Eduardo Henrique dos Santos Cerolli
GRR20190397

Como executar:

dentro da pasta mitnick_attack_lab execute o seguinte comando para inciar o container docker:
docker compose up -d --build

Após isso execute o seguinte comando para abrir um bash dentro do x-terminal,:
docker exec -it x-terminal-10.9.0.5
depois o comando a baixo para dar ao trusted-server acesso a rsh no x-terminal:
echo "10.9.0.6" > /root/.rhosts

Execute os comandos para abrir um bash dentro do atacante e entrar na pasta volumes:
docker exec -it seed-attacker bash
cd volumes/
dentro da pasta execute os seguinte os comandos:
./instala_lib.sh (apertar Y quando solicitado para confirmar a instalação)
python3 arp_spoofing.py
python3 ataque.py
rsh 10.9.0.5 ls ..

o script instala_lib instala a lib NetfilterQueue usada no ataque.py
o scrpit arp_spoofing realiza o arp spoofing para o atacante se passar pelo trusted_server
o scrpit ataque faz uma conexão tcp se passando pelo trusted_server e enviar um mensagem RSH atualizando o arquivo .rhosts para conseguir acesso RSH direto pelo atacante.
o comando rsh listado mostra que o atacante conseguiu executar um comando remotamente no x-terminal

também pode conferir que o ataque deu certo usando o comando no bash do x-terminal:
cat /root/.rhosts

o conteúdo do arquivo ira conter um “+ +” adicionado no ataque, que garante acesso RSH ao atacante.

Documentação detalhada:

Para realização do arp spoof no arquivo arp_spoofing.py primeiro obtenho os endereços MAC das três maquinas, para ambos eu uso comandos bash chamados pelo python, para o endereço do atacante uso o comando ifconfig e para os endereços do x-terminal e do trusted-server primeiro envio mensagens de ping para atualizar a tabela arp do atacante e depois capturo o MAC com o comando arp.

Para efetivamente realizar o arp spoof eu monto uma mensagem de ARP com op 1 “who-is” para garantir que as tabelas sejam envenenadas mesmo que ainda não contenham o IP que eu quero envenenar, para o x-terminal eu envio o IP do trusted-server porem com o MAC do atacante, e para o trusted-server envio com IP do x-terminal e MAC do atacante. Envio cada mensagem dez vezes para ter mais garantia de que vai envenenar.

Para realizar o ataque no arquivo ataque.py, primeiro inicio obtendo a interface de rede usado pelo docker do atacante e inicio uma fila e faço bind para capturar as mensagens que passarão na rede, capturando as mensagens do tipo TCP com porta 514 e recolocando na fila o resto.

Após iniciar a fila forjo uma conexão TCP com o x-terminal me passando pelo trusted-server na porta de destino 514 que é a padrão de RSH, primeiro envio uma mensagem do tipo SYN com um numero de seq arbitrário (123), e recebo uma mensagem do tipo SYN-ACK confirmando meu SYN e recebendo o ISN criado pelo x-terminal, então envio uma mensagem de ACK confirmando esse ISN, com isso o three-way-handshake esta completo e a conexão foi aberta.

Com a conexão podemos enviar uma mensagem de RSH pois o x-terminal acha que somos o trusted-server e quando for validar o acesso a RSH pelo arquivo .rhosts o IP do trusted-server nos concedera acesso, então envio uma mensagem RSH com o comando “echo + + » *root*/.rhosts” que ira liberar acesso RSH no x-terminal a qualquer pessoa por causa da string + + sendo adicionada no final do arquivo.

Após isso o atacante tem acesso RSH ao x-terminal diretamente sem precisar se passar novamente pelo trusted-server.