

**Faculdade Senac Goiás**  
**Gestão da Tecnologia da Informação**  
**FUNDAMENTOS DE TECNOLOGIA DA INFORMAÇÃO**  
**Professora Orientador: Fernando Pirkel Tsukahara**

**Alunos: Lucas Eduardo Rodrigues Couto**  
**Elson Cristino Farias**  
**Mauro Pacheco Viera**  
**Andre Lima Pereira**

**NMAP**

**Goiânia, 13 de junho 2018**

## **1. O que é o NMAP?**

O Nmap ("Network Mapper") é uma ferramenta de código aberto para exploração de rede e auditoria de segurança. Ela foi desenhada para escanear rapidamente redes amplas, embora também funcione muito bem contra hosts individuais. O Nmap utiliza pacotes IP em estado bruto (raw) de maneira inovadora para determinar quais hosts estão disponíveis na rede, quais serviços (nome da aplicação e versão) os hosts oferecem, quais sistemas operacionais (e versões de SO) eles estão executando, que tipos de filtro de pacotes/firewalls estão em uso, e dezenas de outras características. Embora o Nmap seja normalmente utilizado para auditorias de segurança, muitos administradores de sistemas e rede consideram-no útil para tarefas rotineiras tais como inventário de rede, gerenciamento de serviços de atualização agendados, e monitoramento de host ou disponibilidade de serviço.

A saída do Nmap é uma lista de alvos escaneados, com informações adicionais de cada um dependendo das opções utilizadas. Uma informação chave é a "tabela de portas interessantes". Essa tabela lista o número da porta e o protocolo, o nome do serviço e o estado. O estado pode ser aberto (open), filtrado (filtered), fechado (closed), ou não-filtrado (unfiltered). Aberto (open) significa que uma aplicação na máquina-alvo está escutando as conexões/pacotes naquela porta. Filtrado (filtered) significa que o firewall, filtro ou outro obstáculo de rede está bloqueando a porta de forma que o Nmap não consegue dizer se ela está aberta (open) ou fechada (closed). Portas fechadas (closed) não possuem uma aplicação escutando nelas, embora possam abrir a qualquer instante. Portas são classificadas como não filtradas (unfiltered) quando elas respondem às sondagens do Nmap, mas o Nmap não consegue determinar se as portas estão abertas ou fechadas. O Nmap reporta as combinações aberta|filtrada (open|filtered) e fechada|filtrada (closed|filtered) quando não consegue determinar qual dos dois estados descrevem melhor a porta. A tabela de portas também pode incluir detalhes de versão de software quando a detecção de versão for solicitada. Quando um scan do protocolo IP é solicitado (-sO), o Nmap fornece informações dos protocolos IP suportados ao invés de portas que estejam abertas.

## **2. História do NMAP.**

Um dos primeiros passos em qualquer missão de reconhecimento de uma rede é reduzir um conjunto (às vezes enorme) de faixas de endereços IP, em uma lista de anfitriões(hosts) activos e interessantes. Efectuar o rastreio(scan) de cada porta de cada endereço IP é lento e normalmente desnecessário. É claro que o que torna um anfitrião(host) interessante depende muito do propósito do rastreio(scan). Administradores de rede podem estar apenas interessados em hosts que executam um determinado serviço, enquanto os auditores de segurança podem se importar com cada dispositivo que possuir um endereço IP.

Um administrador pode se sentir à vontade em usar o ping ICMP para localizar os anfitriões(hosts) na rede interna, enquanto um profissional externo de análise de vulnerabilidades (penetration tester) pode utilizar um conjunto diversificado de dezenas de sondagens numa tentativa de enganar as restrições do firewall.

As necessidades para o descobrimento de anfitrião(host) são muito diversas e, por isso, o Nmap oferece uma ampla variedade de opções para customizar as técnicas utilizadas. A descoberta de anfitrião(host) às vezes é chamada de rastreo ping(ping scan), mas ela vai muito além dos simples pacotes ICMP de echo request associados com a popular ferramenta conhecida como ping. Os usuários podem saltar a etapa do ping inteiramente com uma lista de rastreo(scan) (-sL) ou desactivado o ping (-P0), ou enfrentar a rede com combinações arbitrárias de sondagens multi-portas TCP SYN/ACK, UDP e ICMP. O objetivo dessas sondagens é solicitar respostas que mostrem que um endereço IP está realmente activo (é utilizado por um anfitrião(host) ou dispositivo de rede). Em muitas redes, apenas uma pequena percentagem dos endereços IP está activa em um dado momento. Isso é particularmente comum com o espaço de endereçamento privado ao abrigo do RFC1918 como, por exemplo, 10.0.0.0/8. Essa rede tem 16 milhões de IPs, mas eu já a vi sendo utilizado em empresas com menos de mil máquinas. A descoberta de anfitriões(hosts) pode encontrar essas máquinas escassamente alocadas em um mar de endereços IP.

### ***3. NMAP nos filmes.***

Embora o Nmap tenha sido usado em alguns filmes obscuros anteriores, foi o The Matrix Reloaded (Wikipedia, IMDB, Amazon) que realmente transformou o Nmap em uma estrela de cinema! Todos nós já vimos muitos filmes, como Hackers, que passam cenas ridículas em 3D como hackers. Então Fyodor ficou chocado ao descobrir que Trinity faz isso corretamente em The Matrix Reloaded. Precisando hackear a rede elétrica da cidade, ela pega a versão 2.54BETA25 do Nmap, usa-a para encontrar um servidor SSH vulnerável e então explora-a usando a exploração SSH1 CRC32 de 2001. Vergonha da cidade por ser vulnerável (notas de tempo)

### ***4. Principais técnicas de escaneamento de portas:***

Como um novato executando um reparo automotivo, posso brigar por horas tentando usar minhas ferramentas rudimentares (martelo, fita adesiva, grifo, etc.) nas tarefas. Quando eu falho miseravelmente e reboco minha lata-velha para um mecânico de verdade ele invariavelmente pesca aqui e ali em um enorme baú de ferramentas até pegar a coisa perfeita que torna a tarefa uma brincadeira. A arte de escanear portas é similar. Os experts entendem as dezenas de técnicas de escaneamento e escolhem as que são apropriadas (ou uma combinação) para uma dada tarefa. Usuários inexperientes e script kiddies, por outro lado, tentam resolver todos os problemas com o scan SYN padrão. Uma vez que o Nmap é gratuito, a única barreira para a maestria em escaneamento de portas é

o conhecimento. Isso certamente é melhor que no mundo automotivo, onde pode ser necessário uma grande habilidade para determinar que você precisa de um compressor de molas e então você tem que pagar milhares de dólares por um.

