**Pontifícia Universidade Católica do Rio Grande do Sul**

**Laboratório de Redes de Computadores**

**Engenharia de Software**

**Carolina Ferreira, Felipe Freitas, Luiza Heller e Mateus Caçabuena**

**Trabalho 2**

**Porto Alegre**

**2024**

Sumário

[1. Introdução 2](#_Toc183388977)

[2. Descoberta de Hosts 3](#_Toc183388978)

[3. Execução do Ataque 7](#_Toc183388979)

[4. Topico 4 12](#_Toc183388980)

# Introdução

O presente relatório descreve o desenvolvimento e a execução de um projeto cujo objetivo foi implementar um ataque do tipo *man-in-the-middle* para capturar o histórico de navegação web de um computador alvo em uma rede local. Este trabalho foi conduzido no contexto da disciplina de Redes, da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), e teve como base a exploração de vulnerabilidades em redes utilizando técnicas avançadas de monitoramento e interceptação de tráfego.

A proposta foi dividida em três etapas principais: a descoberta de hosts ativos na rede, a execução do ataque de ARP Spoofing e a análise do tráfego capturado. Inicialmente, desenvolvemos uma aplicação para identificar os dispositivos conectados à rede, utilizando mensagens ICMP para determinar a atividade dos hosts. Na sequência, realizamos um ataque *man-in-the-middle* por meio de ARP Spoofing, permitindo a interceptação do tráfego entre o alvo e o roteador. Por fim, implementamos uma aplicação para capturar e analisar pacotes DNS e HTTP, reconstruindo o histórico de navegação do host alvo.

Este relatório apresenta uma descrição detalhada de cada etapa, incluindo os métodos e ferramentas utilizadas, os desafios enfrentados e os resultados obtidos. Além disso, são apresentados testes e análises realizados com o auxílio da ferramenta Wireshark, evidenciando a eficácia da solução desenvolvida.

# Descoberta de Hosts

A etapa de descoberta de hosts teve como objetivo identificar os dispositivos ativos em uma rede local, replicando o comportamento de uma varredura inicial no estilo ping scan. O código do arquivo host\_discovery.py foi utilizado para realizar essa descoberta.

**Funcionamento do Código**

O programa utiliza **socket raw** para criar pacotes ICMP personalizados, contendo cabeçalhos IP e ICMP. As principais funcionalidades do código incluem:

**1. Estruturas de Cabeçalho**

O código define as classes IP e ICMP para criar e manipular os cabeçalhos dos pacotes:

* **IP Header**:
  + Contém informações como versão (IPv4), TTL (*Time to Live*), e endereços de origem e destino.
  + Criado utilizando a função socket.inet\_aton para converter endereços IP em um formato binário.
* **ICMP Header**:
  + Contém o tipo (8 = requisição, 0 = resposta), código, checksum, identificador e número de sequência.
  + O checksum é calculado pela função calculate\_checksum, garantindo a integridade do pacote.

**2. Criação de Pacotes**

A função create\_packet combina os cabeçalhos IP e ICMP em um único pacote. Ela realiza:

* Montagem do cabeçalho IP usando struct.pack.
* Geração do checksum do pacote ICMP.
* Reempacotamento do cabeçalho ICMP com o checksum atualizado.

**3. Envio de Pacotes**

A função scan\_host realiza a varredura de um host específico:

* Cria um *socket* bruto e envia o pacote ICMP.
* Aguarda a resposta (usando recvfrom), registrando o tempo de resposta.
* Caso o host não responda dentro do tempo limite (timeout), ele é considerado inativo.

**4. Varredura da Rede**

A função scan executa a varredura em múltiplos hosts:

* Cria threads para realizar a varredura paralela de cada endereço IP da rede especificada.
* Filtra os endereços de *broadcast* e da rede.
* Ordena e exibe os resultados, destacando o pior tempo de resposta e a quantidade de hosts ativos.

**5. Saída do Programa**

Ao final da varredura, o programa exibe:

* Lista de hosts ativos e seus tempos de resposta.
* Total de hosts ativos, inativos e na rede.
* Pior tempo de resposta registrado.

A imagem fornecida no relatório do Wireshark demonstra os resultados práticos da varredura. Foram capturadas mensagens ICMP (protocolo *ping*), enviadas para diferentes hosts da rede, com respostas indicando os dispositivos ativos. Para cada mensagem de requisição (*request*) enviada, o Wireshark registrou uma resposta (*reply*), validando a presença de um host ativo naquele endereço IP.

