

Entendendo as técnicas de análise com o Nmap

# Invasão!

Qualquer um que trabalhe há mais de dois meses com redes de computadores já ouviu falar no Nmap, o mais famoso scanner de vulnerabilidades do mundo livre. Mas de que maneira o Nmap identifica as brechas de segurança em sua rede? Neste artigo, examinaremos algumas de suas técnicas.

POR CHRISTIAN NEY

O ato de varrer uma rede em busca de vulnerabilidades é quase tão antigo quanto as próprias redes. Nos saudosos tempos de outrora, em que o único modo de conexão remota eram as linhas telefônicas, “hackers” usavam modems para testar blocos de números de telefone. Esses vivaldinos registravam em um caderno todas as respostas obtidas – um panaca gritalhão, um sinal de fax, uma voz feminina, um outro modem atendendo à ligação... O processo era chamado de *wardialing* (em português, algo como “guerra de discagem”).

Hoje, os chamados scanners de rede ou *port scanners* é que são a coqueluche. Eles transmitem, em direção ao sistema sob teste, pacotes IP (ou TCP/UDP) especialmente manipulados. Dependendo da reação (ou falta dela) é possível identificar os tipos e modelos de sistemas, bem como mapear os serviços oferecidos e as possíveis vulnerabilidades existentes.

O *Nmap* (The Network Mapper [1]), trazido à luz por Fyodor em setembro de 1997 [2], é provavelmente um dos mais completos scanners de rede que existem. Fyodor estava insatisfeito com os recursos que ferramentas como o *Strobe* [3] e o *Pscan* [4] ofereciam. Ele queria um programa que superasse qualquer outro conhecido – e realmente conseguiu.

O Nmap é distribuído em praticamente qualquer CD de Linux que você possa encontrar por aí. Se sua distribuição preferida não possuir o Nmap, escreva para lá reclamando bastante e depois baixe o programa a partir do site oficial [1]. O Nmap usa a técnica de impressão digital TCP (*TCP fingerprinting*) para identificar o sistema operacional da máquina que estamos varrendo – a máquina sob ataque. O programa consegue determinar há quanto tempo o computador está ligado (*uptime*) e realmente identificar os serviços oferecidos, em um nível de detalhe que inclui o número da porta, o nome do *daemon* responsável e até a versão desse *daemon*.

O número fabuloso de funções oferecidas pelo Nmap indica, por outro lado, que a quantidade de opções da linha de comando é, também, espetacular. Há não menos do que 15 métodos de se varrer um dado nó (tabela 1) e aproximadamente 20 variações e configurações para cada método – alguns deles manipulam os pacotes IP

e TCP de maneira a ficarem completamente irreconhecíveis. Não é preciso ser um especialista para usar o Nmap, mas os iniciantes talvez fiquem um pouco confusos com a teoria envolvida e a quantidade de opções do programa. Infelizmente temos que alertar: para usar o Nmap em toda a sua plenitude, é necessário ter uma boa bagagem a respeito de redes e TCP/IP. Este artigo descreve algumas das técnicas que o Nmap usa para descobrir vulnerabilidades.

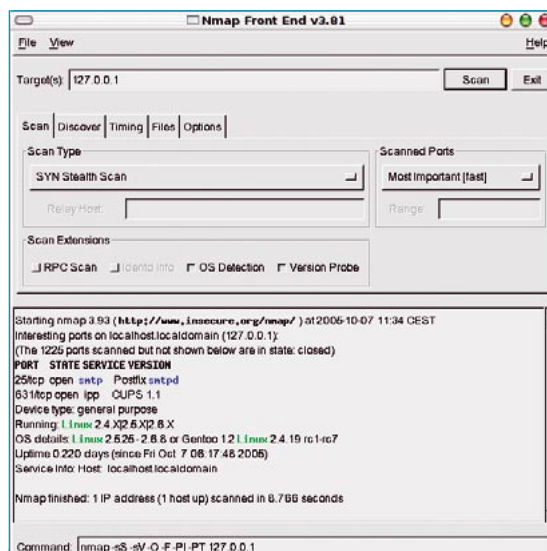


Figura 1: O *Nmapfe*, interface gráfica do Nmap, permite que usemos o programa sem ter que decorar a grande quantidade de opções da linha de comando – e, de quebra, mostra como seria esse comando (observe a linha na parte de baixo da tela).

Tabela 1: Técnicas de varredura

Técnica	Sintaxe	Aplicação
TCP SYN	-sS	Varredura camuflada ("stealth")
Conexão TCP	-sT	Varredura por usuário comum (facilmente detectável)
TCP FIN	-sF	Varredura camuflada ("stealth")
Árvore de Natal	-sX	Varredura camuflada ("stealth")
TCP Vazio (Null)	-sN	Varredura camuflada ("stealth")
Ping	-sP	Determinar se o computador está ligado
Deteção de Versões	-sV	Identificar os serviços e as versões dos daemons
UDP	-sU	Identificar portas UDP abertas
Protocolo IP	-sO	Identificar os protocolos suportados
ACK	-sA	Identificar firewalls
Janela ACK	-sW	Varredura ACK mais especializada
RPC	-sR	Identificar serviços RPC
Lista	-sL	"Boneco de testes"
Passivo ("Idle Scan")	-sI	Varredura usando um "laranja" como despiste
FTP Bounce	-b	Forma antiga de ataque, presente por motivos históricos

## Bandido, eu?

Muitos rotulam o Nmap como sendo uma ferramenta para malfeitores. De fato, os "chapéus pretos" usam e abusam do Nmap como auxiliar no ataque e invasão às suas vítimas.

Mas, assim como um martelo pode ser usado para colocar um prego na parede, também podemos rachar a cabeça de alguém com ele. Da mesma forma, o Nmap é, nas mãos dos administradores de rede, um grande aliado no diagnóstico dos problemas de suas redes. Os responsáveis por sistemas de informação têm por obrigação conhecer todas as falhas e brechas que a rede sob sua batuta possui. Há muitas histórias de administradores que decidem fazer uma análise abrangente de suas redes e descobrem estarecidos que há muitos serviços negligenciados ou mesmo esquecidos. O Nmap também ajuda no inventário da rede, no teste de penetração em firewalls e a documentar as atualizações de todos os sistemas. Uma mão na roda, não?

