****

**Visão**

Com a crescente demanda sobre Tecnologias, percebemos que muitas pessoas apesar de buscarem informações, não possuem fontes que queiram realmente passar o conhecimento da maneira como ela deve ser, livre e com embasamento técnico que permita ser aplicado e utilizado quando necessário, além de serem testados em sua criação, tornando esta informação útil e confiável.

**Missão**

O Laboratório foi criado com a intenção de buscar e disseminar o conhecimento de uma maneira clara e objetiva, de forma gratuita, auxiliando na evolução dos membros e da sociedade na qual estas informações são compartilhadas, buscando o crescimento de todos os envolvidos nesta criação de valores.



Caso você pense que com a leitura dos materiais da How2Security, você irá se tornar um Cracker capaz de invadir sistemas, se você espera encontrar aqui scripts infalíveis para invasão e, a partir deles, sair por aí invadindo computadores, essa não é a leitura indicada. Indicamos, sim a leitura do Código Penal (Lei 2.848/1940), principalmente a Lei Carolina Dickmann (Lei 12.737/2012), nos Artigos 154-A e 154-B.

*154-A Invadir dispositivo informático alheio, conectado ou não à rede de computadores, mediante violação indevida de mecanismo de segurança e com o fim de obter, adulterar ou destruir dados ou informações sem autorização expressa ou tácita do titular do dispositivo ou instalar vulnerabilidades para obter vantagem ilícita:*

*Pena – Detenção, de 3 meses a 1 ano, e multa*

Este material é um conjunto de informações compiladas de documentos e ferramentas do Mundo Underground testadas em ambiente de laboratório na nossa intranet. Desta forma, todo conhecimento aqui condensado é tangível, assim como as orientações das contramedidas.

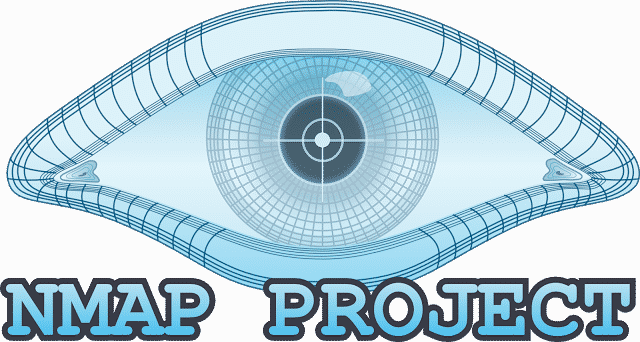
Dessa forma, esperamos ter sido bem claros que, em momento algum, estamos com a pretensão de ensinar a você como se tornar um invasor. Estaremos sim, mostrando muitas das técnicas utilizadas pelos crackers e, em alguns casos, pelos scripts kiddies, para que você, como administrador de redes, seja capaz de identificá-las em tempo hábil para se defender, antes que alguém com desejos menos nobres ô faça por você.

Assim sendo, todo o conteúdo dessa literatura tem apenas o objetivo didático de informar e preparar os administradores de redes dos novos tempos. Em momento algum nos responsabilizamos pelo mau uso desse conhecimento ou por danos causados em seu equipamento ou de terceiros, assim como também não somos responsáveis pelos códigos e ferramentas aqui citados.

Sandro Melo

Adaptado por Wellington Silva aka Well

**0 – NMAP (Netwotk Mapper)**



O NMAP (Network Mapper - Mapeador de Redes) é uma ferramenta de código aberto para exploração de redes e auditoria de segurança. O Nmap é utilizado para determinar quais hosts estão disponíveis na rede, quais serviços o host oferece (nome e versão da aplicação), quais sistemas operacionais está rodando no host, que tipo de firewall/filtros de pacotes estão em uso, e dezena de outras características.

**1 – Especificação de Alvos**

Tudo na linha de comando do Nmap, que não for uma opção (ou argumento de opção) será tratado como uma especificação de host alvo. O caso mais simples é a specificação de um endereço IP ou nome de host alvo para exame.

