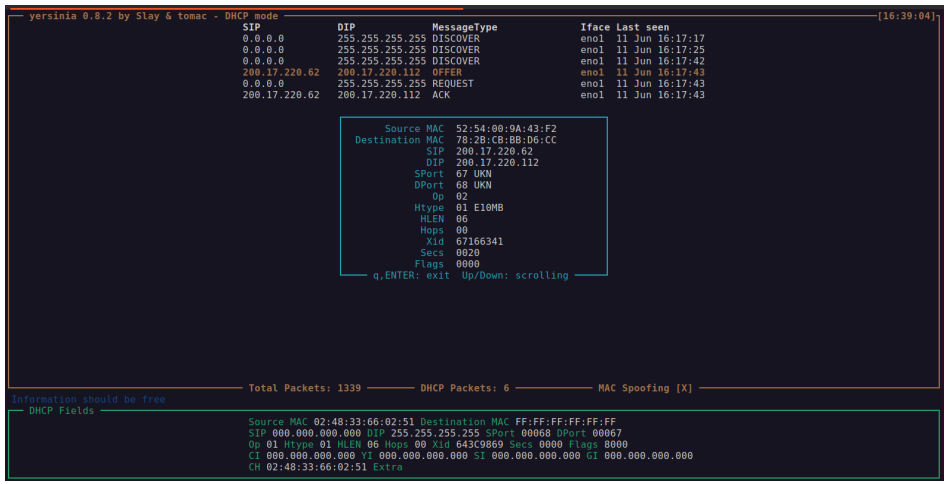
# Análise do Exemplo (Saída Típica)

- A saída do traceroute lista os saltos em ordem, do mais próximo ao mais distante.
- Cada linha representa um roteador e mostra:
  - ✔ O número do salto.
  - ✔ O nome do host (se puder ser resolvido) e o endereço IP do roteador.
  - ✔ Três tempos de ida e volta (RTT), que são os milissegundos (ms) para três tentativas de envio ao mesmo salto.
- * * * indica que não houve resposta daquele salto (pode ser firewall, roteador sobrecarregado, etc.).
- Exemplo:

```
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte
     packets
 1  200.17.220.62  0.319 ms  0.273 ms  0.492 ms
 2  10.10.19.1  2.188 ms  2.164 ms  2.403 ms
 3  10.10.100.1  1.157 ms  1.135 ms  1.112 ms
 4  200.19.74.125  3.273 ms  3.251 ms  3.230 ms
 5  200.238.139.17  1.311 ms  1.289 ms  1.267 ms
 6  170.79.214.60  2.179 ms  1.152 ms  2.039 ms
 7  170.79.213.108  7.589 ms  7.215 ms  7.393 ms
 8  * 187.16.216.55  7.485 ms 187.16.218.58  6.932 ms
 9  * 209.85.248.221  6.887 ms *
10  8.8.8.8  7.318 ms  7.205 ms  7.163 ms
```

# Ataque de Servidor DHCP Malicioso (Rogue DHCP Server)

- **O que é?**
  Um atacante introduz um servidor DHCP não autorizado na rede.
- **Objetivo Principal:**
  Interferir na atribuição de endereços IP e configurações de rede para clientes legítimos, frequentemente visando ataques de **Man-in-the-Middle (MitM)** ou **Negação de Serviço (DoS)**.
- Por exemplo, o Yersinia possui um módulo para este tipo de ataque:



# Rogue DHCP Server: Como Funciona?

1. **Introdução do Servidor Falso:**
   - ✔ Atacante conecta um dispositivo configurado como servidor DHCP (computador, roteador).
2. **Corrida por Respostas:**
   - ✔ Clientes na rede enviam requisições DHCP DISCOVER.
   - ✔ Servidor DHCP legítimo e o malicioso respondem.
   - ✔ A resposta do atacante pode chegar primeiro.
3. **Atribuição Maliciosa:**
   - ✔ Servidor malicioso atribui ao cliente:
     - ✓ **Endereço IP**.
     - ✓ **Gateway Padrão (Default Gateway)** controlado pelo atacante.
     - ✓ **Servidor DNS** controlado pelo atacante.