O Nmap foi, originalmente, desenvolvido para o Linux. Hoje, entretanto, há versões para Windows®, Free/Open/Net/\*BSD e vários sabores de Unix. Apesar de

ter todo o seu potencial explorado apenas na linha de comando, há interfaces gráficas para ele tanto no Unix (*Nmapfe*, figura 1) quanto em sistemas que vieram de algum lugar do noroeste dos Estados Unidos (*Nmapwin* [5]). É possível até colocar o Nmap em um servidor e controlá-lo pelo browser (*PHP-Nmap* [6] – figura 2). A página oficial do Nmap possui uma lista bastante grande com projetos baseados no Nmap [7]. Há até uma versão do programa para o *Zaurus*, assistente pessoal portátil (PDA) da Sharp que usa Linux como sistema operacional.

## O truque das três cartas

O Nmap possui um método de varredura composto por três estágios. Por padrão, esses três estágios são:

⇒ O Nmap tenta determinar se o sistema sob ataque (vamos chamá-lo de "alvo") está "vivo" – isto é, funcionando e conectado. O usuário pode escolher entre o método tradicional que usa uma mensa-

gem *ICMP Echo Request* – a mesma usada pelo comando `ping` – e o método próprio do Nmap, muito mais moderno e eficiente. Lembre-se de que a maioria dos sistemas operacionais de hoje pode ser configurada para não responder a um ping.

⇒ Depois, o Nmap faz uma consulta ao DNS para tentar determinar o nome de host associado ao endereço IP do alvo. Este estágio pode ser desativado se o usuário assim preferir.

⇒ Por fim, o Nmap varre o alvo usando a técnica selecionada pelo usuário quando emitiu o comando. Para interromper o processo de varredura, basta pressionar [Ctrl] + [C]. É possível pedir ao Nmap que grave um arquivo de registro (*log*) que, além de servir para sua análise futura, permite que ele continue o teste de onde parou.

Há quatro estados distintos de porta reconhecidos pelo Nmap – veja a **tab**ela 2. Um dos pontos fortes do Nmap é a grande quantidade de técnicas de varredura disponíveis. Em vez de simplesmente iniciar uma conexão TCP completa – uma negociação em três etapas conhecida como *three-way handshake* – o programa transmite pacotes especialmente armados para contrariar todas as disposições das RFCs vigentes (*RFC*

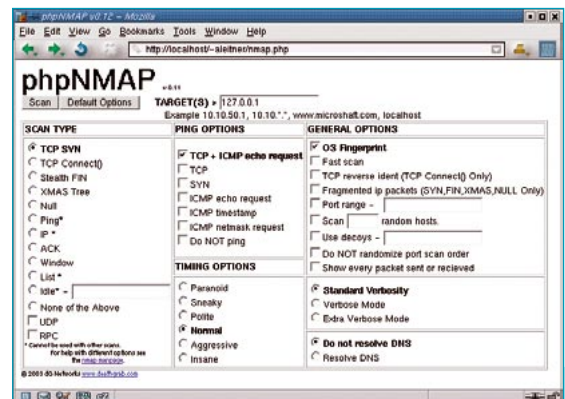
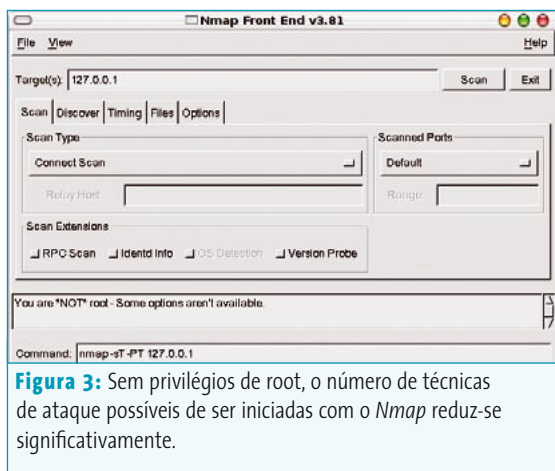


Figura 2: Mesmo sem oferecer um grande número de opções, a interface web para o Nmap, escrita em PHP, é um recurso bastante interessante.



**Figura 3:** Sem privilégios de root, o número de técnicas de ataque possíveis de ser iniciadas com o Nmap reduz-se significativamente.

– Request For Comments, uma espécie de documento normativo).

Analisando a maneira como o alvo responde a esses pacotes "Frankenstein", o Nmap tira conclusões a respeito de suas características e vulnerabilidades. Para a maioria das técnicas, é necessário rodar o Nmap como root, já que o programa trabalha com os chamados *raw sockets* para poder montar seus próprios pacotes adulterados – solenemente esnobando a pilha *TCP/IP* do sistema operacional.

O Nmap é de grande valia para os administradores de sistemas de informação. Você pode varrer todas as portas de uma máquina à cata de problemas. Mais ainda: se necessário, pode varrer redes inteiras! A varredura do tipo *ping* é bastante útil nessas horas. Como o nome sugere, o comando *ping* gera uma mensagem *ICMP Echo Request* e a envia à máquina alvo. Se o alvo existir, estiver ligado e conectado, irá responder com *ICMP Echo Reply*. A varredura *ping* do Nmap funciona de forma semelhante.

O protocolo *ICMP* não usa nenhuma porta, portanto o Nmap não pode usá-lo para investigar muita coisa. Por outro lado, uma varredura desse tipo usa apenas um par de pacotes para cada nó – tornando-a bem rápida. O resultado, entretanto, é impreciso: se não receber-

mos uma resposta, não há como determinar se o alvo está desconectado, se a resposta foi bloqueada por um firewall ou se, simplesmente, o sistema operacional do alvo não responde a pings.

O ping serve apenas como investigação preliminar, dando ao pesquisador uma maneira rápida de verificar quais sistemas reagem, economizando tempo precioso – afinal,

tarefas mais demoradas ainda estão por vir, portanto é melhor concentrá-las em alvos realmente "vivos".

## Um grama de precaução...

A varredura do tipo "Lista" (*-sL*) permite que os usuários verifiquem todas as configurações do Nmap antes de iniciar um ataque real. Isso pode evitar que erros e esquecimentos embaraçosos ponham a perder toda a confidencialidade do ataque. Uma varredura do tipo "Lista" diz ao usuário quais sistemas o Nmap vai investigar (e de que forma o fará), mas sem disparar um ataque verdadeiro. Em suma, é uma simulação.