O Nmap suporta o endereçamento CIDR:

root@kali-wellx86:~# nmap -v 192.168.1.0/24

root@kali-wellx86:~# nmap -v 192.168.2.0/30

root@kali-wellx86:~# nmap -v 192.168.3.0/23

O Nmap suporta o endereçamento por bloco de octetos.

root@kali-wellx86:~# nmap -v 192.168.1.1-254

root@kali-wellx86:~# nmap -v 192.168.2.1-2

root@kali-wellx86:~# nmap -v 192.168.3.1-254 192.168.4.1-254

O Nmap aceita uma junção de todos eles na linha de comando.

root@kali-wellx86:~# nmap -v 192.168.1.0/24 www.how2sec.com.br 192.168.2.1-2

**Opções:**

* **-iL <arquivo.txt> 🡪** Lê as especificações de alvos de <arquivo.txt>. Aqui podemos passar uma grande lista de host armazenada em arquivo para o Nmap analisar. São aceitos as entradas no formato da linha de comando e cada entrada deverá estar separadas por um ou mais espaço, tabs ou newlines.
* **-iR <número>** 🡪 Faz o levantamento de hosts aleatoriamente. O argumento <número> é a quantidade de endereços IP aleatório será gerado pelo Nmap para analisar.
* **--exclude <host1>,<host2>** 🡪 Especifica uma lista, separada por vírgulas, de alvos a serem excluídos da análise.
* **--excludefile <arquivo.txt>** 🡪 Especifica um arquivo onde contém os hosts a serem excluídos da análise.

**2 – Descoberta de Hosts**

Um dos primeiros passos em qualquer missão de reconhecimento de rede é reduzir um conjunto de faixa de endereços IP a uma lista de hosts ativos e interessantes. O exame de cada porta de cada endereço IP é lento e desnecessário. Por isso, a importância de primeiramente analisarmos quais hosts estão ativos através das opções P\* (que seleciona o tipo de ping) que podem ser combinadas para fazer esse levantamento.