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Figura 1: Host Discovery

Na captura anexada, é possível observar:

* **Protocolo Utilizado**: Mensagens ICMP (ping).
* **Atividade Registrada**: Os endereços IP das fontes e destinos indicam os hosts na rede sendo pingados (ex.: 192.168.15.64 para diferentes destinos na rede 192.168.15.X).
* **Respostas ICMP**: Para cada requisição, a linha correspondente mostra "reply in X ms", indicando que o host respondeu ao *ping* com o tempo de resposta registrado.
* **Hosts Ativos e Inativos**: O log também contém mensagens como "No response found!", indicando hosts que não responderam ao ping, ou seja, considerados inativos.

Esses dados validam a funcionalidade do script, que foi capaz de listar os IPs ativos conforme o objetivo do trabalho, utilizando uma abordagem sistemática baseada em ICMP.

# Execução do Ataque

Após a identificação do host alvo na rede, esta etapa foca na execução de um ataque do tipo ARP Spoofing para estabelecer uma posição de *man-in-the-middle* (MITM) entre o dispositivo alvo e o roteador da rede. O objetivo deste ataque é manipular as tabelas ARP dos dispositivos, redirecionando o tráfego de rede através do atacante, permitindo a interceptação e monitoramento das comunicações entre o host alvo e outros dispositivos.

A imagem a seguir retrata o script de ataque:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 2: Script do Ataque

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 3: Script de Ataque

No experimento realizado, ao acessar o container atacante, foram enviados dois *ARP replies* para manipular as tabelas ARP dos dispositivos alvo. O primeiro *ARP reply* foi enviado para o host vítima, informando que o atacante era o gateway, de modo a redirecionar o tráfego da vítima para o atacante. Em seguida, um segundo *ARP reply* foi enviado ao gateway, afirmando que o atacante era o host vítima, completando assim o processo de envenenamento ARP nos dois sentidos. Essa dupla manipulação é essencial para que o atacante consiga interceptar a comunicação bidirecional entre o gateway e o host vítima, possibilitando a captura e análise dos dados trafegados, sem interromper o fluxo normal de pacotes na rede.

**Verificação da Manipulação ARP por Meio das Tabelas ARP**

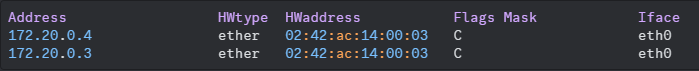
Com base na análise das tabelas ARP obtidas após a execução do ataque, foi possível confirmar o sucesso da manipulação ARP nos dispositivos envolvidos. Abaixo, detalha-se o comportamento observado em cada um dos dispositivos:

1. **Tabela ARP da Vítima**: A tabela ARP do container vítima (tf\_sockraw-labredes2-1) revela que tanto o endereço IP do gateway (172.20.0.3) quanto o endereço IP do atacante (172.20.0.2) estão associados ao mesmo endereço MAC **02:42:ac:14:00:03**. Isso indica que a vítima foi induzida a acreditar que o atacante é o gateway legítimo, redirecionando, portanto, todo o tráfego destinado ao gateway para o atacante.

Tela preta com letras brancas

Descrição gerada automaticamente

1. **Tabela ARP do Gateway**: A tabela ARP do gateway mostra que o endereço IP da vítima (172.20.0.4) também foi associado ao mesmo endereço MAC **02:42:ac:14:00:03**, pertencente ao atacante. Essa configuração confirma que o gateway foi igualmente enganado a acreditar que o atacante é a vítima.



**Análise e Conclusão**

A manipulação das tabelas ARP demonstra o sucesso da técnica de *ARP Spoofing*, pois ambas as partes, vítima e gateway, passaram a mapear o endereço MAC do atacante como sendo correspondente aos endereços IP do outro dispositivo. Isso coloca o atacante em uma posição de *man-in-the-middle*, onde ele é capaz de interceptar o tráfego bidirecional entre a vítima e o gateway.

**Fluxo de Tráfego**

Com o ataque bem-sucedido, o fluxo de tráfego foi alterado da seguinte maneira:

* **Tráfego da Vítima para o Gateway**: Os pacotes enviados pela vítima ao gateway agora passam pelo atacante, que pode interceptá-los, analisá-los ou modificá-los antes de repassá-los ao destino final.
* **Tráfego do Gateway para a Vítima**: Da mesma forma, os pacotes enviados pelo gateway à vítima também são redirecionados ao atacante antes de chegarem ao dispositivo de destino.

Essa configuração evidencia que o atacante assumiu controle total sobre a comunicação entre os dois dispositivos, sem interromper a continuidade do tráfego na rede, o que é característico de um ataque *man-in-the-middle*.

# 4. Topico 4