Mesmo com essa precaução, ainda há que se ter cuidado redobrado durante testes de penetração. Lembre-se sempre de que o Nmap, por padrão, tenta resolver os nomes DNS das máquinas sob ataque. Certifique-se sempre de desativar esse comportamento denunciador.

## Amante à moda antiga

Se o usuário não tiver privilégios de root (figura 3) o Nmap só consegue fazer varreduras usando conexões *TCP* completas (*connect()*). Essa técnica usa as funções do próprio sistema operacional para estabelecer conexões que atendam aos requisitos da RFC vigente para o *TCP/IP*.

Uma conexão *TCP* completa usa a manjadíssima negociação em três etapas do protocolo *TCP*, conhecida como *three-way handshaking*. Sem essa negociação prévia, é impossível estabelecer um canal de comunicação *TCP* entre dois computadores – normalmente, entre um cliente e um servidor; por exemplo, entre o *Internet Explorer* e o *IIS*.

A negociação funciona da seguinte maneira: o cliente que quer iniciar a conexão (em nosso caso, o Nmap) envia um pacote *TCP* com o sinalizador (ou *flag*) *SYN* ativado. O flag *SYN* avisa ao outro sistema que o primeiro quer se conectar a ele. O pacote *TCP*, além do flag *SYN*, informa também as portas de origem e de destino (figura 4).

Se, no alvo, a porta que está sendo testada estiver aberta (ou seja, aceitando conexões), o alvo responde com outro pacote *TCP*, esse com os flags *SYN* e

**Tabela 2: Estados das portas**

Estado	Explicação
Aberta (Open)	É possível se conectar a essa porta sem restrições.
Filtrada (Filtered)	A porta está, provavelmente, sendo bloqueada por um firewall. Se as varreduras do tipo <i>SYN</i> e Conexão Completa descobrirem portas abertas e filtradas, o administrador do firewall pode ter cometido o grave erro de ter implementado uma regra <i>DROP</i> malfeita.
NÃO-Filtrada (Unfiltered)	As varreduras do tipo <i>ACK</i> ou Janela descobriram portas não filtradas pelo firewall. A comunicação com essas portas é, a princípio, possível, mas será necessário usar nelas outros tipos de varredura para obter mais informações.
Fechada (Closed)	A porta pode estar corretamente bloqueada pelo firewall ou mesmo nem existir no sistema sob ataque. Em ambos os casos, é impossível comunicar-se com essas portas.

### Listagem 1: Varredura por conexão TCP completa

```
Porta fechada:
192.168.5.22 -> 192.168.5.10 TCP 60319 > 80 [SYN]
192.168.5.10 -> 192.168.5.22 TCP 80 > 60319 [RST, ACK]
```

```
Porta aberta:
192.168.5.22 -> 192.168.5.10 TCP 60320 > 80 [SYN]
192.168.5.10 -> 192.168.5.22 TCP 80 > 60320 [SYN, ACK]
192.168.5.22 -> 192.168.5.10 TCP 60320 > 80 [ACK]
192.168.5.22 -> 192.168.5.10 TCP 60320 > 80 [RST, ACK]
```

### Listagem 2: Varredura com pacotes TCP-SYN

```
Porta fechada:
192.168.5.22 -> 192.168.5.10 TCP 56522 > 80 [SYN]
192.168.5.10 -> 192.168.5.22 TCP 80 > 56522 [RST, ACK]
```

```
Porta aberta:
192.168.5.22 -> 192.168.5.10 TCP 60420 > 80 [SYN]
192.168.5.10 -> 192.168.5.22 TCP 80 > 60420 [SYN, ACK]
192.168.5.22 -> 192.168.5.10 TCP 60420 > 80 [RST]
```

ACK ligados. Essa é a segunda etapa da negociação. Na terceira etapa, o Nmap envia um pacote TCP com o flag ACK ativado, o que estabelece definitivamente a conexão.

Se não houver nenhum programa servidor naquela porta (ou seja, se a porta não estiver aceitando conexões) o sistema operacional do alvo responde com um pacote TCP com o flag *RST* ligado, forçando o encerramento da conexão.

Esse é o modo correto de estabelecer uma conexão TCP com um dado sistema remoto. Na varredura do tipo *connect()*, o Nmap usa o sistema operacional para iniciar o three-way handshake. Se conseguir uma conexão, a porta está aberta. Se receber um *RST*, a porta está fechada. Simples assim.

Observe que o Nmap manda, ele mesmo, um *RST* para finalizar, o mais rápido possível, a conexão que acabou de abrir, caso esta tenha êxito. Deixar conexões abertas por muito tempo torna muito fácil detectar o ataque. Veja a [listagem 1](#).

## Conexão "meia-boca"

Uma conexão completa, mesmo se fechada imediatamente com aquele quarto pacote *RST*, tem uma desvantagem gritante: conexões perfeitamente estabelecidas figuram nos registros (*logs*) do sistema operacional. Como resultado, é fácil determinar quem anda varrendo determinada rede. Dá para melhorar bastante o quadro se você tiver privilégios de *root*.

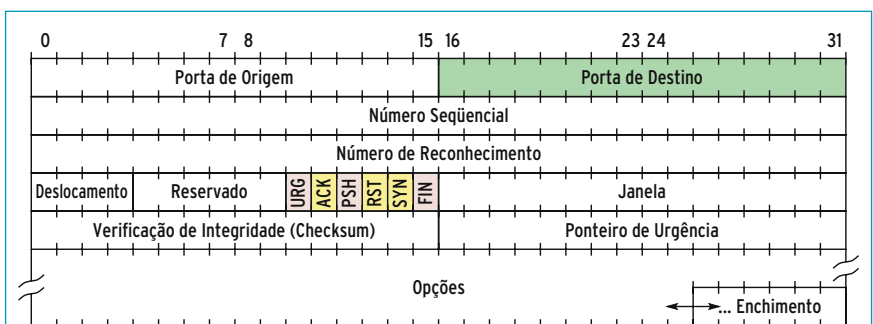
Como *root*, é possível varrer os alvos com a técnica *TCP-SYN*. Além de economizar recursos da rede e do sistema operacional, não depende deste último para nada e possui uma "camuflagem" natural. Em vez de estabelecer uma conexão TCP completa, o Nmap transmite o primeiro *SYN* do three-way handshake. Uma porta fechada reagirá com um *RST* para forçar o cancelamento da conexão. Uma porta aberta responderá com um *SYN/ACK*. Esse *SYN/ACK* já é suficiente para o

Nmap perceber que a porta está aberta. Portanto, em vez de continuar com a conexão, o Nmap manda um *RST* para acabar com a festa – veja a [listagem 2](#). O sistema sob ataque nem percebe esse processo, já que a conexão não se concretizou; de quebra, nada é registrado nos logs.