**Opções:**

* **-sL <nomeHost/CIDR>** 🡪 O exame de lista é uma forma de descobrimos quais hosts existem na faixa de endereço IP do nosso alvo. O exame de lista é um bom cheque de sanidade, para assegurar que você tem os endereços IP apropriados para seus alvos. Se os hosts apresentarem nomes de domínios que não sejam reconhecidos, valerá apena investigar mais para evitar o exame de rede da companhia errada.
* **-sP** 🡪 Esta opção diz ao Nmap para realizar apenas um exame por ping e depois exibir os hosts disponíveis que responderam ao ping. A opção **-sP** envia uma requisição de **Echo ICMP** e um pacote **ACK** na porta **80 TCP**.
* **-PN** 🡪 Não envia o **Echo ICMP** para os hosts.
* **-PS <lista de portas>** 🡪 Esta opção envia um pacote **TCP vazio**, com a flag **SYN** **ativa**. A lista de portas não pode conter espaços, por exemplo: **-PS25,80,113,1050,35000**. Aqui é um início de conexão e o host remoto deverá responder com um pacote com a **flag SYN/ACK** indicando sua existência. No caso da porta está fechada o host remoto deverá responder com um **RST**. Caso não recebe nenhum pacote é um forte indício de que a porta está sendo filtrada por um firewall.
* **-PA <lista de portas>** 🡪 Esta opção envia um pacote **TCP vazio**, com a **flag ACK ativa**. O envio do pacote com o **ACK ativo**, supõe estar reconhecendo dados através de uma conexão TCP estabelecida, mas tal conexão não existe. Então o host remoto deverá sempre responder com um pacote contendo a **flag RST**, revelando sua existência, no processo.
* **-PU <lista de portas>** 🡪 Outra opção para encontrar hosts ativos na rede é através do **ping UDP**, que envia pacotes vazios ( a menos que **--data-length** seja especificada) de UDP às portas dadas. A omissão da porta fará que o pacote seja destinado a porta **31338**. Isso porque o envio para portas abertas é indesejável, para este tipo de exame em particular. Ao atingir uma porta fechada no host alvo, o UDP deverá induzir o host alvo enviar um pacote ICMP de porta inalcançável.
* **-PE; -PP; -PM** 🡪 A opção **-PE** habilita o exame de **echo ICMP** além de outras como **requisição de horário**, **requisição de informações** e **requisição de máscara de endereço**, como sendo os códigos **13**, **15** e **17** respectivamente. As consultas de horário e máscara são solicitadas através das opções **-PP** e **-PM** respectivamente. E a resposta por parte do host remoto deverá ser uma resposta de horário com o código **14** do **ICMP** e/ou uma resposta de máscara de endereço com o código **18**, revelando que o host está ativo.
* **-PO <lista de protocolos>** 🡪 Esta opção envia pacotes IP com o número de protocolo especificado ativo, em seu cabeçalho de IP. A lista de protocolos tem o mesmo formato que as listas de portas das opções discutidas anteriormente de descoberta de hosts. Se nenhum protocolo for especificado o padrão é o envio para múltiplos protocolos tais como: **ICMP** (protocolo **1**), **IGMP** (protocolo **2**), **IP-over-IP** (protocolo **4**). Este método de descoberta procura por qualquer resposta do host alvo, como protocolo inalcançável, que significa que o protocolo dado não é suportado pelo host alvo.
* **-PR** 🡪 **Ping por ARP**. Quando temos o bloqueio do ICMP e estamos dentro da mesma rede Ethernet dos alvos que buscamos a melhor forma e mais confiável é a busca pelos hosts alvos através do **ARP**, já que para se comunicar camada 2 do modelo ISO/OSI devemos mapear endereços de camada 2 com endereços de camada 3. O Nmap faz isso automaticamente quando percebe que estamos mapeando hosts em uma rede local. Para evitar esse comportamento devemos utilizar **--send-ip** para que consultas ARP seja ignorada.
* **--traceroute** 🡪 **Traceroute** são executados após os exames, usando informações dos resultados deste para determinar a porta e o protocolo mais provável de alcançar o host alvo. Ele funciona com todos os tipos de exames, exceto os exames por connect (-sT) e exames ociosos (-sI). O traceroute funciona através do envio de pacotes com um baixo **TTL (Time-to-Live - Temppo-de-Vida)**, numa tentativa de induzir mensagens de ICMP de Tempo Excedido dos saltos intermediários entre o scanner e o host alvo.
* **-n** 🡪 Diz ao Nmap nunca fazer resolução de DNS reverso nos endereços IP ativos que ele encontrar. Como o DNS pode ser lento essa opção pode ser interessante para agilizar nosso exame.
* **-R** 🡪 Por outro lado esta opção diz ao Nmap para sempre fazer resolver (DNS Reverso) de todos os hosts ativos.
* **--system-dns** 🡪 Quando utilizado o DNS local o Nmap faz a consulta DNS e aguarda a resposta para prosseguir, para que seja feita diversas consultas (paralelas) podemos utilizar esta opção.
* **--dns-servers <endereço-ip-dns1, endereço-ip-dns2>** 🡪 Por padrão o Nmap utiliza o DNS configurado no host local para fazer a resolução de nomes. Caso queira especificar um servidor DNS diferente podemos utiliza esta opção e indicar qual servidor DNS utiliza para a resolução de nomes. Esta opção se mostra muito útil em redes privadas. Às vezes, apenas alguns servidores DNS fornecem informações de hosts locais. Então você pode examinar a rede em busca de serviços na porta 53, depois utilizar exame de lista (-sL) especificando cada servidor de nomes encontrado, com a opção **--dns-servers**.

**3 – Estado das Portas no NMAP**

Enquanto muitos scanners de portas têm tradicionalmente misturado todas as portas nos estados aberta ou fechado, o Nmap é muito mais criterioso. Ele divide as portas em seis estados: **open** (aberta), **closed** (fechada), **filtred** (filtrado), **unfiltred** (não filtrado), **open|filtred** (aberto ou filtrado) e **closed|filtred** (Fechado ou filtrado).