A maioria dos tipos de scan está disponível apenas para usuários privilegiados. Isso acontece porque eles enviam e recebem pacotes em estado bruto, o que requer acesso de root em sistemas Unix. Utilizar a conta de administrador no Windows é recomendado, embora o Nmap às vezes funcione com usuários sem privilégios nessa plataforma quando o WinPcap foi carregado no SO. Requerer privilégio de root era uma séria limitação quando o Nmap foi lançado em 1997, pois muitos usuários apenas tinham acesso a contas de shell compartilhadas. Agora o mundo é diferente. Computadores estão mais baratos, muito mais pessoas tem acesso direto e permanente à Internet, e computadores de mesa Unix (incluindo Linux e MAC OS X) são comuns. Uma versão para o Windows do Nmap se encontra disponível atualmente, permitindo que se rode em muito mais computadores de mesa. Por todas essas razões, os usuários têm menos necessidade de executar o Nmap a partir de contas de shell compartilhadas e limitadas. Isso é muito bom pois as opções privilegiadas tornam o Nmap muito mais poderoso e flexível.

#### ***4.1. Escaneamento por TCP SYN scan***

O scan SYN é a opção de scan padrão e mais popular por boas razões. Pode ser executada rapidamente, escaneando milhares de portas por segundo em uma rede rápida, não bloqueada por firewalls intrusivos. O scan SYN é relativamente não-obstrutivo e camuflado, uma vez que ele nunca completa uma conexão TCP. Ele também trabalha contra qualquer pilha TCP padronizada ao invés de depender de idiossincrasias de plataformas específicas como os scans Fin/Null/Xmas, Maimon e Idle fazem. Ele também permite uma diferenciação limpa e confiável entre os estados aberto (open), fechado (closed), e filtrado (filtered).

Esta técnica é frequentemente chamada de escaneamento de porta entreaberta (half-open scanning), porque você não abre uma conexão TCP completamente. Você envia um pacote SYN, como se fosse abrir uma conexão real e então espera uma resposta. Um SYN/ACK indica que a porta está ouvindo (aberta), enquanto um RST (reset) é indicativo de uma não-ouvinte. Se nenhuma resposta é recebida após diversas retransmissões, a porta é marcada como filtrada.

O scan TCP connect é o scan padrão do TCP quando o scan SYN não é uma opção. Esse é o caso quando o usuário não tem privilégios para criar pacotes em estado bruto ou escanear redes IPv6. Ao invés de criar pacotes em estado bruto como a maioria dos outros tipos de scan fazem, o Nmap pede ao sistema operacional para estabelecer uma conexão com a máquina e porta alvos enviando uma chamada de sistema connect(). Essa é a mesma chamada de alto nível que os navegadores da web, clientes P2P, e a maioria das outras

aplicações para rede utilizam para estabelecer uma conexão. É parte da interface de programação conhecida como API de Sockets de Berkeley. Ao invés de ler as respostas em pacotes em estado bruto diretamente dos fios, o Nmap utiliza esta API para obter informações do estado de cada tentativa de conexão.

#### **4.1.1. Exemplos de uso.**

Aqui estão alguns exemplos de utilização do Nmap, desde o simples e rotineiro, até o um pouco mais complexo e esotérico. Alguns endereços IP reais e nomes de domínio foram utilizados para tornar as coisas mais concretas. Nesses lugares você deve substituir os endereços/nomes pelos da *sua própria rede*. Embora eu não ache que o escaneamento de portas de outras redes seja, ou deva ser considerado, ilegal alguns administradores de rede não apreciam o escaneamento não-solicitado de suas redes e podem reclamar. Obter a permissão antecipadamente é a melhor opção.

Para fins de teste, você tem permissão para escanear o host `scanme.nmap.org`. Esta permissão inclui apenas o escaneamento via Nmap e não tentativas de explorar vulnerabilidades ou ataques de negação de serviço (denial of service). Para preservar a banda, por favor não inicie mais do que uma dúzia de scans contra o host por dia. Se esse serviço de alvo livre para escaneamento for abusado, será derrubado e o Nmap irá reportar `Failed to resolve given hostname/IP: scanme.nmap.org`. Essas permissões também se aplicam aos hosts `scanme2.nmap.org`, `scanme3.nmap.org`, e assim por diante, embora esses hosts ainda não existam.

##### **`nmap -v scanme.nmap.org`**

Esta opção escaneia todas as portas TCP reservadas na máquina `scanme.nmap.org`. A opção `-v` habilita o modo verboso (verbose).

##### **`nmap -sS -O scanme.nmap.org/24`**

Inicia um scan SYN camuflado contra cada máquina que estiver ativa das 255 possíveis da rede “classe C” onde o Scanme reside. Ele também tenta determinar qual o sistema operacional que está rodando em cada host ativo. Isto requer privilégio de root por causa do scan SYN e da detecção de SO.

##### **`nmap -sV -p 22,53,110,143,4564 198.116.0-255.1-127`**

Inicia uma enumeração de hosts e um scan TCP na primeira metade de cada uma das 255 sub-redes de 8 bits possíveis na classe B do espaço de endereçamento 198.116. Também testa se os sistemas estão executando `sshd`, `DNS`, `pop3d`, `imapd` ou a porta 4564. Para cada uma destas portas encontradas abertas, a detecção de versão é usada para determinar qual aplicação está executando.

##### **`nmap -v -iR 100000 -P0 -p 80`**

Pede ao Nmap para escolher 100.000 hosts de forma aleatória e escaneá-los procurando por servidores web (porta 80). A enumeração de hosts é desabilitada com -P0 uma vez que enviar primeiramente um par de sondagens para determinar se um hosts está ativo é um desperdício quando se está sondando uma porta em cada host alvo.

**nmap -P0 -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20**

#### ***4.1.2. Captura do escaneamento com WireShark,***

A captura de tráfego em redes pode trazer informações úteis de outras máquinas conectadas nela. Em um pentest interno, podemos simular uma ameaça interna ou um atacante que tenha descoberto uma brecha no perímetro, capturando o tráfego de outros sistemas na rede e pode nos dar informações adicionais interessantes (inclusive usuários e senhas) que podem nos ajudar na fase de exploração. O problema de capturar o tráfego é a quantidade de dados que pode ser produzido. Pouco tempo de captura em uma rede pode inundar o Wireshark com dados que tornam difícil a vida do pentester em descobrir o que realmente é útil. Veremos a seguir como manipular uma rede para pegar acesso ao tráfego que não conseguimos ver.