# Consequências do Ataque

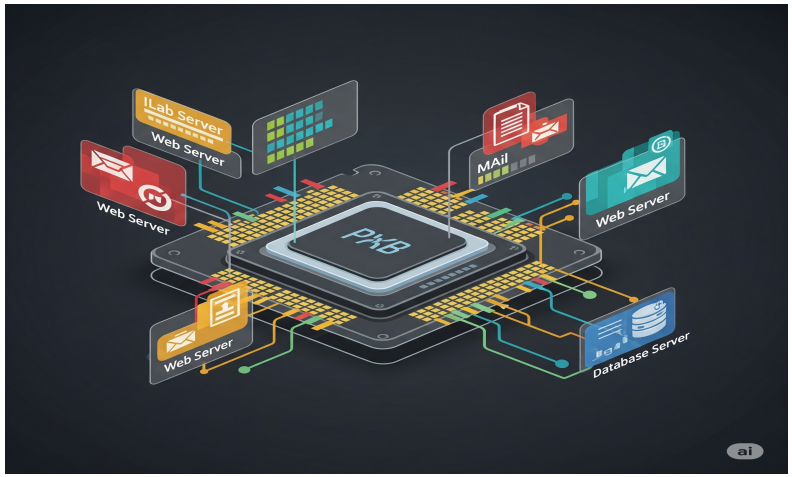
- **Negação de Serviço (DoS):**
  - ✔ Atribuição de IPs inválidos ou esgotamento do pool de endereços do servidor legítimo.
  - ✔ Clientes legítimos ficam sem conectividade.
- **Ataque Man-in-the-Middle (MitM):**
  - ✔ Todo o tráfego do cliente é roteado através da máquina do atacante.
  - ✔ Permite:
    - ✓ **Capturar e inspecionar o tráfego (Sniffing)**.
    - ✓ **Redirecionar o tráfego** para sites falsos (DNS Spoofing, Phishing).
    - ✓ **Modificar o tráfego** em tempo real.
- **Acesso Não Autorizado:**
  - ✔ Redirecionamento de tráfego para serviços controlados pelo atacante, buscando informações sensíveis.
  - ✔ Permite, por exemplo, o roubo de senhas.

# Mitigação de Ataques DHCP Maliciosos

- **DHCP Snooping:**
  - ✔ Recurso de segurança em **switches gerenciáveis**.
  - ✔ Permite que o switch confie apenas em portas específicas para pacotes DHCP vindos de servidores legítimos.
  - ✔ Descartar pacotes DHCP de fontes não autorizadas.
- **Autenticação de Portas (802.1X):**
  - ✔ Garante que apenas dispositivos autorizados possam se conectar à rede.
- **Segmentação de Rede:**
  - ✔ Dividir a rede em VLANs menores pode limitar o alcance de um ataque.
- **Monitoramento de Rede:**
  - ✔ Ferramentas de SIEM (Security Information and Event Management) podem detectar atividades DHCP anômalas.

# Port Scan

- **Port Scan** é um método poderoso para identificação de potenciais problemas de um host:
  - ✔ Processo de identificar quais portas TCP/UDP estão abertas em um host.
  - ✔ Portas abertas indicam serviços em execução que podem ser alvos de ataques.
  - ✔ Pode ser realizado explorando o *Three Way Handshake* do TCP.
- É uma etapa fundamental no reconhecimento de vulnerabilidades de sistemas.



