**Pontifícia Universidade Católica do Rio Grande do Sul**
**Faculdade de Engenharia**
Redes de Comunicação II


# Sniffer Remoto para Monitorar Histórico de Navegação Web

## Objetivo

O objetivo geral do trabalho é desenvolver uma aplicação usando *sockets* para monitorar o histórico de navegação Web de computadores alvo usando um ataque do tipo DHCP *Spoofing* com *man-in-the-middle*. Os objetivos específicos incluem:

- compreender de maneira prática o mecanismo de comunicação por *sockets*;
- entender o funcionamento dos protocolos da camada camada de aplicação (DHCP, DNS e HTTP) e seus problemas de segurança em redes locais.

## Descrição

Utilizando um programa *sniffer* (analisador de pacotes) é possível monitorar todo o tráfego de rede de um *host* e analisar seu conteúdo. Por exemplo, inspecionando pacotes dos protocolos DNS e HTTP é possível obter todo o histórico de navegação Web de um *host*. No entanto, isso normalmente necessita acesso físico a esse *host*. Uma forma de realizar essa monitoração remotamente em outros *hosts* de uma rede local é utilizando um ataque do tipo *man-in-the-middle*, explorando falhas de seguranças típicas de redes locais. Nesse trabalho, usaremos um ataque do tipo DHCP *spoofing* para interceptar o tráfego de rede e monitorar o histórico de navegação Web realizado por cada *host* atacado. A implementação desse trabalho pode ser dividida em duas partes bem definidas:

- Implementação de DHCP *spoofing* com *man-in-the-middle* (ANEXO I);
- Sniffer monitorando o histórico de navegação Web dos *hosts* atacados (ANEXO II).

Tudo deve ser documentado na forma de um relatório. Este relatório deve primeiramente descrever o funcionamento do protocolo DHCP e descrever como foi explorado o problema de segurança usando digramas, trechos de códigos e/ou capturas de tela (sugestão: utilize capturas de telas do Wireshark para facilitar a explicação). O relatório também deve descrever como foram extraídas as informações necessárias para geração do histórico de navegação dos *hosts*. Esse relatório deverá ser entregue juntamente com o código fonte utilizado.

O trabalho deve ser implementado na linguagem C. Exemplos utilizando *sockets* UDP/TCP e *sockets raw*, bem como o uso de *threads*, foram disponibilizados no Moodle.

## Resultados e Entrega

**Grupos:** Individual ou em dupla.
**Entrega:** Upload no Moodle de um arquivo zipado com o nome do(s) aluno(s), contendo:

1. Relatório descrevendo a implementação
2. Código da implementação

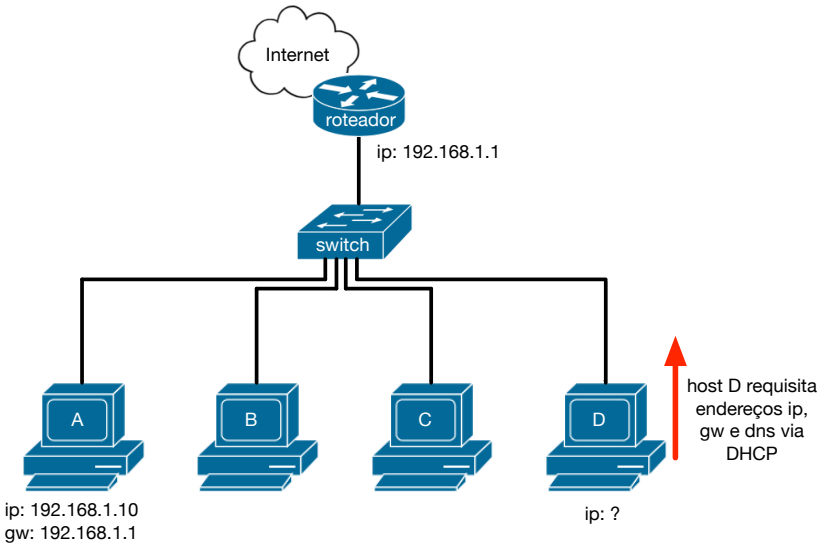
**Prazo final para entrega:** 08/11/2016 até as 19:30 (antes do início da aula)
**Data da apresentação:** 08/11/2016