#### ***4.1.3. Captura do escaneamento com Tcpdump***

[Wireshark](#) é um analisador de protocolo que permite que você capture e navegue interativamente no tráfego de uma rede de computadores em tempo de execução usando a interface de rede do computador.

Este tipo de software, também chamado de Sniffer (ou farejador, em português), é bastante usado por administradores de rede para detectar problemas ou conexões suspeitas, testar se as senhas usadas na rede estão realmente sendo criptografadas e realizar uma série de outras atividades relacionadas a segurança.

Mesmo sendo uma ferramenta altamente técnica, o Wireshark não é tão complicado de usar. Apenas os conceitos envolvidos no processo são voltados para pessoas com conhecimentos profundos de redes.

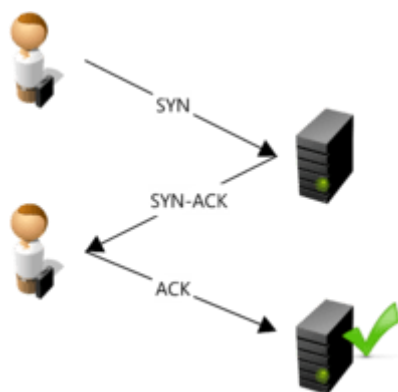
## 4.2. Escaneamento por TCP ACK scan

Começaremos com um SYN scan contra um host. Um SYN scan é um escaneamento TCP que não finaliza o handshake. Uma conexão TCP inicia com um handshake de 3 vias: SYN; SYN-ACK; ACK. Veja abaixo:

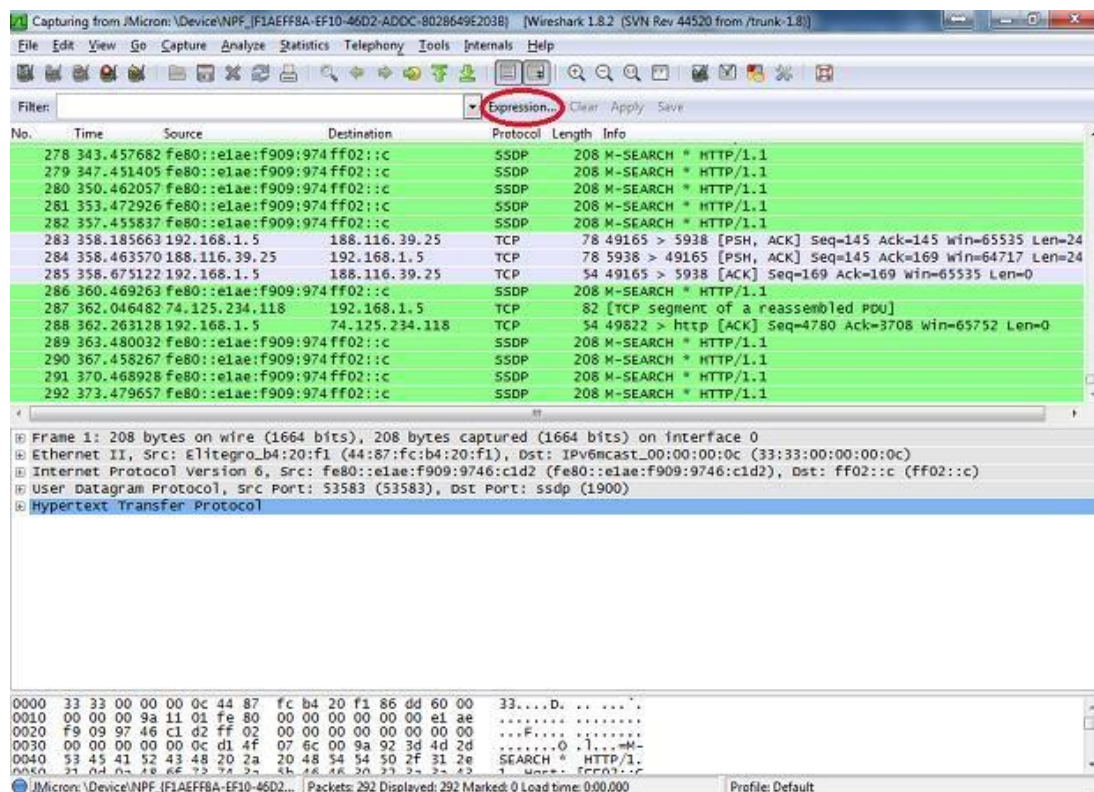
É um SYN scan, o Nmap envia o SYN e espera pelo SYN-ACK se a porta estiver aberta, mas nunca enviará o ACK para completar a conexão. Se o pacote SYN não receber uma resposta SYN-ACK, a porta não está disponível, ou por estar fechada ou a conexão está sendo filtrada. Desta forma, o Nmap verifica se a porta está aberta sem completar a conexão com a máquina alvo. A sintaxe para o SYN scan é com a flag **-sS**

Vejamos um exemplo do uso do SYN scan e ao mesmo tempo vamos incluir a flag **-o** a qual é a opção de output do resultado do Nmap em um arquivo. A opção **-o** diz ao Nmap para logar todo o resultado em alguns formatos, como: .nmap; .gnmap (greppable Nmap) e .xml. O formato .nmap é fácil de visualizar em tela, igual ao resultado obtido durante o scan. A saída do tipo .gnmap (greppable Nmap) é formatado para ser usado com o comando **grep** para buscar informações específicas. XML é um formato padrão usado para importar o resultado em outras ferramentas.

### 4.2.1. Exemplos de uso



### 4.2.2. Captura do escaneamento com Wireshark



### 4.2.3. Captura do escaneamento com Tcpdump

O tcpdump é uma excelente ferramenta para realizar captura e análise de pacotes de rede, recomendada para profissionais que precisem realizar monitoramento e manutenção em uma rede de computadores, além de estudantes que queiram entender a fundo o funcionamento da pilha de protocolos TCP/IP.

#### Opções principais:

##### -i interface

Especificar a interface de rede a partir da qual o tráfego será capturado. Se for usado o parâmetro **any**, o programa irá capturar pacotes em todas as interfaces ativas na máquina.

##### -c num

Sair após capturar num pacotes

##### -D

Mostrar uma lista das interfaces de rede no sistema que estão disponíveis e nas quais o tcpdump pode capturar pacotes.