Apesar de lograr o sistema operacional, esse truque simples não passa a perna em sistemas de detecção de intrusos (*IDS*) como o *Snort* [8] e o *Prelude* [9]. Esses *IDS*s monitoram todas as portas e, se detectarem um grande número de tentativas de conexão frustradas, identificam o processo como uma varredura maliciosa. O *Snort* registra as seguintes informações depois de um portscan como esse:

```
[**] [100:1:1] spp_portscan: PORTSCAN 2
DETECTED from 192.168.5.22 (THRESHOLD 4 2
connections exceeded in 0 seconds) [**]
10/05-19:40:49.540435
```

Para despistar a maioria dos *IDS*s (mas não todos), basta dizer ao Nmap para "segurar a onda" e fazer tudo bem devagar. A idéia é esconder os pacotes de teste na multidão de pacotes do tráfego normal da rede. Também é possível tentar embaralhar a vista do *IDS* com técnicas como *TCP FIN*, "Árvore de Natal" e "TCP Vazio". ➔



**Figura 4:** O Nmap usa os campos do cabeçalho *TCP* para descobrir os detalhes dos sistemas sob ataque. Além dos flags *ACK*, *RST* e *SYN* (em amarelo), algumas técnicas usam combinações pouco usuais dos outros flags (em rosa).

## Segredo de estado

As técnicas TCP-FIN, Árvore de Natal e TCP Vazio possuem uma capacidade de camuflagem sem precedentes. Ao contrário das duas técnicas anteriores (TCP completa e TCP-SYN), estas não iniciam conexão alguma. Em vez disso, mandam um único pacote ao sistema alvo. Os três ataques diferem apenas nos flags TCP que ativam (veja a [figura 4](#)). Nenhum desses flags deveria estar ativo em tráfego normal de rede e, por outro lado, nenhum deles possui o necessário flag SYN para iniciar uma conexão. É pela resposta a esses pacotes (ou falta dela) que o Nmap tenta inferir a disponibilidade do sistema alvo. Quando a porta está fechada, qualquer sistema operacional que atende às normas e RFCs pertinentes responderá com um pacote RST, forçando o encerramento da conexão.

Por outro lado, se a porta estiver aberta (ou seja, há um serviço qualquer "escutando" nela), o sistema sob ataque não saberá o que fazer para responder a esse pacote "alienígena", já que não há uma conexão previamente estabelecida. Infelizmente, as RFCs não dão instruções claras sobre como responder a pacotes dessa natureza. Como consequência, cada sistema operacional se comporta de um jeito diferente. A [listagem 3](#) mostra como um sistema Linux responde a eles: simplesmente ignorando-os.

Sistemas Windows lidam com isso de forma bastante diversa. Eles respondem a esse tipo de pacote com um RST. Como essa é a mesma resposta tanto para portas abertas quanto para portas fechadas, não é possível discernir entre uma e outra. Pode parecer a forma correta de se fazer a coisa, mas isso traz um efeito colateral: só máquinas Windows têm esse comportamento, portanto essa é uma maneira fácil de identificar se o sistema operacional rodando no alvo veio de Redmond.

## Listagem 3: FIN, Árvore de Natal e Vazio

```

FIN, Porta fechada:
192.168.5.22 -> 192.168.5.10 TCP 56485 > 80 [FIN]
192.168.5.10 -> 192.168.5.22 TCP 80 > 56485 [RST, ACK]

FIN, Porta aberta:
192.168.5.22 -> 192.168.5.10 TCP 43406 > 80 [FIN]
192.168.5.22 -> 192.168.5.10 TCP 43407 > 80 [FIN]

Árvore de Natal, Porta fechada:
192.168.5.22 -> 192.168.5.10 TCP 49499 > 80 [FIN, PSH, URG]
192.168.5.10 -> 192.168.5.22 TCP 80 > 49499 [RST, ACK]

Árvore de Natal, Porta aberta:
192.168.5.22 -> 192.168.5.10 TCP 47109 > 80 [FIN, PSH, URG]
192.168.5.22 -> 192.168.5.10 TCP 47110 > 80 [FIN, PSH, URG]

Vazio, Porta fechada:
192.168.5.22 -> 192.168.5.10 TCP 50508 > 80 []
192.168.5.10 -> 192.168.5.22 TCP 80 > 50508 [RST, ACK]

Vazio, Porta aberta:
192.168.5.22 -> 192.168.5.10 TCP 55971 > 80 []
192.168.5.22 -> 192.168.5.10 TCP 55972 > 80 []

```

Se o alvo não responder, o Nmap rotula aquela porta como fechada ou filtrada ([tabela 2](#)). Os firewalls tendem a descartar pacotes desse tipo sem emitir qualquer comentário.

Outras técnicas podem oferecer muito mais informações sobre o sistema sob ataque. Uma delas é a "Detecção de Versão". Mas cuidado: essa técnica não tenta, de forma alguma, se esconder sob qualquer camuflagem e usa rotinas de identificação de portas bastante agressivas. Se seu objetivo é não ser descoberto, cuidado com ela!

## Até o osso

A detecção de versões não procura por portas abertas. Em vez disso, faz testes numa porta em que, já se sabe de antemão, existe algum serviço "na escuta" ([listagem 4](#)). Os potenciais candidatos a isso foram identificados anteriormente por varredura do tipo TCP-SYN. A detecção de versão abre uma conexão normal com uma porta e se comunica

com o serviço lá presente – cuidado, isso cria uma linha no arquivo de log. Pelo canal de comunicação, envia uma série de pacotes de teste e, baseado nas reações a eles, tenta identificar o software responsável pelo serviço e sua versão. O Nmap armazena os resultados em seu banco de dados de ataques sob o título *nmap-service-probes*. A versão 3.93 inclui 2895 assinaturas de serviços.

O Nmap pode empregar essas mesmas técnicas na detecção de todo o sistema alvo – incluindo o sistema operacional. O chamado *OS Fingerprinting* (impressão digital do sistema operacional) é um dos recursos mais espetaculares do Nmap. Para falar a verdade, o Nmap é o mestre supremo nesse terreno, batendo qualquer outro programa, tanto livre quanto comercial.