**ANEXO I**

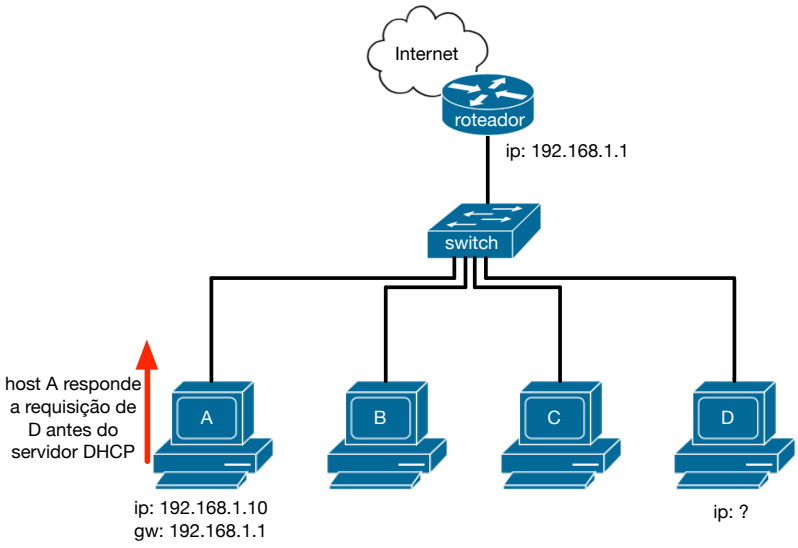
**Ataque DHCP Spoofing**

Ataques do tipo DHCP *spoofing* consistem basicamente em implementar um servidor DHCP simplificado que seja capaz de responder requisições de clientes antes do servidor principal da rede, informando endereços (IP do *host*, máscara, IP do *gateway* padrão e IP do servidor DNS) forjados. Ambos os servidores irão responder as requisições, mas, por normalmente estar mais próximo da vítima e possuir uma implementação mais simples, a resposta do atacante tende a chegar primeiro garantindo o sucesso da técnica. O ataque é descrito em mais detalhes nos diagramas a seguir, considerando um *host* atacante A e um *host* vítima D.

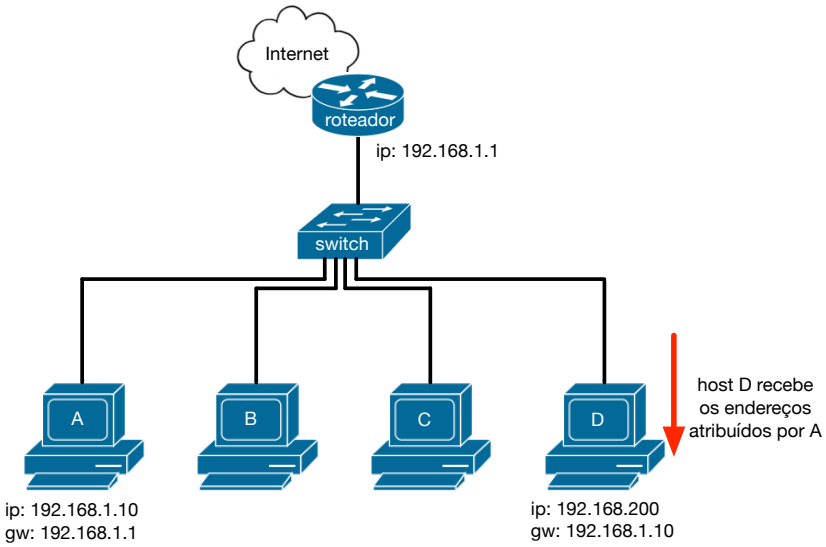
**Passo 1:** um *host* D que precisa de um endereço IP (por exemplo, durante a inicialização do sistema) envia uma mensagem via broadcast (DHCP Discover) para descobrir o endereço do servidor DHCP. Assim que o servidor DHCP responder (DHCP Offer) oferecendo o endereço IP, máscara..., o *host* cliente irá enviar uma mensagem a esse servidor aceitando o novo endereço IP oferecido (DHCP Request) e o servidor confirma com um DHCP Ack.




**Passo 2:** um *host* atacante A tenta responder a requisição enviada pelo *host* D antes do servidor DHCP principal da rede.



**Passo 3:** em caso de sucesso, o *host* atacante A torna-se o servidor DHCP responsável por atribuir endereços ao *host* vítima D (DHCP Ack).




Os endereços atribuídos ao *host* D podem, por exemplo, configurar o *gateway* padrão para apontar para o *host* atacante, permitindo a implementação de um ataque do tipo *man-in-the-middle*. Após esses passos, toda a comunição do *host* D com a Internet irá passar pelo *host* atacante.

Obs: o protocolo DHCP tem alguns detalhes de funcionamento que precisam ser estudados/entendidos para que essa técnica seja realizada. Uma atenção especial deve ser dada aos campos "Option" desse protocolo. Para um melhor entendimento do funcionamento do protocolo, recomenda-se a leitura do RFC que define o protocolo.