##### -n

Não realizar a resolução de nomes

##### -r arquivo

Ler os pacotes a partir de arquivo, o qual foi criado anteriormente com a opção -w

##### -w arquivo



Escrever os pacotes em arquivo, em vez de mostrá-los formatados na saída padrão

**-X**

Mostrar o conteúdo do pacote em Hexadecimal e ASCII; se for usado **-XX**, também mostrará o cabeçalho Ethernet.

**src IP/Nome**

Capturar pacotes originados apenas do host com o IP ou hostname especificado.

**dst IP/Nome**

Capturar pacotes destinados apenas ao host com o IP ou hostname especificado.

**greater BYTES, > BYTES**

Capturar apenas pacotes que sejam maiores que BYTES

**less BYTES, < BYTES**

Capturar apenas pacotes que sejam menores que BYTES

**port NUM**

Trabalhar com pacotes destinados ou originados de uma porta específica NUM

**portrange A-B**

Capturar pacotes cujas portas estejam no intervalo entre A e B.

**A and B**

Capturar os pacotes somente se satisfizerem às condições A e B simultaneamente

**A or B**

Capturar os pacotes se satisfizerem à condição A ou à condição B, ou a ambas

**not A**

Capturar os pacotes que não satisfaçam à condição A.

**-v**

Produzir uma saída ligeiramente mais verbosa. Algumas informações adicionais são exibidas, tais como os valores dos campos TTL, Identificação, Comprimento Total e Opções do pacote IP.

**-vv**

Saída mais verbosa do que a opção -v. Por exemplo, pacotes SMB são completamente decodificados.

### **4.3. Escaneamento por TCP FIN scan**

Embora o Nmap tenha crescido em funcionalidade ao longo dos anos, ele começou como um eficiente scanner de portas, e essa permanece sua função principal. O simples comando **nmap <alvo>** escaneia mais de 1660 portas TCP no host <alvo>. Embora muitos scanner de portas tenham tradicionalmente agrupado todas as portas nos estados aberto ou fechado, o Nmap é muito mais granular aberto (open)

Uma aplicação está ativamente aceitando conexões TCP ou pacotes UDP nesta porta. Encontrar esse estado é frequentemente o objetivo principal de um escaneamento de portas. Pessoas conscientes sobre a segurança sabem que cada porta aberta é um convite para um ataque. Invasores e profissionais de avaliação de segurança querem explorar as portas abertas, enquanto os administradores tentam fechar ou proteger com

firewalls sem bloquear usuários legítimos. Portas abertas são também interessantes para scans não-relacionados à segurança pois mostram os serviços disponíveis para utilização na rede. fechado (closed)

Uma porta fechada está acessível (ela recebe e responde a pacotes de sondagens do Nmap), mas não há nenhuma aplicação ouvindo nela. Elas podem ser úteis para mostrar que um host está ativo em um determinado endereço IP (descoberta de hosts, ou scan usando ping), e como parte de uma detecção de SO. Pelo fato de portas fechadas serem alcançáveis, pode valer a pena escanear mais tarde no caso de alguma delas abrir. Os administradores deveriam considerar o bloqueio dessas portas com um firewall. Então elas apareceriam no estado filtrado, discutido a seguir. filtrado (filtered)

O Nmap não consegue determinar se a porta está aberta porque uma filtragem de pacotes impede que as sondagens alcancem a porta. A filtragem poderia ser de um dispositivo firewall dedicado, regras de roteador, ou um software de firewall baseado em host. Essas portas frustram os atacantes pois elas fornecem poucas informações. às vezes elas respondem com mensagens de erro ICMP tais como as do tipo 3 código 13 (destino inalcançável: comunicação proibida administrativamente), mas os filtros que simplesmente descartam pacotes sem responder são bem mais comuns. Isso força o Nmap a tentar diversas vezes só para o caso de a sondagem ter sido descartada por congestionamento da rede ao invés de filtragem. Isso reduz a velocidade do scan dramaticamente. não-filtrado (unfiltered)

O estado não-filtrado significa que uma porta está acessível, mas que o Nmap é incapaz de determinar se ela está aberta ou fechada. Apenas o scan ACK, que é usado para mapear conjuntos de regras de firewall, classifica portas com este estado. Escanear portas não-filtradas com outros tipos de scan, tal como scan Window, scan Syn, ou scan FIN, podem ajudar a responder se a porta está aberta. Open filtrem

O Nmap coloca portas neste estado quando é incapaz de determinar se uma porta está aberta ou filtrada. Isso acontece para tipos de scan onde as portas abertas não dão nenhuma resposta. A falta de resposta também pode significar que um filtro de pacotes descartou a sondagem ou qualquer resposta que ela tenha provocado. Portanto não sabe-se com certeza se a porta está aberta ou se está sendo filtrada. Os scans UDP, IP Protocol, FIN, Null, e Xmas classificam portas desta forma.

#### **4.3.1. Exemplos de uso**

```

root@wks01:/home/vivek# nmap --top-ports 10 192.168.1.1

Starting Nmap 5.00 ( http://nmap.org ) at 2012-11-27 03:30 IST
Interesting ports on 192.168.1.1:
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    open  ssh
23/tcp    closed telnet
25/tcp    closed smtp
80/tcp    open  http
110/tcp   closed pop3
139/tcp   closed netbios-ssn
443/tcp   closed https
445/tcp   closed microsoft-ds
3389/tcp  closed ms-term-serv
MAC Address: BC:AE:C5:C3:16:93 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds

```

#### 4.3.2. Captura do escaneamento com WireShark

Wireshark é um analisador de protocolos de rede de forma gráfica, que nos permite aprofundar em cada pacote que se move em uma rede. Wireshark pode ser usado para capturar pacotes Ethernet, wireless, bluetooth e outros tipos de tráfego. Ele pode decodificar diferentes protocolos que ele vê, então você poderá, por exemplo, reconstruir o áudio de uma ligação Voice over IP (VoIP). Veremos o básico de como capturar e analisar o tráfego com esta ferramenta.

#### 4.3.3. Captura do escaneamento com Tcpdump

O tcpdump é uma excelente ferramenta para realizar captura e análise de pacotes de rede, recomendada para profissionais que precisem realizar monitoramento e manutenção em uma rede de computadores, além de estudantes que queiram entender a fundo o funcionamento da pilha de protocolos TCP/IP.

Ele faz uso da biblioteca libpcap para realizar a captura de pacotes, e existe uma versão da ferramenta para Windows, chamada de WinDump, que usa a biblioteca WinPcap. Neste artigo vamos nos focar no tcpdump em si, usando para isso um sistema Linux (Ubuntu; qualquer outro sistema Linux irá servir para testar os exemplos mostrados).