O processo de descoberta do sistema operacional leva em conta as sutis diferenças no comportamento da pilha TCP/IP. O Nmap possui uma tabela com a maneira como cada sistema operacional

reage a determinados estímulos (isto é, como responde a determinados pacotes). Comparando os resultados dos testes com essa tabela, é possível determinar qual sistema operacional está instalado no alvo. O Nmap 3.93 (a versão mais recente à época desta edição) possui não menos do que 1707 “impressões digitais”. O processo todo é bastante discreto e difícil de ser detectado: nenhuma conexão é iniciada entre o Nmap e o alvo e são usados, ao todo, apenas 30 pacotes bastante simples.

O OS fingerprinting começa com um varredura de portas comum para determinar se há portas abertas no sistema alvo. Depois, dispara uma bateria de oito testes simples, que enviam pacotes especialmente criados para provocar respostas específicas. Alguns desses pacotes nunca ocorreriam em tráfego normal de uma

rede típica – e, portanto, o processo é presa fácil para IDSs. Mas, se não houver um IDS, o sistema sendo testado nem percebe que está sendo escaneado. Ao mesmo tempo em que esses oito testes são disparados, os pacotes são analisados e a opção TCP timestamp determina o tempo de ativação (uptime) do alvo.

Se o Nmap não conseguir identificar o sistema alvo, todos os dados dos testes são

apresentados. Com isso, o próprio usuário pode analisar e tentar descobrir o que é que roda no alvo. Por outro lado, se o usuário já souber de antemão o SO do alvo mas o Nmap não for capaz de identificá-lo, ele ou ela pode publicar a assinatura encontrada na página apropriada do site oficial [10]. Com isso, os usuários do Nmap podem ajudar o programa a identificar cada vez mais sistemas operacionais diferentes. ➔

#### Listagem 4: Detecção de versões

```
192.168.5.22 -> 192.168.5.10 TCP 59555 > 80 [SYN]
192.168.5.10 -> 192.168.5.22 TCP 80 > 59555 [SYN, ACK]
192.168.5.22 -> 192.168.5.10 TCP 59555 > 80 [ACK]
192.168.5.22 -> 192.168.5.10 HTTP GET / HTTP/1.0
192.168.5.3 -> 192.168.5.22 TCP 80 > 59555 [ACK]
192.168.5.3 -> 192.168.5.22 HTTP HTTP/1.0 200 Ok
192.168.5.22 -> 192.168.5.10 TCP 59555 > 80 [ACK]
192.168.5.22 -> 192.168.5.10 TCP 59555 > 80 [FIN, ACK]
192.168.5.3 -> 192.168.5.22 HTTP Continuation or non-HTTP traffic
192.168.5.22 -> 192.168.5.10 TCP 59555 > 80 [RST]
```

**Simples**   
Consultoria

**Tecnologia + Humana**

**Usabilidade**  
**Arquitetura da Informação**  
**Acessibilidade**  
**Gerenciamento de Conteúdo**  
**Treinamentos Especializados Zope e Plone**

[www.simplesconsultoria.com.br](http://www.simplesconsultoria.com.br)

## Sem filtros

Há situações em que o Nmap não consegue distinguir entre uma porta filtrada e uma porta aberta (veja a [tabela 2](#)). Para casos como esses, as varreduras do tipo ACK são uma mão na roda. A varredura com pacotes ACK é bastante simples e deveras camuflada. Ela não consegue detectar se uma porta está aberta ou fechada, mas serve a um outro propósito muito especial: detecta a presença de um firewall e pode até ser usada para pesquisar as regras de filtragem desse firewall.

Para conseguir tal proeza, o Nmap transmite, em direção a uma porta qualquer do alvo, um único pacote TCP com o flag ACK ativado. Se não houver um firewall entre o Nmap e o alvo (ou se ele existir mas estiver configurado para deixar essa porta aberta), o alvo deve responder com um pacote RST. Se a resposta for um *ICMP Destination Unreachable*, podemos ter certeza de que existe um firewall entre o Nmap e o alvo e que ele está bloqueando essa porta – ou seja, a porta está filtrada.

O Nmap também executa as chamadas “Varreduras de Janela” (*Window Scans*), uma variação da varredura por ACK, mas que também descobre portas abertas. Nesse tipo de varredura, o Nmap também envia um pacote ACK mas, entre outras coisas, analisa o tamanho da janela TCP (um dos campos mais importantes do protocolo TCP)

ajustado pelo alvo quando este responde. Pacotes RST com um tamanho de janela nulo (zero bytes) indicam que a porta está aberta.

O número de sistemas operacionais que respondem a essa técnica é bastante pequeno [*X-WInScan*] e vai continuar diminuindo ao longo dos anos. Mesmo assim, a varredura de janela pode ser de grande valia para obter informações importantes sobre a plataforma sob ataque. Isso torna especialmente interessante se precisarmos de mais informações a respeito de sistemas reconhecidamente seguros.

## Só TCP não basta

Além do protocolo TCP, o Nmap também faz das suas com o UDP. Há realmente poucas opções de varredura com o UDP, já que esse protocolo é por demais simplório e não tem nada parecido com os flags de controle do TCP.

Se por um lado poucos dados podem ser obtidos a partir disso, por outro o processo de varredura é bem simples. Quando a porta está fechada, o alvo responde com um *ICMP Port Unreachable*. Quando está aberta, pode ser que o alvo envie algum tipo de dado ou, o que é mais provável, fique na mais profunda mudez. O silêncio esconde o estado real da porta: o Nmap a classifica como aberta, embora isso possa ser um falso positivo já que a porta pode estar filtrada (veja a [tabela 2](#)). Em caso de dúvida, a

detecção de versão pode ajudar na supressão da ambigüidade.

Muitos sistemas limitam o envio de mensagens ICMP a um determinado número delas por segundo. O Nmap é esperto o bastante para perceber esse comportamento e reduzir a velocidade, engatando a segunda ou mesmo a primeira marcha. Por isso, varreduras no protocolo UDP podem demorar um pouco para terminar.

