Explore o mecanismo de script NSE do Nmap

Script de segurança

O scanner de segurança Nmap possui um poderoso gerador de scripts. Analisamos a ferramenta para avaliar como eles são organizados.

por Ron McCarty

O Network Mapper (ou Nmap [1], como é mais conhecido) é um dos scanners de rede mais populares disponíveis para Linux, embora também funcione em outras plataformas. Administradores de sistemas e especialistas em segurança usam o Nmap para escanear, investigar e realizar inventários da rede. O Nmap é útil para as tarefas de segurança, mas também muito útil para resolver problemas de rede. É possível usar o Nmap para determinar se um serviço está sendo executado ou se sofreu alguma mudança. O Nmap também é muito bom para identificar aplicativos e versões de sistemas operacionais por meio de protocolos de impressões digitais. Como é projetado para trabalhar a partir da linha de comando, presta-se muito bem à criação de scripts em Shell para interpretar linguagem de scripts, como o Bourne again Shell (Bash) ou linguagens de processamento de textos, como Perl, sed ou awk. No entanto, o Nmap também tem um recurso nativo de escrita. O mecanismo de scripts do Nmap (NSE) possui muitas vantagens sobre o Bash ou outras linguagens de processamento de textos, conheça algumas:

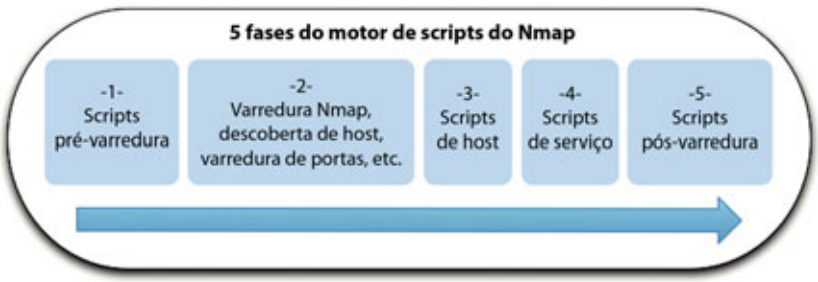

Figura 1: O NSE divide o processo de escaneamento em cinco fases.

- Consciência sobre a fase: como veremos a seguir, o Nmap organiza o processo de escaneamento em fases. O NSE, por sua vez, entende a fase do Nmap, permitindo ao programador evitar recorrências complexas ou algoritmos if-then para determinar o estado do escaneamento;
- Linguagem comum para portabilidade: uma linguagem comum torna o script portável para outras plataformas. Um script Bash, por outro lado, funcionaria bem no Linux mas mal em um sistema Windows;
- Distribuição baseada em comunidade: uma linguagem comum permite um padrão comum e um sistema para a distribuição de scripts tanto dentro do Nmap como separadamente.

A linguagem de scripts Nmap usa a linguagem de programação Lua. Lua é uma linguagem interpretada rápida, poderosa e leve [2]. É provavelmente uma das mais conhecidas no campo das linguagens de scripts por ser usada no

popular jogo World of Warcraft [3]. No entanto, Lua também é usada com o Adobe Lightroom e outras ferramentas.

Uma ampla gama de scripts está disponível online no web site Nmap.org [4], e alguns deles estão empacotados com o aplicativo. Você pode usar o comando `ls` para obter uma lista de scripts NSE instalados localmente (**listagem 1**).

O guia de referência do Nmap [5] oferece uma boa descrição dos scripts distribuídos com o Nmap.

Entenda os Scripts Nmap

O Nmap organiza o processo de escaneamento em cinco fases (**figura 1**). Os scripts NSE se dividem em categorias ligadas às fases do processo de escaneamento, como:

- Scripts anteriores à regra – executados antes que o Nmap tenha executado qualquer escaneamento na rede. Eles podem cuidar de atividades de limpeza, criação de tabelas, determinação de ambientes ou até mesmo geração da lista de endereços que o Nmap vai escanear. Esses scripts são chamados de *prerules*;
- Scripts host – executados contra cada host específico para o Nmap assim que o escaneamento do host estiver completo. Um script para coletar os endereços MAC dos hosts da LAN é um bom exemplo de script que poderia ser executado durante a fase de scripts host e são executados somente contra hosts existentes. Os scripts de host são chamados de *hostrules*;
- Scripts de serviço – são executados antes dos scripts host e somente se um determinado serviço estiver presente. O Nmap inclui muitos scripts de serviço e essa é a fase na qual receberá o maior foco em scripts suportados pela comunidade. O termo *portrule* identifica a fase.