* **open (aberto)** 🡪 Uma aplicação está ativamente aceitando conexões TCP/UDP nesta porta.
* **closed (fechado)** 🡪 Uma porta não está aceitando conexões por não haver aplicação atendendo nesta porta TCP/UDP.
* **filtred (filtrada)** 🡪 O Nmap não pode determinar se a porta está aberta porque a filtragem de pacotes impede que suas provas a alcancem.
* **unfiltred (não filtrada)** 🡪 O estado não filtrado significa que uma porta está acessível, mas o Nmap não pode determinar se ela está aberta ou fechada.
* **open|filtred (aberta|filtrada)** 🡪 O Nmap põe a porta neste estado quando é incapaz de determinar se ela está aberta ou filtrada. Isto ocorre nos tipos de exames em que as portas abertas não fornecem resposta.
* **closed|filtred (fechada|filtrada)** 🡪 Este estado é usado quando o Nmap é incapaz de determinar se uma porta está fechada ou filtrada. Ele só é usado no exame ocioso de ID de IP.

**4 – Técnicas de Exame de Portas**

Esta seção documenta as cerca de dezenas de técnicas de exames de portas suportadas pelo Nmap. Somente um método pode ser usado de cada vez, com exceção do exame UDP (-sU) pode ser combinado com qualquer um dos exames de TCP.

* **-sS** 🡪 Exame por **TCP flag SYN**. Esta técnica é conhecida como **half-open** (meio-aberta), porque você não abre uma conexão TCP completa, tornando a busca muito rápida. Você envia um pacote **SYN**, como se fosse abrir uma conexão real e, então, espera por uma resposta. Um **SYN/ACK** indica que a porta está atendendo (aberta), enquanto o **RST** (reset) indica uma porta fechada. Se nenhuma resposta for recebida, depois de várias retransmissões, a porta será marcada como filtrada. Ela também será marcada como filtrada se um erro de ICMP de inalcançável (tipo 3, código 1, 2, 3, 9, 10 ou 13) for recebida.
* **-sT** 🡪 Estabelece uma conexão TCP completa.
* **-sU** 🡪 Exame por UDP. Embora a maioria dos serviços populares na Internet rode em cima do protocolo TCP, temos serviços importantes rodando em UDP. DNS, SNMP e DHCP (portas 53, 161/162 e 67/68) são três dos serviços mais comuns. O exame UDP funciona enviando um cabeçalho vazio a cada porta alvejada. Se um erro de ICMP de porta inalcançável (tipo 3, código 3) for retornado, a porta está fechada. Outros erros de ICMP de inalcançável (tipo 3, códigos 1, 2, 9, 10 ou 13) marcam a porta como filtrada. Eventualmente, um serviço responderá com um pacote de UDP, provando que ela está aberta. Se nenhuma resposta for recebida após várias retransmissões, a porta será classificada como **open|filtred**. Para melhorar a resposta podemos utiliza **-sV** para diferenciar as portas abertas das filtradas com o banner de resposta.
* **-sN, -sF, -sX** 🡪 Exame do tipo **TCP Nulo**, por **FIN** e de **Natal**. Quando exames destes tipos forem feitos e a RFC 793 for obedecida, qualquer pacote que não contenha os bits **SYN**, **RST** ou **ACK** ativos resultará num **RST** se a porta estiver fechada, e absolutamente nenhuma resposta se a porta estiver aberta.
* **-sA** 🡪 Exame por **ACK**. Este exame é diferente dos outros discutidos até agora por nunca determinar portas abertas. Ele é usado para mapear conjuntos de regras de firewalls, determinando se eles são de estado ou não, e quais portas estão filtradas. O exame ACK, quando examinando sistemas não filtrados, tanto as portas abertas quantos as fechadas retornarão um pacote **RST**. O Nmap marcará como filtradas as que não responderem ou responderem com ICMP (tipo 3, códigos 1, 2, 3, 9, 10 ou 13).
* **-sW** 🡪 Exame de **Janela TCP**. O exame por janela (window) é exatamente o mesmo que o exame ACK, exceto que ele explora um detalhe de implementação de certos sistemas, quando o **RST** for retornado. Ele o faz pelo exame do valor da janela TCP dos pacotes RST. Em alguns sistemas, as portas abertas usam um tamanho de janela positivo (mesmo para pacotes RST), enquanto as fechadas têm uma janela igual a zero.
* **-sM** 🡪 Exame de TCP de Maimon. O exame foi nomeado em homenagem ao seu descobridor, Uriel Maimon. Ele descreveu a técnica na edição 49 da Revista Phrack (novembro de 1996). Esta técnica é exatamente a mesma que os exames nulos, por FIN e de Natal, com exceção de que a prova é FIN/ACK. De acordo com a RFC 793 (TCP), um RST deve ser gerado em resposta a tal prova, se a porta estiver aberta ou fechada. Entretanto, Uriel percebeu que muitos sistemas derivados do BSD simplesmente descartavam o pacote, se a porta estivesse aberta.
* **--scanflags <FLAGS TCP>** 🡪 Exame personalizado de Flags TCP. Um usuário avançado não precisa se limitar aos tipos de exames enlatados. A opção --scanflags permite que você projete seu próprio exame, especificando as flags TCP arbitrária.
* **-sI <host\_zumbi:porta>** 🡪 Exame **Idle PC**. Ao invés de um ataque direto ao host alvo, ele utiliza um host zumbi através da exploração de geração predizível de sequência de ID de fragmentação de IP, no host zumbi, para coletar informações sobre as portas abertas no alvo.
* **-sO** 🡪 Exame de protocolos IP. Esta opção envia pacotes IP com o número de protocolo especificado ativo, em seu cabeçalho de IP. A lista de protocolos tem o mesmo formato que as listas de portas das opções discutidas anteriormente de descoberta de hosts. Se nenhum protocolo for especificado o padrão é o envio para múltiplos protocolos tais como: **ICMP** (protocolo **1**), **IGMP** (protocolo **2**), **IP-over-IP** (protocolo **4**). Este método de descoberta procura por qualquer resposta do host alvo, como protocolo inalcançável, que significa que o protocolo dado não é suportado pelo host alvo.
* **-b <host-ftp>** 🡪 Uma funcionalidade interessante do protocolo FTP (RFC 959) é o suporte às chamadas conexões de FTP por representante. Isto permite que um usuário se conecte a um servidor FTP e, depois, solicite que arquivos sejam enviados a um servidor de terceiros. Tal funcionalidade é oportuna para abusos em muitos níveis, de forma que a maioria dos servidores deixou de suportá-la. Um dos abusos que esta funcionalidade permite é fazer com que o servidor FTP examine as portas de outros hosts enviando um arquivo para cada porta no host alvo e analisando o erro para saber se a porta está aberta ou não. Os argumentos utilizados são: **-b <user>:<password>@<server>:<port>**, e pode ser omitido o usuário e senha em caso de conexões anônimas.