#### **4.4. Escaneamento por TCP Xmas scan,**

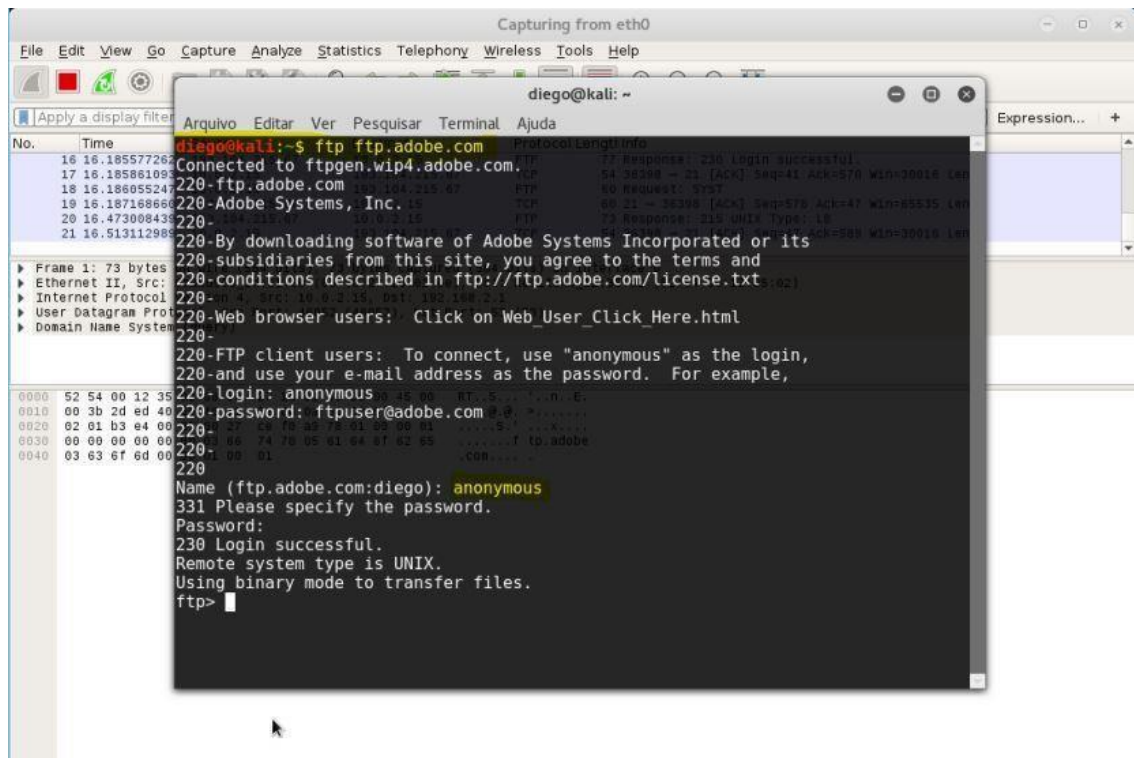
Um invasor usa uma varredura TCP XMAS para determinar se as portas estão fechadas na máquina de destino. Esse tipo de varredura é realizado enviando segmentos TCP com todos os sinalizadores enviados no cabeçalho do pacote, gerando pacotes que são ilegais com base no RFC 793. O comportamento esperado do RFC 793 é que qualquer segmento TCP com um sinalizador fora do estado enviado para uma porta aberta é descartada, enquanto os segmentos com sinalizadores de fora do estado enviados para portas fechadas devem ser manipulados com um RST em resposta. Esse comportamento deve permitir que um invasor verifique portas fechadas enviando certos tipos de pacotes de interrupção de regras (fora de sincronia ou não permitidos pelo TCB) e detecte portas fechadas por meio de pacotes RST. A principal vantagem desse tipo de varredura é a capacidade de varrer firewalls sem estado ou filtros ACL. Esses filtros são configurados para bloquear o acesso a portas geralmente, impedindo pacotes SYN, interrompendo, assim, qualquer tentativa de 'construir' uma conexão. Os pacotes XMAS, como os pacotes FIN ou ACK fora do estado, tendem a passar por tais dispositivos sem serem detectados. Muitos sistemas operacionais, no entanto, não implementam exatamente o RFC 793 e, por esse motivo, as verificações FIN não funcionam como esperado em relação a esses dispositivos. Alguns sistemas operacionais, como o Microsoft Windows, enviam um pacote RST em resposta a qualquer segmento TCP fora de sincronia (ou malformatado) recebido por um soquete de escuta (em vez de descartar o pacote via RFC 793), impedindo assim que um invasor faça a distinção entre portas abertas e fechadas.

##### **4.4.1. Exemplos de uso**

1. Velocidade: A digitalização TCP XMAS é rápida em comparação com outros tipos de digitalizações
2. Furtivo: a varredura TCP XMAS já foi furtiva, mas agora é facilmente detectada por sistemas IDS / IPS
3. Porta Aberta: Detecta uma porta aberta sem resposta ao segmento
4. Porta Fechada: Detecta que um fechado através de um RST recebido em resposta ao FIN
5. Porta Filtrada: Não é possível distinguir entre uma porta filtrada e uma porta aberta
6. Porta não filtrada: Não é possível distinguir entre uma porta não filtrada e uma porta filtrada sem estado

As varreduras de XMAS são limitadas pelo leque de plataformas em que trabalham. Além disso, como as portas abertas são inferidas por meio da não geração de respostas, não é possível distinguir uma porta aberta de uma porta filtrada sem análise adicional. Por exemplo, o escaneamento XMAS de um sistema protegido por um firewall com monitoração de estado pode indicar que todas as portas estejam abertas. Por causa de sua óbvia natureza de quebra de regras, as varreduras de XMAS são marcadas por quase todos os sistemas de prevenção de intrusão ou detecção de intrusão.

#### 4.4.2. Captura do escaneamento com Wireshark



Você poderá ver os pacotes no Wireshark de um sistema com o endereço IP da sua máquina para o servidor ftp da Adobe, e vice versa, onde na coluna **Protocol** está como **FTP**. O Wireshark está capturando o tráfego que está se movendo entre o Kali e o servidor.