**Encaminhamento de pacotes**

Por padrão, o Linux descarta pacotes que são destinados a outros *hosts*. Desta forma, para implementar um ataque do tipo *man-in-the-middle*, é necessário habilitar a funcionalidade de encaminhamento de pacotes do kernel do Linux (IP Forwarding). Isso fará com que o tráfego entre o *host* alvo e o roteador não seja interrompido durante o ataque.

Para habilitar a funcionalidade de IP Forwarding, execute o seguinte comando no Linux:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

**ANEXO II**

**Monitoração do Histórico de Navegação Web das Vítimas**

Uma vez que um ataque do tipo *man-in-the-middle* foi realizado com sucesso, toda comunicação realizada entre a vítima e a Internet passará pelo *host* atacante. Desta forma, é possível implementar um programa *sniffer* utilizando *sockets raw* que analisa as resquisições DNS e HTTP e gera um arquivo contendo o histórico de navegação Web dos *hosts* alvo.

O arquivo de saída deve estar no formato HTML e cada entrada do histórico deve conter as seguintes informações:

- data e hora do acesso;
- endereço IP do host;
- nome do host;
- URL completa do endereço Web acessado (caso o endereço utilize HTTPS, fornecer apenas o nome do domínio).

Um exemplo de arquivo de saida é fornecido a seguir.

```
<html>
<header>
<title>Histórico de Navegação</title>
</header>
<body>
<ul>
<li>12/10/2016 22:24 - 192.168.25.21 (MBP-de-Marcelo) - <a
href="http://www.pucrs.br/facin/agenda/">http://www.pucrs.br/facin
/agenda/</a></li>
<li>12/10/2016 22:30 - 192.168.25.103 (Marcelo-PC) - <a
href="http://globoesporte.globo.com/blogs/especial-blog/brasil-
mundial-fc/post/jogo-da-paz-maradona-se-estranha-com-veron-e-
deixa-o-gramado-
irritado.html">http://globoesporte.globo.com/blogs/especial-
blog/brasil-mundial-fc/post/jogo-da-paz-maradona-se-estranha-com-
veron-e-deixa-o-gramado-irritado.html</a></li>
<li>12/10/2016 22:35 - 192.168.25.21 (MBP-de-Marcelo) - <a
href="https://www.facebook.com/">https://www.facebook.com/...</a><
/li>
<li>12/10/2016 22:36 - 192.168.25.21 (MBP-de-Marcelo) - <a
href="https://www.facebook.com/">https://www.facebook.com/...</a><
/li>
<li>12/10/2016 22:36 - 192.168.25.103 (Marcelo-PC) - <a
href="http://gshow.globo.com/tv/noticia/2016/10/angelica-e-
luciano-huck-sao-entrevistados-no-programa-do-
jo.html">http://gshow.globo.com/tv/noticia/2016/10/angelica-e-
luciano-huck-sao-entrevistados-no-programa-do-jo.html</a></li>
<li>12/10/2016 22:37 - 192.168.25.21 (MBP-de-Marcelo) - <a
href="https://www.facebook.com/">https://www.facebook.com/...</a><
/li>
<li>12/10/2016 22:40 - 192.168.25.21 (MBP-de-Marcelo) - <a
href="https://www.hipchat.com/">https://www.hipchat.com/...</a></l
i>
</ul>
</body>
</html>
```

A escolha pelo formato HTML foi realizada para permitir a visualização do histórico de navegação de forma similar a fornecida pelos navegadores Web (ex: Google Chrome). Um exemplo de visualização do arquivo criado é apresentado a seguir.


Obs: para obter todas as informações necessárias para gerar um histórico de monitoração como o demonstrado acima, será necessário combinar informações extraídas de pacotes DNS e HTTP. Note que apesar de simples, essa tarefa não é trivial! Recomenda-se um estudo detalhado desses protocolos, através de suas RFCs, e a realização de experimentos em laboratório utilizando Wireshark para monitorar o conteúdo dos pacotes desses protocolos. Também será necessário filtrar as requisições para não poluir o arquivo gerado. Por exemplo, o acesso a uma página Web pode gerar centenas de requisicões HTTP adicionais para obter todos os objetos ligados à página, tais como arquivos de imagens, folhas de estilo (CSS), java scripts, etc. O nome do *host* alvo pode ser obtido de mais de uma forma. Uma sugestão é a extração dessa informação dos campos "Option" de mensagens DHCP.