Os scripts NSE são divididos nas seguintes categorias:

- Description
- Categories
- Phase
- Action

A seção `Description` oferece uma descrição de script incluído entre colchetes:

```
Description = [[
Exemplo para a Linux Magazine
]]
```

A seção `Categories` define a categoria na qual o script é executado (**quadro 2**). Associar um script com uma categoria, permite que ele seja executado como parte de uma categoria de escaneamento. Um script pode pertencer a múltiplas categorias. O formato para a seção de categorias é:

Listagem 1: Obter uma lista de scripts NSE locais

```
01 root@dal01:~# ls -l /usr/share/nmap/scripts
02 total 564
03 -rw-r--r-- 1 root root 15655 2009-11-06 03:39 asn-query.nse
04 -rw-r--r-- 1 root root 1864 2009-11-06 03:39 auth-owners.nse
05 -rw-r--r-- 1 root root 705 2009-11-06 03:39 auth-spoof.nse
06 -rw-r--r-- 1 root root 5582 2009-11-06 03:39 banner.nse
07 ~
08 ~
09 -rw-r--r-- 1 root root 6636 2009-11-06 03:39 sslv2.nse
10 -rw-r--r-- 1 root root 5376 2009-11-06 03:39 telnet-brute.nse
11 -rw-r--r-- 1 root root 5780 2009-11-06 03:39 upnp-info.nse
12 -rw-r--r-- 1 root root 89999 2009-11-06 03:39 whois.nse
```

Listagem 2: Código Portrule

```
01 portrule = function(host, port)
02 local svc = { std = { ["http"] = 1, ["http-alt"] = 1 },
03 ssl = { ["https"] = 1, ["https-alt"] = 1 } }
04 if port.protocol ~= 'tcp'
05 or not ( svc.std[port.service] or svc.ssl[port.service] ) then
06 return false
07 end
08 -- Don't bother running on SSL ports if we don't have SSL.
09 if (svc.ssl[port.service] or port.version.service_tunnel == 'ssl')
10 and not nmap.have_ssl() then
11 return false
12 end
13 return true
14 end
```

Listagem 3: Execução da ação

```
01 socket:set_timeout(5000)
02 try(socket:connect(host.ip, port.number, port.protocol))
03 try(socket:send("USER anonymous\r\n"))
04 try(socket:send("PASS IEUser@\r\n"))
05 while status do
06 status, result = socket:receive_lines(1);
07 if string.match(result, "^230") then
08 isAnon = true
09 break
10 end
```

Quadro 1: Mais sobre o Nmap

Esse artigo foca na criação de scripts presumindo que você possui um conhecimento prévio da ferramenta Nmap. Se você busca a introdução básica para o Nmap, o guia de referência oficial [5] é um bom começo.

O Nmap está disponível para a maioria das distribuições, então se não estiver atualmente em seu sistema, provavelmente pode instalá-lo por meio do gerenciador de pacotes de sua distribuição. Se não estiver disponível em sua distribuição, faça o download direto do web site [6].

```
categories = {"default", "safe"}
```

A entrada precedente especifica que o script será executado a qualquer momento que o administrador escolher por executar scripts tanto na categoria *Default* quanto na categoria *Safe*.

A seção *Phase* identifica a fase na qual o script será executado. Note que as configurações `hostrule` e `portrule` requerem parâmetros e recebem os parâmetros do mecanismo de escaneamento do Nmap. A seção `Phase` termina com um comando Lua: `end` (as fases são implementadas como funções Lua). A **listagem 2** mostra o código `portrule` do script `html-title.nse`

O portrule na **listagem 2** cria as variáveis `svc.std` e `svc.ssl` como arrays associativas (**linhas 2** e **3**) e depois checa se o protocolo de transporte não é TCP (**linha 4**) ou se a porta não está executando o serviço em HTTP ou HTTP sobre SSL (**linha 5**). Se essas condições não forem verdadeiras, o script simplesmente retornará o valor `false` (**linha 6**) e o Nmap não imprimirá qualquer resultado.

Por outro lado, se a porta está sendo executada em TCP, a lógica continua e o script determina se o Nmap tem acesso as bibliotecas SSL do cliente. Se não tiver, retornará o valor `false` (**linha 11**), sem imprimir nada durante a execução. Se todos os testes forem positivos, a função fica válida e a lógica definida na seção `Action` é levada a cabo.

A seção `Action` é o algoritmo contendo a lógica do script uma vez que a fase lógica é executada. A fase lógica usa variáveis Lua padrão, sintaxe e funções, combinadas com as bibliotecas incluídas no Nmap. A presença de bibliotecas Nmap NSE poupa o desenvolvedor de ter que lidar com funções como fazer conexão com a porta ou ler a partir de um serviço conectado à rede.

O código na **listagem 3**, que vem do script anônimo FTP Nmap `ftp-anon.nse`, mostra quão fácil é conectar e enviar dados para um socket e conferir um código de status.