**5 – Especificação de Portas para Exames**

Além de todos os métodos discutidos anteriormente, o Nmap oferece opções para especificação de quais portas serão examinadas e se a ordem dos exames será aleatória ou sequencial.

* **-p <portas>** 🡪 Estas opção especifica quais portas você que examinar e sobrepõe as padrão. Número de portas individuais são aceitas, assim como, faixas de portas separadas por um hífen (p.e.: 1-1024). Você pode especificar -p- para examinar as portas de 1 a 65535. O exame de porta zero é permitido se você especificar explicitamente. Podemos informar que a porta é TCP ou UDP através do argumento T: ou U: respectivamente. Por exemplo, se especificarmos -p U:53,111,137,T:21-25,80,139,8080, fará com que as portas UDP 53, 111 e 137 e as portas TCP de 21 até 25, 80, 139 e 8080 seja examinadas. Para um exame de portas que vá da porta especificada até a mais inferior a ela podemos usar colchetes, por exemplo, -p[-1024], com isso, todas as portas inferiores a 1024 serão examinadas.
* **-F** 🡪 Especifica que você deseja examinar menos portas que o padrão. Normalmente, o Nmap examina as 1000 portas mais comuns para cada protocolo especificado. Com a opção -F esse número é reduzido para 100.
* **-r** 🡪 Não randomiza as portas a serem examinadas, seguindo a sequência delas.
* **--port-ratio <0|1>** 🡪 Examina todas as portas do arquivo nmap-services com uma razão maior que o número especificado como argumento.
* **--top-ports <Número => 1>** 🡪 Examina as portas com razão mais altas encontrada no arquivo nmap-services.