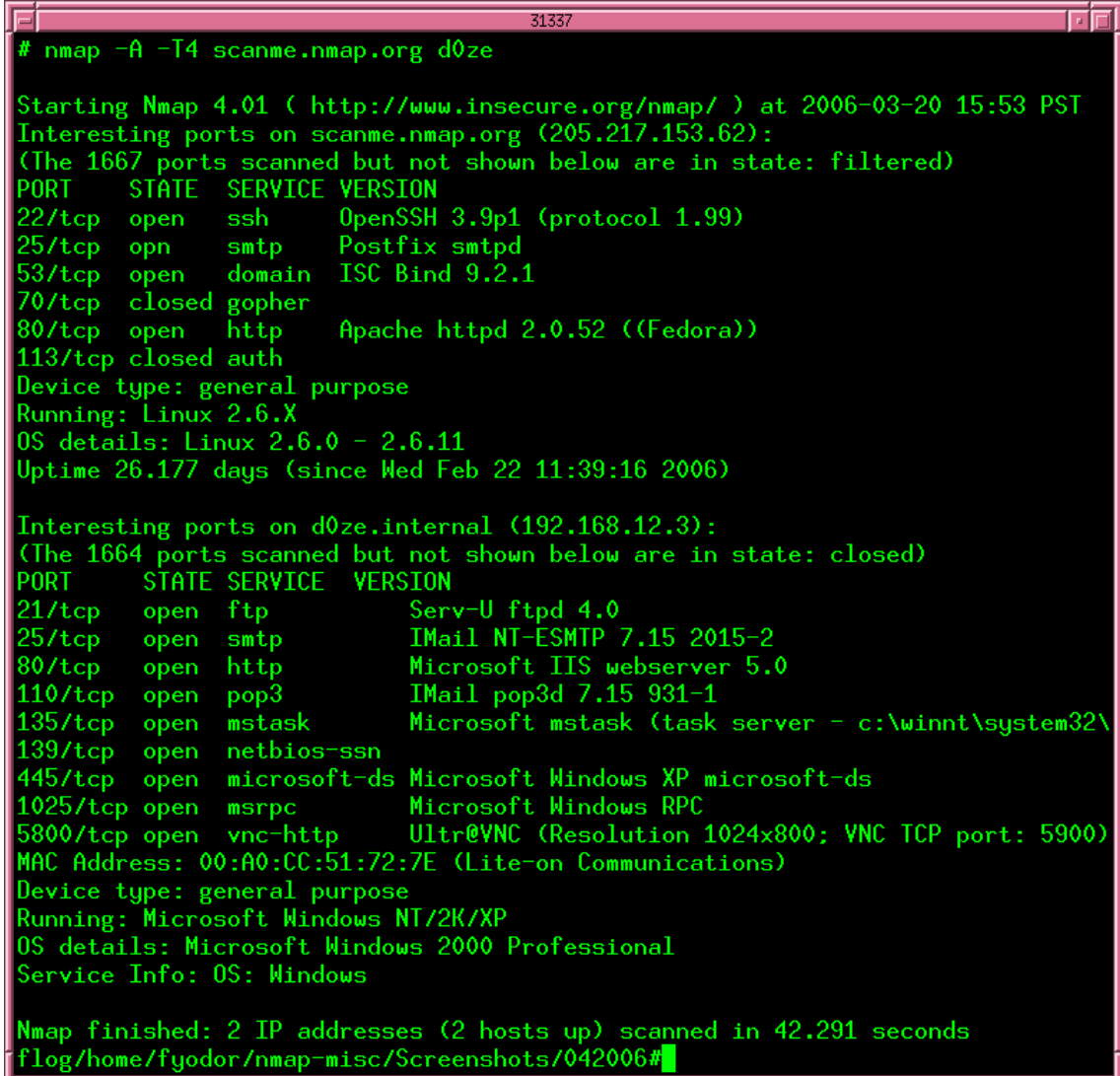
#### 4.4.3. Captura do escaneamento com Tcpdump

```
fabio@fabio-xubuntu:~$ sudo tcpdump -XX -i eth0
[sudo] password for fabio:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
09:52:31.838111 ARP, Request who-has 192.168.1.104 tell 192.168.1.1, length 46
0x0000:  ffff ffff ffff c8d7 19ee 2a17 0806 0001  .....*.....
0x0010:  0800 0604 0001 c8d7 19ee 2a17 c0a8 0101  .....*.....
0x0020:  0000 0000 0000 c0a8 0168 0000 0000 0000  .....h.....
0x0030:  0000 0000 0000 0000 0000 0000  .....
09:52:32.472537 IP 192.168.1.105.23969 > c9060257.virtua.com.br.domain: 62610+ PTR? 104.1.168.192.in-addr.arpa. (44)
0x0000:  c8d7 19ee 2a17 0800 2766 e9e1 0800 4500  ....*...f...E.
0x0010:  0048 7526 4000 4011 3810 c0a8 0169 c906  .Hu&@.@.8...i..
0x0020:  0257 5da1 0035 0034 8db4 f492 0100 0001  .W]..5.4.....
0x0030:  0000 0000 0000 0331 3034 0131 0331 3638  .....104.1.168
0x0040:  0331 3932 0769 6e2d 6164 6472 0461 7270  .192.in-addr.arp
0x0050:  6100 000c 0001  a....
```

O tcpdump é uma excelente ferramenta para realizar captura e análise de pacotes de rede, recomendada para profissionais que precisem realizar monitoramento e manutenção em uma rede de computadores, além de estudantes que queiram entender a fundo o funcionamento da pilha de protocolos TCP/IP.

## 4.5. Escaneamento UDP

### 4.5.1. Exemplos de uso



```
# nmap -A -T4 scanme.nmap.org d0ze

Starting Nmap 4.01 ( http://www.insecure.org/nmap/ ) at 2006-03-20 15:53 PST
Interesting ports on scanme.nmap.org (205.217.153.62):
(The 1667 ports scanned but not shown below are in state: filtered)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 3.9p1 (protocol 1.99)
25/tcp    open  smtp      Postfix smtpd
53/tcp    open  domain    ISC Bind 9.2.1
70/tcp    closed gopher
80/tcp    open  http      Apache httpd 2.0.52 ((Fedora))
113/tcp   closed auth
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.0 - 2.6.11
Uptime 26.177 days (since Wed Feb 22 11:39:16 2006)

Interesting ports on d0ze.internal (192.168.12.3):
(The 1664 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp       Serv-U ftpd 4.0
25/tcp    open  smtp      IMail NT-ESMTP 7.15 2015-2
80/tcp    open  http      Microsoft IIS webserver 5.0
110/tcp   open  pop3      IMail pop3d 7.15 931-1
135/tcp   open  mstask    Microsoft mstask (task server - c:\winnt\system32\
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows XP microsoft-ds
1025/tcp  open  msrpc     Microsoft Windows RPC
5800/tcp  open  vnc-http  Ultr@VNC (Resolution 1024x800; VNC TCP port: 5900)
MAC Address: 00:A0:CC:51:72:7E (Lite-on Communications)
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows 2000 Professional
Service Info: OS: Windows

Nmap finished: 2 IP addresses (2 hosts up) scanned in 42.291 seconds
flog/home/fyodor/nmap-misc/Screenshots/042006#
```