Quadro 2: Categorias de script

A comunidade Nmap classifica os scripts por categoria. Se você está procurando um deles, ou planeja escrever seus próprios scripts, é importante entender a metodologia pela qual são classificados:

- **Auth** – A categoria é composta por mais de 30 scripts para testar configurações de protocolos de autenticação e controles no host alvo. A maioria desses scripts está relacionada a testes de vulnerabilidade e de força bruta, portanto devem ser executados somente em redes onde se permitem métodos de testes de detecção de intrusos.
- **Broadcast** – A categoria Broadcast é bem pequena, com seis scripts, e usa métodos broadcast e multicast para descobrir informações na rede. Os scripts incluem identificação de clientes Dropbox, descoberta de serviços host usando serviços DNS e serviços web multicasting.
- **Default** – Esta categoria inclui mais de 50 scripts. A equipe que desenvolve o Nmap usa seis critérios para colocar um script nessa categoria: velocidade, grau de utilidade, verbosidade, confiabilidade, grau de intrusão e privacidade.
- **Discovery** – Os scripts Discovery servem para usar informações de redes e sistemas operacionais para mapear hosts de rede. Mais de 70 scripts vêm com o mecanismo de escaneamento Nmap.
- **DoS** – Os scripts que podem causar falhas no serviço ou outros problemas mais graves estão na categoria DoS. Seu uso deve ser avaliado com cuidado, devido à possibilidade de indisponibilizar serviços essenciais. Há somente dois deles.
- **Exploit** – Os scripts Exploit tentam executar um exploit quando o Nmap identifica um host vulnerável. Conta com apenas um script.
- **External** – Os scripts External comunicam-se com hosts diferentes dos hosts e targets locais. Por exemplo, um script que atualiza automaticamente um banco de dados com informação heurística é classificado como External.
- **Fuzzer** – Esses scripts descobrem novas vulnerabilidades ou técnicas de identificação e funcionam enviando dados randômicos ao servidor.
- **Intrusive** – Os scripts Intrusive podem derrubar serviços e sistemas ou consumir enormes quantidades de largura de banda.
- **Malware** – Identificam malwares sendo executados em hosts remotos.
- **Safe** – São adaptáveis à rede e não são considerados intrusivos.
- **Version** – Cobre scripts projetados para identificar explicitamente as versões específicas de serviços.
- **Vuln** – Categoria para identificar vulnerabilidades, sem explorá-las.

Listagem 4: Execução de scripts Default

```
01 root@dal01:~# nmap -sC www.mcwrite.net
02
03 Starting Nmap 5.00 ( http://nmap.org ) at 2011-03-06 19:35 CST
04 Interesting ports on blogs.typepad.com (204.9.177.195):
05 Not shown: 998 filtered ports
06 PORT STATE SERVICE
07 80/tcp open http
08 | robots.txt: has 7 disallowed entries
09 |_ /t/trackback /t/comments /t/stats /t/app /.m/ / *
10 |_ html-title: Ron McCartys Blog
11 2909/tcp open unknown
12
13 Nmap done: 1 IP address (1 host up) scanned in 12.93 seconds
14 root@dal01:~#
```

Uso do NSE

Muitos dos scripts NSE básicos são membros da categoria padrão. Use a opção `-sC` para executar scripts padrão (**listagem 4**).

O parâmetro `-A` também executa os scripts padrão, junto com a execução de diversas opções de escaneamento. Para executar um script específico, use o parâmetro `--script` com o nome do mesmo. O exemplo:

```
nmap --script html-title.
```

executa o script `Whois` incluído no NSE (**listagem 5**). O NSE também permite que diversos scripts sejam executados separados por vírgulas, ou categorias completas por meio da inclusão do nome da categoria. Por exemplo, o comando na **listagem 6** executa todos os scripts da categoria *Malware*.

Conclusão

Neste artigo, introduzi os leitores ao NSE. Espero que essa discussão dê uma boa base para determinar quando você precisa de scripts. Um entendimento sólido das categorias Nmap e scripts, além de algum conhecimento do mecanismo de criação de scripts em si, o ajudará a usar o Nmap em todo seu potencial. Divirta-se! ■

Mais informações

[1] Nmap: http://nmap.org/

[2] Lua: http://www.lua.org/about.html

[3] World of Warcraft: http://us.battle.net/wow/en/

[4] Scripts Nmap: http://nmap.org/nsedoc/

[5] Guia de referência do Nmap: http://nmap.org/book/man.html

[6] Downloads relacionados ao Nmap: http://nmap.org/download.html

Gostou do artigo?

Queremos ouvir sua opinião. Fale conosco em cartas@linuxmagazine.com.br

Este artigo no nosso site: http://lnm.com.br/article/5598

Listagem 5: Execução do script html-title

```
01 root@dal01:~# nmap --script html-title www.mcwrite.net
02
03 Starting Nmap 5.00 ( http://nmap.org ) at 2011-03-06 19:18 CST
04 Interesting ports on blogs.typepad.com (204.9.177.195):
05 Not shown: 998 filtered ports
06 PORT STATE SERVICE
07 80/tcp open http
08 |_ html-title: Ron McCarty's Blog
09 2909/tcp open unknown
10
11 Nmap done: 1 IP address (1 host up) scanned in 8.47 seconds
12 root@dal01:~#
```

Listagem 6: Execução de scripts de uma categoria específica

```
01 root@dal01:~# nmap --script malware www.mcwrite.net
02
03 Starting Nmap 5.00 ( http://nmap.org ) at 2011-03-06 19:23 CST
04 Interesting ports on blogs.typepad.com (204.9.177.195):
05 Not shown: 998 filtered ports
06 PORT STATE SERVICE
07 80/tcp open http
08 2909/tcp open unknown
09
10 Nmap done: 1 IP address (1 host up) scanned in 11.02 seconds
11 root@dal01:~#
```