# Exemplo de Port Scan com Nmap

- **Scan TCP** por padrão, utiliza o SYN Scan , o Nmap não completa a conexão TCP:

```
$ sudo nmap -p 1-1023 <IP_ALVO>
```

- **Scan TCP Connect (-sT):**
  - ✔ Realiza o handshake TCP completo.
  - ✔ Mais ruidoso (deixa logs no alvo).

```
$ nmap -sT <IP_ALVO>
```

- **Scan de UDP (-sU):**
  - ✔ Envia pacotes UDP e espera por respostas (ICMP Port Unreachable indica porta fechada).

```
$ nmap -sU <IP_ALVO>
```

# Exemplo de Nmap (prático)

- **Scan SYN Stealth em portas comuns:**

```
sudo nmap -sS -p 21,22,23,25,80,443 <IP_ALVO>
```

- **Scan de versão de serviços (detecção de vulnerabilidades):**

```
$ sudo nmap -sV p 21,22,23,25,80,139,443,666 <IP_ALVO>
Starting Nmap 7.80 ( https://nmap.org ) at 2025-06-10
     16:00 -03
Nmap scan report for 192.168.1.100
Host is up (0.0023s latency).
Not shown: 917 closed ports
PORT     STATE SERVICE
21/tcp   closed ftp
22/tcp   closed ssh
23/tcp   closed telnet
25/tcp   filtered smtp
80/tcp   open  http
139/tcp  open  netbios-ssn
443/tcp  open  https
666/tcp  filtered doom
```

- ✔ **Closed** O alvo respondeu indicando que não há servidor aguardando nestas portas.
- ✔ **Open** Há servidor escutando nas portas 80, 139 e 443.
- ✔ **Filterd** significa que uma porta está bloqueada por um firewall ou outro dispositivo, previnindo o scan de determinar o estado da porta.

# Ataques de Negação de Serviço (DoS)

- Objetivo: Tornar um serviço, host ou rede indisponível para seus usuários legítimos.
- Métodos comuns:
  - ✔ Consumo de largura de banda.
  - ✔ Consumo de recursos do servidor (CPU, memória, conexões).
  - ✔ Exploração de vulnerabilidades em protocolos ou aplicações.

# Ataque DoS: TCP SYN Flood

- Explora o handshake de três vias do TCP (SYN, SYN-ACK, ACK).
- O atacante envia uma grande quantidade de pacotes SYN para o servidor.
- O servidor responde com SYN-ACK, alocando recursos para cada conexão pendente.
- O atacante não envia o ACK final, deixando as conexões em estado de *half-open*.
- Isso esgota os recursos do servidor, impedindo novas conexões legítimas.
- **Ferramentas como hping3 podem ser usadas para este ataque.**
- TCP SYN Flood na porta 80:

```
$ sudo hping3 -S --flood -p 80 <endereco destino>
```

**Aviso:** Não execute este comando contra sistemas sem permissão!

# Ataque DoS: UDP Flood

- Inunda a porta alvo com pacotes UDP. Se o serviço na porta alvo responder (por exemplo, um serviço DNS na porta 53), os recursos do servidor podem ser esgotados. Se não houver serviço, o servidor pode enviar respostas ICMP "Port Unreachable", sobrecarregando a si mesmo e a rede.
- **Ferramentas como hping3 podem ser usadas para este ataque.**
- UDP Flood na porta 53:

```
$ sudo hping3 -2 --flood -p 53 <endereco destino>
```

**Aviso:** Não execute este comando contra sistemas sem permissão!

# Ataques Man-in-the-Middle (MitM)

- O atacante se posiciona entre duas partes que se comunicam.
- Intercepta, lê e potencialmente modifica o tráfego sem que as partes percebam.
- Exemplos: ARP Spoofing, DNS Spoofing, SSL Stripping.

# ARP Spoofing como Ataque MitM

- O atacante envia pacotes ARP falsificados para a rede.
- Ele engana os hosts, fazendo-os acreditar que o MAC do atacante corresponde ao IP do gateway (e vice-versa).
- Todo o tráfego entre os hosts passa pelo atacante.
- **Ferramentas:** *arpspoof*, *ettercap*.