**6 – Detecção de Serviços**

O Nmap não apenas informa que serviço por padrão estão nas portas abertas, mas vasculha a fundo por assinaturas contida em seu arquivo nmap-services-probe atrás de serviços que está rodando, versão de aplicativo onde está o serviço, etc.

* **-sV** 🡪 Habilita a detecção de versão, alternativamente você pode usar **-A**, que habilita a detecção de versão, dentre outras coisas.
* **--allports** 🡪 Por padrão o Nmap pula a porta 9100 TCP, porque algumas impressoras simplesmente imprimem qualquer coisa enviada a esta porta, levando a dezenas de páginas de requisições HTTP GET. Com essa opção ele ignora essa diretiva padrão do Nmap.
* **--version-intensity <intensidade>** 🡪 A intensidade padrão é **7** e quanto maior o valor maior será a precisão, porém, demora mais, podemos informar o valor de intensidade que vai de **0** a **9**.
* **--version-light** 🡪 Esta opção equivale a intensidade 2 da opção anterior. Ela agiliza o exame porém é menos precisa.
* **--version-all** 🡪 Esta opção equivale a intensidade 9, assegurando que todas as provas serão tentadas para cada porta.
* **--version-trace** 🡪 Esta opção faz com que o Nmap apresente extensivas informações de depuração sobre o exame de versão.
* **-sR** 🡪 Faz o exame de RPC. Ele pega todas as portas TCP e UDP encontradas aberta e as inunda com comandos NULL de programas de SunRPC, numa tentativa de determinar se elas são portas de RPC e, se sim, qual o número de programa e de versão elas servem.

**6 – Detecção de S.O.**

Uma das funcionalidades mais bem conhecidas no Nmap é a detecção de S.O. remoto, usando a coleta de impressão digital da pilha TCP/IP. Para isso o Nmap examina os pacotes TCP e UDP enviado ao host e depois de dezenas de testes de amostragem de ISN de TCP, de suporte e ordenação de opções TCP, de amostragem de ID de IP, e verificação do tamanho da janela inicial, o Nmap compara os resultados com sua base de dados de mais de mil impressões digitais de S.O., o nmap-os-db, e exibe os detalhes do S.O. se houver uma correspondência.

* **-O** 🡪 Habilita a detecção de S.O..
* **--osscan-limit** 🡪 Limita a detecção a alvos que responde por pelo menos uma porta aberta e outra fechada. Caso não encontre nenhuma porta aberta não se dá ao trabalho de fazer essa analise agilizando o processo de exame.
* **--osscan-guess**; **--fuzzy** 🡪 Quando o Nmap é incapaz de detectar uma correspondência perfeita de S.O. ele às vezes oferece correspondências aproximadas como possibilidades.
* **--max-os-tries** 🡪 Quando o Nmap falha na realiza a detecção do S.O. ele faz 5 tentativas se a condição for favorável e 2 se não for favorável por padrão. Com a opção **--max-os-tries** ele irá tentar o máximo possível.

**7 – Scripts NSE**

O mecanismo de scripts do Nmap (**NSE - Nmap Scripts Engineering**) é uma das funcionalidades mais poderosas e flexíveis do Nmap. Ele permite que os usuários escrevam (e compartilhem) scripts simples (usando a linguagem de programação LUA), para automatizar uma ampla variedade de tarefas de rede. Estes scripts são executados em paralelo, com a velocidade e a eficiência que temos no Nmap. Cada script contém um campo que o associa a uma ou mais categoria, são elas: **safe** (seguros), **intrusive** (intrusivo), malware (malisioso), version (versão), discovery (descoberta), vuln (vulnerabilidades), **auth** (autenticação) e **default** (padrão). Os scripts do NMAP ficam em ***/usr/share/nmap/scripts***.