## Baixo nível

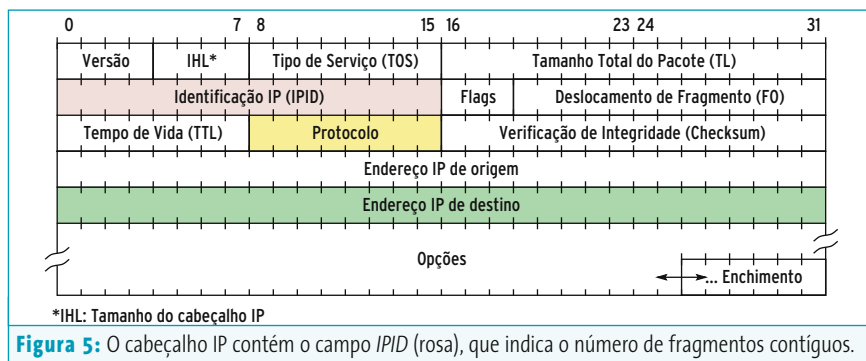
O Nmap possui um modo de varredura exclusivo para o protocolo IP. Mas ele não serve para identificar portas abertas ou filtradas. Em vez disso, simplesmente informa ao usuário quais protocolos de camada 4 o alvo reconhece e consegue trabalhar. Se o alvo for uma máquina com Linux, o Nmap provavelmente vai descobrir *ICMP* (Internet Control Message Protocol), *IGMP* (Internet Group Multicast Protocol), TCP, UDP e *IPv6* (para túneis IPv6-sobre-IPv4). Para determinar esses protocolos, o programa testa todos os números de protocolo de 1 a 255 e espera pelas respostas.

Essa informação permite ainda que o Nmap, além de determinar os protocolos suportados, faça uma espécie de detecção improvisada do sistema operacional. Por exemplo, apenas roteadores e servidores especiais usam o protocolo *Virtual Router Redundancy Protocol* (VRRP) ou a alternativa livre, o *CARP*.

## Chamando todos os carros!

As Varreduras de RPC são capazes de identificar serviços (e portas) especiais como NFS e NIS, que dependem da tecnologia RPC. Essa técnica de varredura só vale a pena como complemento das anteriores e ativa automaticamente a detecção de versões.

Para descobrir serviços RPC escondidos, a técnica usa a instrução especial



**Figura 5:** O cabeçalho IP contém o campo *IPID* (rosa), que indica o número de fragmentos contíguos.

`PROC=0`. Ela não pede ação alguma ao serviço RPC, mas força-o a revelar sua existência – em outras palavras, "cutuca para que saia da moita". Serviços não-RPC não reconhecem a instrução e não respondem. Como o processo todo depende da interação com um aplicativo rodando no alvo, não é nada discreto. Por outro lado, com um pouco de sorte é possível obter muitas dicas do sistema sob investigação.

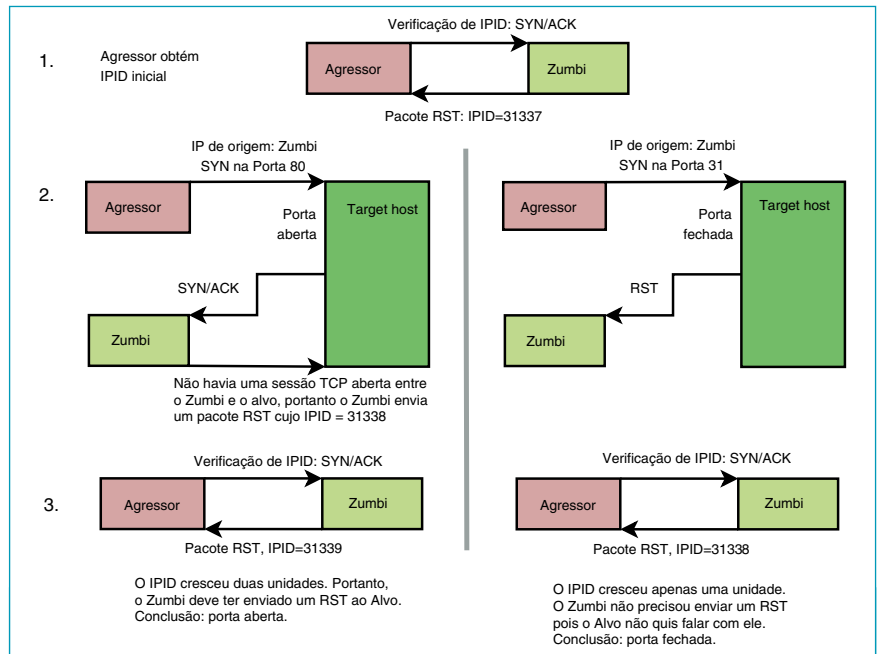
## Arenques vermelhos

Mesmo as técnicas de varredura mais discretas e camufladas do Nmap deixam muitos rastros. Em um teste de penetração, o auditor pode precisar de detalhes que só uma varredura "barulhenta" tem condições de revelar.

Você certamente será detectado. A única coisa que pode fazer, então, é tentar evitar a sua identificação. Uma das maneiras de fazer isso é inundar a rede sob ataque com pacotes "impostores". Isso vai abarrotar os registros dos IDSs com informações falsas e dará bastante trabalho aos administradores de segurança.

Isso é o que se chama *decoy* – em português, chamariz, isca, engodo, armadilha... Quando o Nmap usa um decoy, simula um sem-número de varreduras simultâneas. O pulo do gato: cada varredura vem de um IP diferente! Obviamente, todos os IPs são falsos (menos o seu...) e isso dilui a possibilidade de determinação da origem do ataque. Não tenha dúvida alguma, o procedimento será detectado – afinal, a rede do alvo sofrerá uma avalanche de varreduras. Mas é exatamente o ruído feito por esse tropel que torna muito mais difícil identificar o legítimo agressor.

Do ponto de vista do administrador, usar decoys é um excelente teste do desempenho dos nós da rede, do firewall e dos sistemas de detecção de intrusos (IDS). Por padrão, o Nmap pode simular



**Figura 6:** A varredura passiva é uma técnica bastante engenhosa que usa um "laranja" como intermediário ou "zumbi". Os pacotes enviados ao alvo originam-se, na realidade, do zumbi. O agressor tira conclusões a partir da progressão do número guardado no campo IPID do cabeçalho IP.

até 128 falsas varreduras simultâneas. Se sua máquina for parruda, e sua conexão rápida o bastante, pode aumentar o valor da constante `MAX_DECOYS` no arquivo `nmap.h` – parte do código fonte do Nmap – e recompilá-lo.