# Exploração de Bugs Conhecidos de Servidores

- Muitos softwares e sistemas operacionais possuem vulnerabilidades conhecidas (CVEs, *Common Vulnerabilities and Exposures*).
- Atacantes buscam por versões de softwares com bugs conhecidos e usam estes *exploits*.
- **Exemplo:** *Buffer overflow*, *cross-site scripting* (XSS).

# Principais Fontes: NVD e CVE

- **National Vulnerability Database (NVD) - NIST**
  - **Site:** `https://nvd.nist.gov/`
  - **O que é:** A base de dados pública mais abrangente de vulnerabilidades, mantida pelo governo dos EUA.
  - **Conteúdo:** Agrega vulnerabilidades com IDs **CVE**, fornecendo descrições, classificações de gravidade (CVSS), produtos afetados (CPE) e referências a patches.
- **Common Vulnerabilities and Exposures (CVE) - MITRE**
  - **Site:** `https://cve.mitre.org/`
  - **O que é:** Programa que fornece um dicionário de vulnerabilidades conhecidas, atribuindo um ID **CVE** único a cada uma (ex: CVE-2023-12345).
  - **Importância:** É o padrão da indústria para nomear e referenciar vulnerabilidades. O NVD enriquece esses IDs com mais detalhes.

# Outras Fontes Importantes

- **Exploit-DB**
  - **Site:** `https://www.exploit-db.com/`
  - **O que é:** Arquivo de **exploits públicos** e papers relacionados, mantido pela Offensive Security.
  - **Uso:** Excelente para encontrar exploits práticos e entender como uma vulnerabilidade pode ser explorada.
- **OWASP Top 10**
  - **Site:** `https://owasp.org/www-project-top-ten/`
  - **O que é:** Lista das 10 principais categorias de riscos de segurança para aplicações web.
  - **Relevância:** Ajuda a entender as classes de vulnerabilidades mais críticas, mesmo que focado em web, muitos princípios se aplicam a aplicações de rede.
- **Bases de Conhecimento de Fornecedores**
  - Grandes fabricantes (Cisco, Microsoft, Oracle) mantêm seus próprios boletins de segurança e bases de dados para seus produtos.
  - Ideal para pesquisar vulnerabilidades em um produto específico.

# Como a Exploração Funciona?

- **Reconhecimento:** Identificar a versão do software/serviço (ex: Nmap -sV).
- **Pesquisa:** Buscar por exploits públicos para aquela versão (ex: Exploit-DB, Metasploit).
- **Execução:** Utilizar o exploit para obter acesso (shell, dados, etc.).
- **Prevenção:** Manter sistemas e softwares atualizados.
- **Atenção:** Mesmo mantendo sistemas autalizados, existe o tempo entre a descoberda do exploit e a implementação/divulgação do patch correspondente.

# Softwares para Exploração de Vulnerabilidades

- Ferramentas de exploração permitem testar a segurança de sistemas e aplicações.
- Diferentemente de scanners que identificam vulnerabilidades, estas ferramentas visam explorar falhas para demonstrar o impacto.
- Uma das primeiras propostas foi o antigo Security Administrator Tool for Analyzing Networks (SATAN), atualmente obsoleto; as ferramentas atuais são muito mais avançadas.
- Artigo "Improving the Security of Your Site by Breaking Into it", 1996, Dan Farmer: `http://www.porcupine.org/satan/admin-guide-to-cracking.html`

# Nessus: Scanner

- **O que é:** Um dos scanners de vulnerabilidade mais conhecidos e respeitados: `https://docs.tenable.com/nessus/Content/GettingStarted.htm`.
  - ✔ **nmap** *turbinado*.
- **Função no Ciclo de Exploração:** Primariamente para identificar vulnerabilidades, não explorá-las.
- **Capacidades:**
  - ✔ Varreduras detalhadas em redes, sistemas operacionais, bases de dados e aplicações.
  - ✔ Identifica vulnerabilidades conhecidas (CVEs).
- **Uso Prático:**. O Nessus fornece uma lista de falhas que podem ser exploradas com outras ferramentas.
- **Disponibilidade:** Versão gratuita (Nessus Essentials) para uso não comercial.