* **-sC** 🡪 Realiza um exame de script usando o conjunto padrão de scripts. É equivalente a **--script=default**.
* **--script <categoria|arquivo>|all** 🡪 Executa o script por categoria, ou o arquivo indicado ou ainda todos os scripts. Os scripts do Nmap são armazenados em um subdiretório de scripts do Nmap por padrão, por uma questão de eficiência esses scripts são indexados em uma base dentro do subdiretório de scripts do Nmap chamado script.db.
* **--script-args <arg1>=<valor1>,<arg2>=<valor2>** 🡪 Permite que você passe argumentos para os scripts que serão executados.
* **--script-trace** 🡪 Se esta opção for especificada, toda comunicação de chegada e de saída realizada por um script será exibida.
* **--script-updatedb** 🡪 Esta opção atualiza a base de dados de scripts, encontrada em scripts/scripts.db, que é utiliza pelo Nmap para indexar os scripts, ele só é preciso ser executado caso você tenha acrescentado ou alterado algum script.
* **--script-help <script>** 🡪 Exibe ajuda a respeito do script

**8 – Temporizadores do NMAP**

Algumas opções aceitam um parâmetro de tempo. Este é especificado em milissegundos, por padrão, embora você possa indicar "s", "m" ou "h" ao valor para especificar segundos, minutos ou horas. Desta forma, os argumentos para **--host-timeout 900000**, **900s** e **15m** todas significam a mesma coisa.

* **--min-hostgroup <número hosts>; --max-hostgroup <número hosts>** 🡪 O Nmap tem a habilidade de realizar exames de portas ou exames de versões de múltiplos hosts em paralelo. Ele faz dividindo o espaço de endereços IP alvo em grupos e, depois, examinando um grupo por vez.
* **--min-parallelism <número>; --max-parallelism <número>** 🡪 Este número controla o número total de exames paralelos. Por padrão o Nmap controla isso de forma eficiente calculando o ideal de exames paralelos.
* **--min-rtt-timeout <tempo>, --max-rtt-timeout <tempo>, --initial-rtt-timeout <tempo>** 🡪 O Nmap mantém um valor de tempo de expiração de uma execução para determinar quanto tempo ele irá esperar por uma resposta a um exame, antes de desistir ou de retransmitir. Seus valores podem ser ajustados para uma rede que contém muitos filtros antes de alcançar o host alvo. Em uma rede local não costuma-se passar de 100 milissegundos.
* **--max-retries <número>** 🡪 Especifica o número máximo de retransmissões no exame de portas.
* **--host-timeout <tempo>** 🡪 Alguns hosts simplesmente levam muito tempo para serem examinados. Isto pode ser devido a hardware ou softwares de rede com um desempenho fraco ou não confiável. Para isso, podemos marcar um tempo máximo em que o Nmap irá aguardar por resposta do host, como por exemplo, 30m aguardando um host, lembrando que ele irá aguardar a resposta fazendo outros exames em paralelo.
* **--scan-delay <tempo>, --max-scan-delay <tempo>** 🡪 Esta opção faz com que o Nmap espere pelo menos o tempo dado, entre cada exame que ele envia a um dado host. Isto é particularmente necessário em alguns hosts como por exemplo hosts executando Solaris que normalmente respondem a cada 1 segundo por exame de porta UDP, fazendo necessário informar para **--scan-delay 1s**.
* **--min-rate <número>, --max-rate <número>** 🡪 A temporização do Nmap faz um bom trabalho na descoberta da taxa apropriada de conduzir um exame. Porém as vezes são necessários ajustes neste valor.
* **--defeat-rst-ratelimit** 🡪 Alguns sistemas aplicam limites de taxas de mensagens de pacotes com a **flag RST** que eles geram. Isto pode retardar o Nmap drasticamente. O uso desta opção elimina esse comportamento porém deixa o exame menos preciso.
* **-T <paranoid|sneaky|polite|normal|agressive|insane>** 🡪 As vezes esses ajustar todos esses valores podem gerar confusão, por isso o Nmap tem seis modelos de ajustes de temporizadores utilizando o **-T<Número>** para ajustar automaticamente. Você pode especificá-los com a opção -T e seus números (0 a 5) ou seus nomes. Os nomes dos gabaritos são **paranoid (0 - paranóico), sneaky (1 - furtivo), polite (2 - educado), normal (3), agressive (4 - agressivo) e insane (5 - insano)**. Os dois primeiros são para evitar IDS/IPS. O modo polite retarda o exame para usar menos largura de banda e recursos do host alvo. O modo normal é o padrão. O modo agressive acelera os exames pela presunção de que você está em uma rede razoavelmente rápida e confiável. Por fim, o modo insane presume que você está em uma rede rápida ou querendo sacrificar algo da precisão em prol da velocidade.