## À meia-noite encarnarei no teu cadáver

As "Varreduras Passivas" (*Idle scans*) tentam descobrir as portas abertas do alvo usando uma terceira máquina, que funciona como "laranja". Não há troca de pacotes entre a máquina que roda o Nmap e o alvo, portanto o auditor (ou o "hacker") está a salvo. O Nmap faz uso de um truque bastante engenhoso de impostura de IPs (mais conhecido como *spoofing*) para fazer o "laranja" (chamado de "zumbi") ricochetear pacotes vindos do Nmap em direção ao alvo. Para que o truque funcione, algumas condições devem ser satisfeitas:

⇒ O "zumbi" (algumas vezes – e sarcasticamente – chamado de "proxy") deve estar ativo e conectado, mas deve ter

pouquíssimo tráfego – quanto menos, melhor, sendo nenhum tráfego a situação ideal.

⇒ O IPID dos zumbis precisa ser facilmente previsível. Para servir como zumbi, o "laranja" escolhido deve aumentar o valor do IPID em uma unidade para cada novo pacote. O baixo tráfego (condição anterior) é vital para garantir que o IPID (campo de identificação do cabeçalho IP) não seja tão perturbado – veja a **figura 5**.

O próprio Nmap é capaz de identificar os candidatos a zumbi mais apropriados. Ele manda seis pacotes SYN/ACK para o laranja e verifica os IPIDs dos pacotes RST que voltam em resposta. Se o laranja escolhido não servir, o Nmap interrompe o processo com a mensagem:

```
Idlescan is unable to obtain meaningful results from proxy 192.168.5.99 (192.168.5.99).
I'm sorry it didn't work out.
QUITTING!
```

Se os IPIDs progredirem de forma previsível, o Nmap repete o processo por mais quatro vezes, usando pacotes cujo endereço de origem pertence ao sistema a ser investigado – ou seja, o alvo. Com isso, o zumbi manda pacotes RST em resposta a esse estímulo, mas como o endereço de origem está forjado, essas respostas são enviadas para o alvo, não para o computador que está rodando o Nmap. Para obter os resultados, o Nmap manda um pacote SYN/ACK adicional, mas dessa vez com seu próprio endereço de origem. O zumbi só servirá para intermediar uma varredura passiva se o IPID no RST seguinte for cinco unidades maior do que o pacote original.

## Visão além do alcance

Toda a preparação que vimos até aqui tinha o único propósito de verificar se o zumbi era idôneo. A partir de agora, começa a segunda fase do processo: executar a varredura passiva propriamente dita. O Nmap usa uma abordagem semelhante para investigar o alvo: manda para ele pacotes SYN, forjando um pacote cujo endereço de origem é o endereço IP do zumbi. O alvo vai responder, obviamente, para o zumbi.

É aí que está toda a graça do processo. Se a porta sendo testada no alvo estiver fechada, o alvo vai mandar um RST para o zumbi, que o ignora solenemente. Se, pelo contrário, a porta no alvo estiver aberta, esta iniciará a segunda etapa da negociação (o *three-way handshake*) respondendo com um pacote SYN/ACK. O zumbi não sabe de nada a respeito dessa tentativa de conexão e responde com um RST – dessa forma incrementando seu próprio IPID.

A lógica, então, é simples: se o IPID do "laranja" não se alterar desordenadamente, é porque recebeu um RST do alvo – portanto a porta no alvo está fechada. Já se o IPID do zumbi for perturbado, é

porque recebeu um SYN/ACK do alvo e teve de responder com um RST – provando que a porta no alvo está aberta. O Nmap precisa, então, enviar um pacote de teste ao zumbi para monitorar a perturbação do IPID – veja a figura 6 para mais detalhes.

Para acelerar o processo, o Nmap considera de antemão que a maioria das portas estará fechada. Ele inicia o teste em 30 portas TCP escolhidas aleatoriamente e manda pacotes SYN a cada uma delas. Se o IPID aumentar, o Nmap infere a quantidade de portas abertas no alvo. Em uma próxima etapa, o programa reduz o número de portas aleatórias até identificar os números das que estão realmente abertas.

## Exemplos

Os exemplos que preparamos a seguir foram especialmente criados para mostrar que o Nmap não é uma ferramenta voltada para malfetores, mas sim um auxiliar e tanto para os atarefadísimos administradores de rede de hoje em dia. Antes de colocar a mão na massa, recomendamos uma pitada de precaução: truques e experimentos com protocolos não são um mar calmo e sereno. Cuidado com as minas submarinas. É possível que justo aquele seu servidor importante apresente comportamento inesperado. É claro que tirar do ar um sistema fundamental é uma maneira enfática de mostrar a seu chefe as vulnerabilidades existentes, mas pode custar seu emprego. Imagine, por exemplo, que seus testes paralisem o sistema de Voz sobre IP da empresa, deixando todas as filiais sem telefone. Catastrófico, não acha?

Para testes de rotina, escreva em um arquivo de texto os números IP de seus sistemas mais críticos e especifique, com a opção `--excludefile`, esse arquivo. Isso permitirá varrer toda a sua rede sem colocar em risco seu contracheque.

O Nmap também possui funções para geração de arquivos de registro, seja para documentação ou para comparar duas varreduras. A opção `-oA` permite três formatos de saída, permitindo a análise tanto manual como automática dos dados. O *NDiff* [11] é bastante útil para comparar dados de diferentes varreduras.

## Problemas com licenças

Empresas com filiais distantes podem não ter (e, normalmente, nunca têm...) pessoal técnico em todos os escritórios. O Nmap pode dar uma mãozinha nesse detalhe também, monitorando tudo o que acontece nesses locais. Isso é desejável não só pelo aspecto da segurança: verificar os programas que estão instalados em cada máquina pode prevenir muitas dores de cabeça com programas não autorizados (ou mesmo piratas) instalados pelos usuários.

Uma varredura simples com ping lista as máquinas da filial; técnicas mais complexas recolhem informações sobre as versões e atualizações dos programas e ajuda a identificar os funcionários que não respeitam as regras a respeito de software não autorizado:

```
nmap -vv -sS -O -T Polite -n -oA 2
filiais
192.168.6.0/24
```

O comando mostrado inicia uma varredura SYN `-sS` para investigar toda a rede 192.168.6.0 (classe C) e coletar informações a respeito dos sistemas operacionais (`-O`). Como não queremos consultas ao DNS, usamos a opção `-n`. Para economizar a banda do link com a filial, usamos `-T Polite` para diminuir a cadência dos testes. `-oA filiais` registra dados mais detalhados nos logs. O Nmap possui três formatos de registro: o mais legível para humanos é o `filiais.nmap`; há ainda o `filiais.gnmap`, fácil de ser esmiuçado

com a ferramenta `grep`. Por fim, o arquivo XML `filiais.xml` é mais palatável para softwares interpretadores.