### 4.5.2. Captura do escaneamento com Wireshark

O Wireshark é uma ferramenta muito poderosa que vai muito além de um simples sniffer. O que muitos não sabem é que existem várias formas de se aproveitar o potencial desta ferramenta, mas neste primeiro artigo iremos iniciar do básico. Vamos aprender a sniffar a rede de forma efetiva, criar filtros para buscar apenas a informação que queremos, veremos como um black hat utilizaria esta ferramenta para roubar senhas e para finalizar, como utilizar o Wireshark para diagnosticar problemas de rede ou se um firewall está bloqueando os pacotes corretamente.

Antes de iniciarmos com a prática, é necessário entender o conceito de sniffing. Sniffing, a um grosso modo, seria você ficar com os ouvidos atentos para ouvir qualquer coisa que seja dita ao seu alcance.

Atualmente, quase todos os ambientes utilizam switches e não mais hubs, o que torna o *sniffing* um POUCO mais difícil, pois os switches não enviam os dados para todas as portas como um hub faz, ele envia diretamente para a porta onde se encontra o host de destino; então, se você tentar sniffar uma rede com switch você irá apenas ouvir o que for *broadcast*, ou sua própria conexão. Para conseguir ouvir tudo sem ser o gateway da rede, é necessário um ataque de arp spoof, ou estourar a tabela CAM do switch. Mas isto será tema para um próximo artigo, sendo necessário entender este conceito.

#### **4.5.3. Captura do escaneamento com Tcpdump**

O tcpdump é uma excelente ferramenta para realizar captura e análise de pacotes de rede, recomendada para profissionais que precisem realizar monitoramento e manutenção em uma rede de computadores, além de estudantes que queiram entender a fundo o funcionamento da pilha de protocolos TCP/IP. O site oficial do tcpdump, e também da biblioteca libpcap (sobre o qual ele é baseado). O tcpdump, que é software livre, roda na linha de comandos, estando disponível em diversos sistemas operacionais, como Linux, BSD, OS X, AIX e outros. Ele faz uso da biblioteca libpcap para realizar a captura de pacotes, e existe uma versão da ferramenta para Windows, chamada de WinDump, que usa a biblioteca WinPcap. Neste artigo vamos nos focar no tcpdump em si, usando para isso um sistema Linux (Ubuntu; qualquer outro sistema Linux irá servir para testar os exemplos mostrados).

#### **4.6. Escaneamento de versões dos softwares**

NMAP é uma ferramenta de código aberto para exploração de rede e auditoria de segurança, como mostramos no artigo anterior como fazer um escaneamento de segurança. Ele foi projetado para escanear rapidamente redes de grande porte, embora ele funcione bem em hosts individuais. Nmap utiliza pacotes IP-packets para determinar quais hosts estão disponíveis na rede, quais serviços (nome da aplicação e versão) e o que está aberto (e versões de SO) eles estão executando, que tipo de filtros de pacotes / firewalls estão em uso e dezenas de outras características. Embora o Nmap é comumente utilizado para auditorias de segurança, muitos sistemas e administradores de rede acham que é útil para tarefas de rotina, tais como inventário de rede, horários de atualização gestão de serviços e host de monitoramento ou serviço de uptime.

##### **4.6.1. Exemplos de uso**



```

root@kali:~# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:59:21.022071 IP tiger.lan.60868 > 239.255.255.250:1900: UDP, length 97
23:59:21.022956 IP kali.lan.53726 > dsldevice.lan.domain: 40563+ PTR? 250.255.25
5.239.in-addr.arpa. (46)
23:59:21.258446 IP dsldevice.lan.domain > kali.lan.53726: 40563 NXDomain 0/1/0 (
103)
23:59:21.259028 IP kali.lan.59182 > dsldevice.lan.domain: 8741+ PTR? 73.1.168.19
2.in-addr.arpa. (43)
23:59:21.262315 IP dsldevice.lan.domain > kali.lan.59182: 8741* 1/0/0 PTR tiger
lan. (66)
23:59:21.262701 IP kali.lan.33027 > dsldevice.lan.domain: 51326+ PTR? 254.1.168.
192.in-addr.arpa. (44)
23:59:21.265046 IP dsldevice.lan.domain > kali.lan.33027: 51326* 1/0/0 PTR dslde
vice.lan. (71)
23:59:21.265681 IP kali.lan.53178 > dsldevice.lan.domain: 54535+ PTR? 74.1.168.1
92.in-addr.arpa. (43)
23:59:21.268178 IP dsldevice.lan.domain > kali.lan.53178: 54535* 1/0/0 PTR kali
lan. (65)
^C

```

#### 4.6.2. Captura do escaneamento com WireShark

The screenshot shows the Wireshark 1.8.4 interface. The 'Capture' pane on the left lists available interfaces, including 'Intel DC1140 PCI Fast Ethernet Adapter'. The 'Start' button is highlighted. The 'Display Filter' dialog is open, showing the filter 'eth[0]'. The packet list shows several packets, including a 'ms-wbt-server [ACK]' packet. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.



