TRABALHO FINAL DE LABORATÓRIO DE REDES

Aloysio de Medeiros Winter, Carlo Smaniotto Mantovani e Felipe Elsner Silva

---

O trabalho desenvolvido consiste na criação de um *sniffer* de rede para detecção de ataques realizados em uma dada rede, para tanto é utilizado um Socket RAW associado a uma interface de rede especificada. Para representar a funcionalidade do *sniffer*, foram selecionados dois tipos de ataque, para quais o *sniffer* foi especialmente implementado:

- ICMP Flooding;
- ARP Spoofing.

Com o desenvolvimento deste trabalho, foi possível visualizar o funcionamento claro do *sniffer* que monitora especificamente esses dois ataques. Os ataques foram realizados através de duas ferramentas diferentes, uma para cada tipo de ataque.

Configuração de Rede para Teste

Para se visualizar claramente o funcionamento das ferramentas de ataque e do *sniffer* desenvolvimento, foi utilizado a ferramenta *docker* e *docker-compose* para criar redes isoladas, cujo tráfego nelas poderia ser observado explicitamente, sem qualquer outro tráfego desnecessário.

Ao criar uma rede, a ferramenta cria uma interface de rede associada a essa rede e às máquinas que estão conectadas a ela. Logo, o *sniffer* desenvolvido foi associado a uma das interfaces de rede criadas.

No caso, foram criadas duas redes, cada uma com sua interface, para simular um ataque de uma máquina externa a uma rede específica, assim, as máquinas criadas (contêineres), em cada rede foram:

- Rede NW (interface br0):
  - Idle
    - Endereço de rede IPv4: 172.20.0.2.
  - Victim
    - Endereço de rede IPv4: 172.20.0.3.

    - Spoofer
        - Endereço de rede IPv4: 172.20.0.4.

- Rede NW2 (interface br1)
    - Attacker
        - Endereço de rede IPv4: 172.21.0.2.

Para monitoramento, o *sniffer* foi associado a interface "br0" em todos os testes, a máquina "Attacker" foi responsável por ataques de ICMP *flooding* e a máquina "Spoofer" por ataques de ARP Spoofing.

ICMP Flooding

ICMP Flooding ou Ping Flooding, é uma forma de ataque de negação de serviço que visa sobrecarregar um servidor ou dispositivo de rede enviando uma grande quantidade de solicitações de mensagens ICMP do tipo *Echo-request* (pings). Essas solicitações possuem o objetivo de sobrecarregar o alvo com tráfego de rede, consumindo seus recursos e, eventualmente, tornando-o inacessível para usuários legítimos.

A ferramenta utilizada para o representar esse ataque e, dessa forma, testar o *sniffer* desenvolvido é o próprio comando *ping* presente em diversos sistemas operacionais nativamente. Esse comando possui uma *flag* "-f" que realiza *flooding* quando especificada junto com um dado endereço de rede qualquer.

Assim, ao se realizar esse ataque da máquina "Attacker" (172.21.0.2) para a máquina "Victim" (172.20.0.3) se pode observar o seguinte resultado no *Wireshark*:

1563…	6.290932287	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.290944574	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.290946851	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.290959556	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.290961860	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.290974122	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.290976793	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.290988868	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.290992452	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.291005398	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.291007531	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.291018198	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.291020508	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply
1563…	6.291033067	172.20.0.1	172.20.0.3	ICMP	98 Echo (ping) request
1563…	6.291035324	172.20.0.3	172.21.0.2	ICMP	98 Echo (ping) reply

Figura 1: Monitoramento de ICMP *flooding* por *Wireshark*

Nota-se a presença do endereço "172.20.0.1", isso ocorre pois esse endereço representa o *Default Gateway* da rede da máquina que recebe os pacotes ICMP, logo os pacotes devem passar por esse endereço.

Além disso, o *sniffer* ao detectar esse tráfego incomum bloqueia o tráfego temporariamente, como demonstrado abaixo:

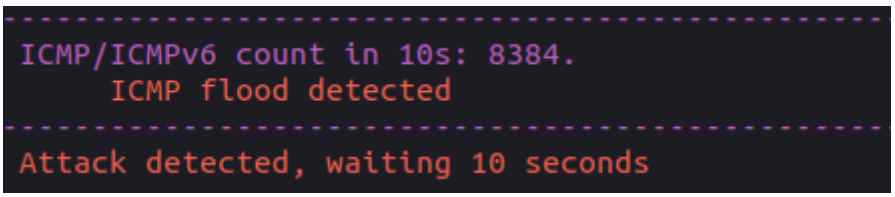

Figura 2: Detecção de ICMP Flooding pelo *sniffer*

A detecção de ataque do *sniffer* para esse tipo de ataque é realizada através de uma verificação da quantidade de pacotes ICMP na rede em um intervalo de 10 segundos, caso esse valor seja consideravelmente alto, um ataque é detectado e o tráfego na rede é bloqueado por 10 segundos.

