Coluna do Alexandre Borges

NMAP – quarta parte

Avance em seus conhecimentos sobre a poderosa ferramenta NMAP.

Neste mês vamos comentar a respeito de outros detalhes do fantástico NMAP, tendo agora a oportunidade de explorar algumas opções muito interessantes, complementando as técnicas que abordamos nas primeiras colunas.

Retomando os textos anteriores, quando iniciamos o assunto do NMAP e mostramos como é simples realizar um levantamento de quais máquinas estão funcionando na rede de destino (*ping scanning*), trataremos de algumas possibilidades adicionais para agregar valor na execução deste tipo de escaneamento. Acompanhe os exemplos:

```
# nmap -sP -g 53 --data-length 32  192.168.1.0/24
```

Neste caso, estamos realizando um levantamento de quais máquinas estão funcionando na rede 192.168.1.0/24 através de um ping scanning (`-sP`), fixando a porta de origem do escaneamento (é possível enganar alguns firewalls que justamente deixam passar qualquer pacote vindo da porta de serviço do DNS) e, ainda, atribuindo um tamanho ao pacote usado pelo NMAP. Por quê? Porque alguns IDS produzem um alerta quando possuem pacotes com tamanho zero, porém quando fixamos um tamanho, estamos fugindo deste controle. O número 32 é o tamanho convencional de um ping proveniente de sistemas Windows e o número 56 é o respectivo tamanho de ping para sistemas Linux.

```
# nmap -sP -PS80 192.168.1.0/24
```

Este é um exemplo curioso pois é o mesmo exercício do comando anterior – um escaneamento para descobrir quais máquinas estão funcionando – contudo, estão sendo utilizados pacotes TCP e os mesmos são então enviados para a porta 80. O tamanho é zero e isto é uma desvantagem, mas o leitor pode usar a opção `--datalength <comprimento do pacote>` como demonstramos acima.

Qual é a vantagem disto? É que muitos sites bloqueiam pacotes ICMP e, neste caso, estão sendo usados pacotes TCP destinados à porta 80 que, costumeiramente, está aberta no firewall. Aliás, aqui a conexão não chega a ser estabelecida pois o NMAP envia um RST na terceira parte do *three-handshake*, assim como também ele não está interessado se a porta 80 está ou não aberta na máquina de destino, pois vindo uma resposta (SYN+ACK ou RST) é um sinal que a máquina alvo está em funcionamento. É evidente que máquinas que não estiverem com a porta 80 aberta ou com algo sendo executado não apresentarão resposta.

```
# nmap --packet-trace -sP -PA80 192.168.1.0/24
```

Esta é outra combinação curiosa. Neste exemplo estamos realizando um *TCP ACK scan* para descobrir quais máquinas estão up na rede, ou seja, ao invés de enviar um pacote SYN para depois esperar um SYN+ACK (como no exemplo anterior), o NMAP está enviando um pacote com o parâmetro ACK em 1. É claro que se a máquina de destino estiver no ar, ela responderá com um pacote RST, mas isto não importa e sim que a máquina esteja funcionando. É incrível notar que muitos firewalls estão preparados com regras para bloquear pacotes SYN entrantes, todavia esquecem de fazer o mesmo com estes pacotes ACK iniciais. É claro que tais pacotes seriam facilmente bloqueados com uma regra *stateful*.

A opção `--packet-trace` foi incluída por caráter educacional pois ela mostra todos os pacotes e flags enviados pelo NMAP. É muito instrutivo.

Agora que completamos de vez a etapa do levantamento de quais máquinas estão respondendo ou não em nossa rede alvo, já é possível apresentar outras possibilidades adicionais de scaneamentos de portas que são muito interessantes e que certamente o leitor vai apreciar.

Na edição passada, comentei a respeito da opção `-sS` (escaneamento invisível); alguns complementos também serão úteis aqui:

```
# nmap -sS <opção> 192.168.1.0/24
```

Onde <opção> pode ser:

- --host-timeout: tempo máximo que o NMAP se dedica ao escaneamento de uma máquina. Este tempo é medido em milissegundos.
- --min-rtt-timeout,--max-rtt-timeout: respectivamente tempos mínimos e máximos que o NMAP aguarda pela resposta de uma porta. Este tempo é em milissegundos e isto é útil para impedir que o NMAP sofra com atrasos da rede e, eventualmente, pule uma porta que poderia estar aberta apenas porque a resposta demorou para chegar.
- -d# : coloca o NMAP em modo de debug, onde # define o nível (de 1 a 9). O leitor pode esperar por uma saída muito detalhada com esta opção.
- -D <decoy1,decoy2,decoy3,...>: esta opção é muito interessante pois além dos pacotes normais que saem com o IP da máquina que está realizando o escaneamento, também envia diversas "iscas" (ou engodos) com o IP de origem que for escolhido na linha de comando do NMAP. A intenção é confundir o administrador da rede sendo atacada para que o mesmo tenha dúvidas sobre qual endereço IP realiza a invasão. Segue um exemplo:

```
# nmap -n -PN -D187.145.23.99,10.10.10.1,90.
➥ 90.10.2 -sS 192.168.1.100
```

Neste caso estamos executando um escaneamento invisível (-sS) na máquina 192.168.1.100, entretanto, enviamos diversos pacotes que servem meramente como engodo para confundir o IDS ou mesmo o administrador da máquina alvo relatando IPs de origem que não são os nossos (187.145.23.99, 10.10.10.1 e 90.90.10.2). A opção -n elimina a necessidade da resolução de nomes através do DNS, permitindo que o escaneamento seja bem mais rápido. A opção -PN também contribui, pois está isentando o NMAP de ter que executar um ping antes do escaneamento e varrer as portas assim mesmo. Lembre-se de que, se permitirmos que o NMAP realize um ping e não houver resposta, ele simplesmente para por ali e não faz qualquer escaneamento de portas!

No mês que vem volto com outras técnicas de escaneamento diferentes do -sS e que serão muito úteis em diversas situações. Até mais. ■

Alexandre Borges (alex_sun@terra.com.br) é instrutor independente e ministra regularmente treinamentos de tecnologia Oracle (áreas de Solaris, LDAP, Cluster, Containers/OracleVM, MySQL, e Hardware), Symantec (Netbackup, Veritas Cluster,Backup Exec, Storage Foundation e SEP) e EC-Council (CEH e CHFI), além de estar sempre envolvido com assuntos relacionados ao kernel Linux.