### 4.6.3. Captura do escaneamento com Tcpdump

```
fabio@fabio-xubuntu:~$ sudo tcpdump -i eth0 port 53
[sudo] password for fabio:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:14:08.834015 IP 192.168.1.105.32729 > c9060257.virtua.com.br.domain: 60227+ A? www.bosontreinamentos.com.br. (
46)
10:14:08.834388 IP 192.168.1.105.59964 > c9060257.virtua.com.br.domain: 59035+ AAAA? www.bosontreinamentos.com.br
. (46)
10:14:08.867312 IP c9060257.virtua.com.br.domain > 192.168.1.105.59964: 59035 1/1/0 CNAME bosontreinamentos.com.br
. (119)
10:14:08.867736 IP c9060257.virtua.com.br.domain > 192.168.1.105.32729: 60227 2/3/3 CNAME bosontreinamentos.com.br
., A 186.202.153.82 (186)
10:14:09.314336 IP 192.168.1.105.25309 > c9060257.virtua.com.br.domain: 45976+ PTR? 87.2.6.201.in-addr.arpa. (41)
10:14:09.330894 IP c9060257.virtua.com.br.domain > 192.168.1.105.25309: 45976 1/2/0 PTR c9060257.virtua.com.br. (
113)
10:14:09.331523 IP 192.168.1.105.26142 > c9060257.virtua.com.br.domain: 12677+ PTR? 105.1.168.192.in-addr.arpa. (
44)
10:14:09.359350 IP c9060257.virtua.com.br.domain > 192.168.1.105.26142: 12677 NXDomain* 0/1/0 (130)
10:14:10.195335 IP 192.168.1.105.44184 > c9060257.virtua.com.br.domain: 25213+ A? fonts.googleapis.com. (38)
10:14:10.206372 IP 192.168.1.105.61246 > c9060257.virtua.com.br.domain: 63139+ A? pagead2.googlesyndication.com.
(47)
10:14:10.231581 IP c9060257.virtua.com.br.domain > 192.168.1.105.61246: 63139 4/4/4 CNAME pagead46.l.doubleclick.
net., A 173.194.118.109, A 173.194.118.121, A 173.194.118.122 (278)
10:14:10.231683 IP c9060257.virtua.com.br.domain > 192.168.1.105.44184: 25213 2/4/4 CNAME googleadapis.l.google.c
```

```
fabio@fabio-xubuntu:~$ sudo tcpdump -i eth0 port 53
[sudo] password for fabio:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:14:08.834015 IP 192.168.1.105.32729 > c9060257.virtua.com.br.domain: 60227+ A? www.bosontreinamentos.com.br. (
46)
10:14:08.834388 IP 192.168.1.105.59964 > c9060257.virtua.com.br.domain: 59035+ AAAA? www.bosontreinamentos.com.br
. (46)
10:14:08.867312 IP c9060257.virtua.com.br.domain > 192.168.1.105.59964: 59035 1/1/0 CNAME bosontreinamentos.com.br
. (119)
10:14:08.867736 IP c9060257.virtua.com.br.domain > 192.168.1.105.32729: 60227 2/3/3 CNAME bosontreinamentos.com.br
., A 186.202.153.82 (186)
10:14:09.314336 IP 192.168.1.105.25309 > c9060257.virtua.com.br.domain: 45976+ PTR? 87.2.6.201.in-addr.arpa. (41)
10:14:09.330894 IP c9060257.virtua.com.br.domain > 192.168.1.105.25309: 45976 1/2/0 PTR c9060257.virtua.com.br. (
113)
10:14:09.331523 IP 192.168.1.105.26142 > c9060257.virtua.com.br.domain: 12677+ PTR? 105.1.168.192.in-addr.arpa. (
44)
10:14:09.359350 IP c9060257.virtua.com.br.domain > 192.168.1.105.26142: 12677 NXDomain* 0/1/0 (130)
10:14:10.195335 IP 192.168.1.105.44184 > c9060257.virtua.com.br.domain: 25213+ A? fonts.googleapis.com. (38)
10:14:10.206372 IP 192.168.1.105.61246 > c9060257.virtua.com.br.domain: 63139+ A? pagead2.googlesyndication.com.
(47)
10:14:10.231581 IP c9060257.virtua.com.br.domain > 192.168.1.105.61246: 63139 4/4/4 CNAME pagead46.l.doubleclick.
net., A 173.194.118.109, A 173.194.118.121, A 173.194.118.122 (278)
10:14:10.231683 IP c9060257.virtua.com.br.domain > 192.168.1.105.44184: 25213 2/4/4 CNAME googleadapis.l.google.c
```

## 7. Referências

[https://nmap.org/man/pt\\_BR/index.html](https://nmap.org/man/pt_BR/index.html)

[https://nmap.org/man/pt\\_PT/man-host-discovery.html](https://nmap.org/man/pt_PT/man-host-discovery.html)

<https://nmap.org/movies/>

[https://nmap.org/man/pt\\_BR/man-port-scanning-techniques.html](https://nmap.org/man/pt_BR/man-port-scanning-techniques.html)

[https://nmap.org/man/pt\\_BR/man-port-scanning-techniques.html](https://nmap.org/man/pt_BR/man-port-scanning-techniques.html)

[https://nmap.org/man/pt\\_BR/man-examples.html](https://nmap.org/man/pt_BR/man-examples.html)

[http://www.dicas-l.com.br/arquivo/como\\_utilizar\\_o\\_tcpdump.php#.WyBMp9Zv85k](http://www.dicas-l.com.br/arquivo/como_utilizar_o_tcpdump.php#.WyBMp9Zv85k)

<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/09/como-usar-o-wireshark.html>

<https://www.diegomacedo.com.br/escaneando-portas-com-nmap/>

<http://www.bosontreinamentos.com.br/redes-computadores/tcpdump-capturar-e-analisar-trafego-de-rede-no-linux/>

[https://nmap.org/man/pt\\_BR/man-port-scanning-basics.html](https://nmap.org/man/pt_BR/man-port-scanning-basics.html)

<https://www.diegomacedo.com.br/introducao-ao-wireshark-deteccao-e-captura-de-trafego-em-redes/>

<https://capec.mitre.org/data/definitions/303.html>

<https://www.diegomacedo.com.br/introducao-ao-wireshark-deteccao-e-captura-de-trafego-em-redes/>

[https://www.google.com.br/search?q=Captura+do+escaneamento+com+Tcpdump&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi2w-3C0dHbAhUJxpAKHXTUC30Q\\_AUICygC#imgsrc=B\\_WiNLBKCRAdmM:](https://www.google.com.br/search?q=Captura+do+escaneamento+com+Tcpdump&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi2w-3C0dHbAhUJxpAKHXTUC30Q_AUICygC#imgsrc=B_WiNLBKCRAdmM:)

<http://www.bosontreinamentos.com.br/redes-computadores/tcpdump-capturar-e-analisar-trafego-de-rede-no-linux/>

[https://www.google.com.br/search?q=Exemplos+de+uso+Escaneamento+UDP&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiQgruU6NHbAhXIhJAKHa7tDwEQ\\_AUICigB&biw=1920&bih=974#imgsrc=NpYKTv7yeiW5DM:](https://www.google.com.br/search?q=Exemplos+de+uso+Escaneamento+UDP&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiQgruU6NHbAhXIhJAKHa7tDwEQ_AUICigB&biw=1920&bih=974#imgsrc=NpYKTv7yeiW5DM:)

<http://www.bosontreinamentos.com.br/redes-computadores/tcpdump-capturar-e-analisar-trafego-de-rede-no-linux/>