## Epidemia

Infestação por vermes de Internet ("worms") são uma ocorrência bastante comum em redes com muitos computadores rodando Windows. Ferramentas de espionagem (spyware) e controle remoto como o `BackOrifice` ainda são bastante comuns. O Nmap pode auxiliar administradores criativos a improvisar um "spybot de pobre", testando as portas normalmente usadas por programas maliciosos conhecidos:

```
nmap -vv -sS -n --excludefile excecoes.2
txt -p wormports -oA infectados
192.168.5.0/24
```

O comando ativa o modo de detalhamento máximo `-vv`. Uma lista de portas conhecidas e potencialmente perigosas é indicada com o parâmetro `-p`. Os vermes usam essas portas para obedecer a comandos de seus mestres, baixar código malicioso ou se auto-propagarem. As máquinas na rede 192.168.5.0/24 são testadas com uma varredura SYN (`-sS`), mas sem resolução de nomes `-n`. Os computadores cujos endereços estão no arquivo `excecoes.txt` serão poupados do teste. Os resultados serão guardados nos arquivos `infectados.nmap`, `infectados.gnmap` e `infectados.xml`.

O recurso de detecção da versão dos daemons (`-sV`) faz um trabalho muito bem feito ao detectar quais softwares maliciosos, spywares, cavalos de tróia, vermes e vírus estão "na escuta" nas portas mais comuns. O Nmap consulta seu banco de dados interno para identificar os serviços em questão. Entretanto, o processo é demoradíssimo e pode representar um gargalo em sua rede – o que o torna impraticável dependendo do caso.

## Remendos

Vermes, vírus e outros tipos de software do mal se aproveitam de vulnerabilidades conhecidas publicamente. O *SQL Slammer* é um belo exemplo de verme que ganhou notoriedade por sua capacidade de se alastrar. Mesmo o *OpenSSL*, um servidor reconhecido por sua segurança, ganhou uma publicidade não muito boa graças ao verme *Scalper*. A única maneira de diminuir os riscos é a atualização constante dos sistemas.

Para testar e ter certeza de que não há em sua rede nenhum serviço com essas vulnerabilidades conhecidas, siga as instruções do exemplo anterior. Colabore para com o tráfego de sua rede e divida a detecção das versões dos daemons e a detecção do sistema operacional em duas ações distintas:

```
nmap -vv -sS -A -n --excludefile 2
excecoes.txt -oA vuln_versoes 2
192.168.5.0/24
```

Novamente, a varredura do tipo SYN (`-sS`) vem em nosso auxílio. As opções são mais ou menos as mesmas. A novidade é a opção `-A`, que combina a detecção de versões e a "impressão digital" do sistema operacional. Os resultados dão uma visão geral das máquinas que precisam de atualização ou remendos de segurança. Tenha em mente, entretanto, que muitos patches não atualizam o número da versão do programa. O Nmap não é um substituto para um programa de administração de atualizações genuíno. Use-o apenas como auxiliar ou "quebra-galho".

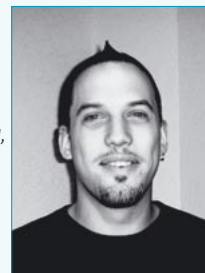
Uma última dica: use um "sniffer" ou programa de captura de tráfego (como o `tcpdump` [13] ou o `Ethereal` [14]) para monitorar como as varreduras funcionam. Reproduza todos os exemplos deste artigo e veja os resultados no sniffer. Diversão garantida.

## Versátil e poderoso

O Nmap consegue lidar com uma quantidade enorme de técnicas de varredura, todas muito sofisticadas e a maioria bastante difícil de ser detectada. O fato de a ferramenta poder ser empregada para fins escusos não deve impedir que os administradores a usem em aplicações legítimas. O uso de scanners é uma forma eficientíssima de obter informações abrangentes e detalhadas sobre sua rede – e o Nmap é estrela maior no reino dessas controvertidas ferramentas. ■

### SOBRE O AUTOR

*Christian Ney é administrador de sistemas em uma empresa aérea regional europeia, mantendo sob seu cajado inúmeras máquinas Unix e alguns firewalls. Em seu tempo livre, contribui com dezenas de projetos de código aberto.*



### INFORMAÇÕES

- [1] Nmap: [www.insecure.org/nmap/](http://www.insecure.org/nmap/)
- [2] Fyodor, "A arte da varredura de portas", Phrack 51: [www.phrack.org/phrack/51/P51-11](http://www.phrack.org/phrack/51/P51-11)
- [3] Strobe: [ftp.surfnet.nl/security/coast/scanners/strobe/](http://ftp.surfnet.nl/security/coast/scanners/strobe/)
- [4] Pscan: [www.packetstormsecurity.com/UNIX/scanners/pscan.c](http://www.packetstormsecurity.com/UNIX/scanners/pscan.c)
- [5] Nmapwin: [nmapwin.sourceforge.net](http://nmapwin.sourceforge.net)
- [6] PHP-Nmap: [phpnmap.sourceforge.net](http://phpnmap.sourceforge.net)
- [7] Projetos derivados do Nmap: [www.insecure.org/nmap/nmap\\_relatedprojects.html](http://www.insecure.org/nmap/nmap_relatedprojects.html)
- [8] Snort IDS: [www.snort.org](http://www.snort.org)
- [9] Prelude IDS: [www.prelude-ids.org](http://www.prelude-ids.org)
- [10] Identificando serviços e portas: [www.insecure.org/cgi-bin/nmap-submit.cgi](http://www.insecure.org/cgi-bin/nmap-submit.cgi)
- [11] NDiff: [www.vinecorp.com/ndiff/](http://www.vinecorp.com/ndiff/)
- [12] OpenSSL: [www.openssl.org](http://www.openssl.org)
- [13] TCPDump: [www.tcpdump.org](http://www.tcpdump.org)
- [14] Ethereal: [www.ethereal.com](http://www.ethereal.com)