# Metasploit Framework

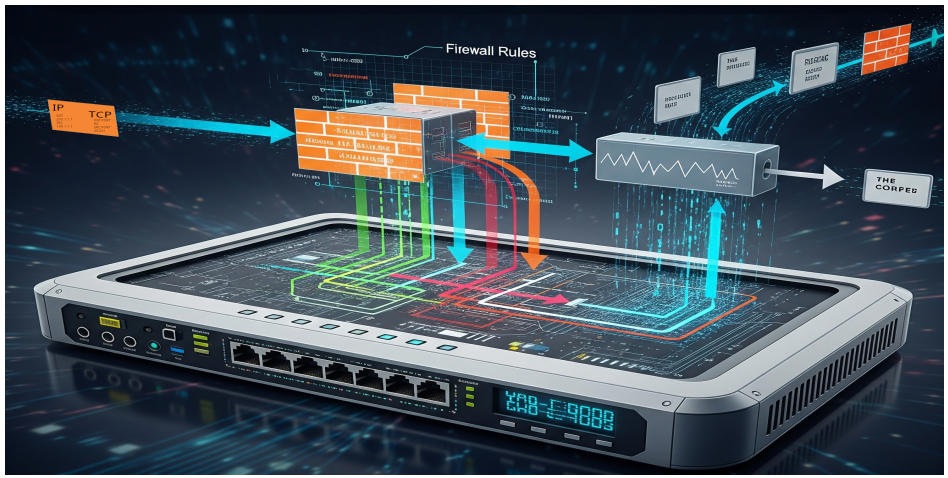
- **O que é:** Plataforma open-source (com versão Pro comercial) líder para desenvolvimento e execução de exploits.
- **Por que Usar:** É a ferramenta padrão para testadores de penetração e hackers éticos.
- **Capacidades Principais:**
  - ✔ **Milhares de Exploits:** Para SOs, aplicações de rede, softwares.
  - ✔ **Payloads Customizáveis:** Código malicioso executado no alvo (ex: shell reversa).
  - ✔ **Módulos Auxiliares:** Para varredura, fuzzing e coleta de informações.
  - ✔ **Pós-Exploração:** Ajuda a coletar dados do alvo comprometido.
- **Disponibilidade:** Pré-instalado em distribuições como o Kali Linux.

# Burp Suite: Especialista em Web

- **O que é:** Conjunto integrado de ferramentas para testar a segurança de **aplicações web**.
- **Por que Usar:** Fundamental para vulnerabilidades que se expõem via HTTP/HTTPS.
- **Capacidades Principais:**
  - ✔ **Proxy Interceptador:** Inspeciona, modifica e reenviada requisições/respostas HTTP/S.
  - ✔ **Scanner de Vulnerabilidades:** (Pro) Identifica falhas comuns em aplicações web (SQL Injection, XSS).
  - ✔ **Intruder:** Automação de ataques (força bruta, fuzzing, enumeração).
- **Uso Prático:** Essencial para testar APIs, sites e qualquer serviço web.
- **Disponibilidade:** Versão gratuita (Community) e versão paga (Professional).

# Firewall: O que é?

- Dispositivo de segurança de rede que monitora e filtra o tráfego de rede de entrada e saída.
- Opera com base em um conjunto de regras predefinidas.
- Atua como uma barreira entre uma rede confiável e uma rede não confiável (como a internet).



# Filtro de Pacotes (Packet Filtering)

- Forma mais básica de firewall.
- Toma decisões de permitir/negar com base em informações do cabeçalho do pacote:
  - ✔ Endereço IP de origem/destino.
  - ✔ Porta de origem/destino.
  - ✔ Tipo de protocolo (TCP, UDP, ICMP).
- **Exemplo (regra conceitual):** *Permitir apenas tráfego HTTP (porta 80) e HTTPS (porta 443) da internet para o servidor web interno.*
- As regras de um firewall são avaliadas em ORDEM SEQUENCIAL.
- A primeira regra que corresponde ao pacote é aplicada.
- Uma regra de *Negar Tudo* (Implicit Deny) é fundamental no final para garantir que apenas o tráfego explicitamente permitido passe.