ARP Spoofing

ARP Spoofing é uma técnica de ataque em rede na qual um atacante envia pacotes ARP falsificados em uma rede local, associando endereços MAC falsos a endereços IP legítimos. Isso pode levar a uma série de problemas de segurança, como interceptação de tráfego de rede, ataques de homem no meio (Man-in-the-Middle), e redirecionamento de tráfego. No caso, os endereços MAC são considerados falsos se eles diferem do endereço MAC da máquina que o endereço de rede está associado.

A ferramenta utilizada para os ataques realizados é a ferramenta "arpspoof" que faz parte do pacote "dsniff" em distribuições Linux. Ela possui a seguinte sintaxe:

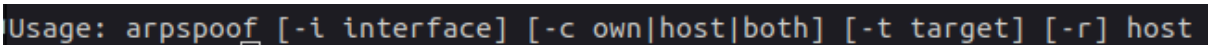

Figura 3: Sintaxe da ferramenta de ARP Spoofing

Ela foi utilizada para alterar a ARP cache de uma dada máquina alvo de forma que a máquina atacante que executa esse comando altera a ARP cache da máquina associando um endereço de rede especificado ao endereço MAC da máquina atacante. Por exemplo, foi executado o seguinte comando:

Figura 4: Comando utilizado para ARP Spoofing

Ao executar esse comando, a ARP Cache do alvo "172.20.0.2" é alterada de forma que o endereço MAC de "172.20.0.3" passe a ser o endereço MAC da máquina que executa o comando, no caso, a máquina "172.20.0.4". Assim, se obtém o seguinte resultado antes e após o ataque:

Figura 5: Estado da ARP Cache em 172.20.0.2 antes do ataque

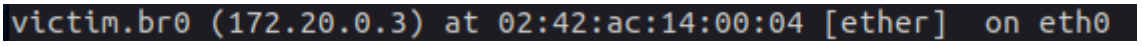

Figura 6: Estado da ARP Cache em 172.20.0.2 após o ataque

Se observa que, o endereço de rede não é alterado na ARP cache, porém o endereço MAC passa a ser o da máquina 172.20.0.4 (final 04), indicando que a manipulação da ARP Cache do alvo foi realizada corretamente.

Além disso, através do monitoramento pelo *Wireshark* é possível verificar uma característica comum a ataques de ARP Spoofing que é a presença de uma quantidade de ARP Replies consideravelmente maior do que ARP Requests durante o decorrer do ataque:

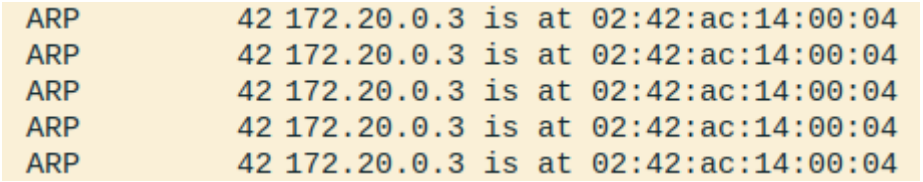

Figura 7: ARP Replies em sequência representados no *Wireshark*

Por fim, a detecção de ARP Spoofing pelo *sniffer* é realizada através da verificação da proporção de ARP Replies em relação a ARP Requests (Reply/Request) em um dado intervalo de tempo:

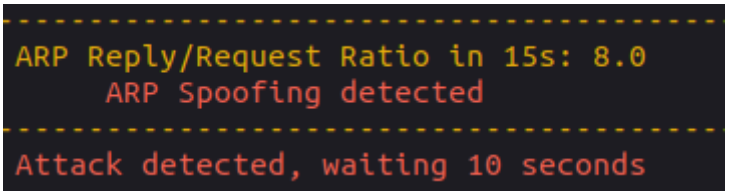

Figura 8: Detecção de ARP Spoofing pelo *sniffer*

Caso se observe uma quantidade de ARP Replies consideravelmente maior em um intervalo de 15 segundos, o *sniffer* detecta um ataque e bloqueia o tráfego na rede por 10 segundos.

Conclusão

Em resumo, se verificou de forma explícita o funcionamento das ferramentas de ataque e o funcionamento do *sniffer* desenvolvimento que corretamente detectava ataques na rede quando ocorriam. Por fim, se conclui que as ferramentas utilizadas para ataque e o *sniffer* atendem os requisitos do trabalho.