# Regras de Firewall: Exemplo de Filtro de Pacotes

| Ord. | Ação | Prot. | Porta Origem. | IP Origem | Porta Dest. | IP Destino | Descrição |
|---|---|---|---|---|---|---|---|
| 1 | **Permitir** | TCP | Qualquer | Qualquer | 22 | 192.168.1.50 | Acesso SSH (Secure Shell) |
| 2 | **Permitir** | TCP | Qualquer | Qualquer | 80 | 192.168.1.50 | Acesso HTTP (Web) |
| 3 | **Permitir** | UDP | Qualquer | Qualquer | 53 | 192.168.1.50 | Acesso DNS (Domain Name System) |
| 4 | **Permitir** | ICMP | Qualquer | Qualquer | Qualquer | Qualquer | Permite ICMP |
| 5 | **Negar** | Qualquer | Qualquer | Qualquer | Qualquer | Qualquer | Descarta todo o resto do tráfego (Implicit Deny) |

- **Deve ser instalado na posição de um roteador onde o tráfego é obrigado a passar.**
- Firewalls modernos oferecem inspeção de estado (stateful), controle de aplicação e prevenção de intrusões.

# Outras Capacidades de um Firewall

- **Firewalls com Estado (Stateful Firewalls):**
  - ✔ Monitoram o estado das conexões ativas.
  - ✔ Mais inteligentes, permitem apenas respostas a conexões iniciadas internamente.
- **Firewalls de Próxima Geração (NGFW - Next-Generation Firewalls):**
  - ✔ Inspeção profunda de pacotes (DPI - Deep Packet Inspection).
  - ✔ Controle de aplicações.
  - ✔ Sistema de Prevenção de Intrusões (IPS) integrado.
  - ✔ Filtro de URL, antivírus.
- **Proxy Firewalls (Application-Layer Firewalls):**
  - ✔ Operam na camada de aplicação.
  - ✔ Oferecem maior granularidade de controle e segurança.
  - ✔ Agem como intermediários entre clientes e servidores.

# Engenharia Social: O Elo Mais Fraco da Segurança

- Série de *táticas psicológicas* usadas por cibercriminosos para manipular indivíduos a realizar ações ou divulgar informações confidenciais.
- Explora o *elemento humano*, muitas vezes o elo mais fraco na cadeia de segurança, manipulando emoções como curiosidade, medo, urgência ou desejo de ajudar.
- **Tipos Comuns de Ataques de Engenharia Social:**
  - ✔ **Phishing:** Envio de comunicações fraudulentas (e-mails, SMS, chamadas) de fontes confiáveis, para induzir o usuário a clicar em links maliciosos, baixar anexos infectados, fornecer credenciais.
  - ✔ **Pretexting:** Criação de um cenário falso (história) para obter informações (ex: se passar por TI, banco).
  - ✔ **Baiting (Isca):** Promessa de algo atraente (ex: pendrive infectado, download grátis) em troca de acesso/informações.
  - ✔ **Scareware:** Alertas e ameaças falsas para induzir a instalação de software malicioso ou compra de programas falsos.
  - ✔ **Tailgating/Piggybacking (Físico):** Seguir uma pessoa autorizada para entrar em áreas restritas sem autenticação.

# Conclusão

- A segurança de redes é um campo dinâmico e essencial.
- Conhecer as técnicas de ataque é fundamental para construir defesas robustas.
- Firewalls são a base da defesa, mas a segurança em camadas é crucial.
- Manter-se atualizado sobre novas ameaças e vulnerabilidades é imprescindível.
- O treinamento das equipes previne o uso de Engenharia Social.
- Entre 30% a 60% das violações de dados são provenientes da rede interna!