**9 – Modo Evasivo Evitando Firewall/IDS/IPS**

Muitos sistemas hoje em dia ficam guardados por sistemas que identificam e sinalizam para os sysadmins possíveis ataques. Com isso, é bem sabermos as opções que temos para fazer o bypass desses dispositivos para avaliarmos nossas defesas.

* **-f** 🡪 Faz a fragmentação do pacotes, utilizando pacotes IP minúsculos. A ideia é dividir o cabeçalho TCP entre vários pacotes e tornar-se difícil para os filtros de pacotes o que você está fazendo. Os pacotes são divididos em pacotes de 8 bytes cada. Caso ainda assim o host alvo esteja causando problemas ainda podemos utilizar a opção **--send-eth** para eliminar a camada de IP e enviar quadros crus de Ethernet.
* **--mtu <tamanho>** 🡪 Também podemos informar o tamanho do MTU que cada pacote terá. Não utilize essa opção em conjunto com a opção -f.
* **-D <IP Spoffing 1>,<IP Spoffing 2>** 🡪 Camufla um exame enviando as requisições como se tivesse originado em todos os endereços IP Spoffados além do host real.
* **-S <endereço IP>** 🡪 Em alguns casos o Nmap não conseguirá determinar seu endereço de origem. Nestes caso você pode especificar por qual endereço IP de origem você quer fazer o exame.
* **-e <interface>** 🡪 Diz ao Nmap por qual interface de rede enviar e receber os pacotes.
* **--source-port <número da porta>, -g <número de porta>** 🡪 Simula a porta de origem. Isso pode ser necessário em filtros que fixa a porta de origem em suas regras.
* **--data-lenght <número>** 🡪 Apenda dados aleatórios aos pacotes enviados, fazendo que eles não sejam enviados vazios.
* **--ip-options <S|R<rota>|L<rota>|T|U...>, --ip-options <string em hexa>** 🡪 Envia os pacotes com as opções de IP especificadas. Para ver as opções nos pacotes enviados basta especificar **--packet-trace**.
* **--ttl <valor>** 🡪 Ajusta o campo Time-To-Live do IP.
* **--randomize-hosts** 🡪 Embaralha a ordem dos hosts alvos. Isto pode tornar os exames menos óbvios para vários sistemas de monitoramento de rede, especialmente quando você a combina com opções de temporização lenta.
* **--spoof-mac <mac|prefixo mac|nome do fornecedor>** 🡪 Envia os pacotes com outro MAC Address.
* **--badsum** 🡪 Envia pacotes com somas-verificadoras (checksum) TCP/UDP inválidas. Como virtualmente todos os pacotes com a soma-verificadora deveriam ser descartados no host alvo, quaisquer respostas recebidas estarão provavelmente vindas de um Firewalls/IDS/IPS que não se preocupou de checar a soma-verificadora.

**10 – Saída do Comando NMAP**

Qualquer ferramenta de segurança só é útil quando a saída que ela gera se apresentam de forma compreensível e organizada.

* **-oN <nome\_arquivo>** 🡪 Solicita que a saída normal seja direcionada para um arquivo cuja o nome foi especificado.
* **-oX <nome\_arquivo>** 🡪 Solicita que a saída seja em XML para um arquivo cuja o nome foi especificado.
* **-oS <nome\_arquivo>** 🡪 Solicita que a saída seja em leet (l33t) para um arquivo cuja o nome foi especificado.
* **-oG <nome\_arquivo>** 🡪 Solicita que a saída seja “grepável” para o um arquivo cuja o nome foi especificado.
* **-oA <nome\_base>** 🡪 Solicita que a saída seja nos formatos normal, XML e “grepáve”l, de uma só vez. Eles serão armazenados no formato ***<nome\_base>.nmap***, ***<nome\_base>.xml*** e ***<nome\_base>.gnmap***, respectivamente.
* **-v** 🡪 Modo verboso, aumentando o nível de detalhes produzido na saída padrão.
* **-d <nível>** -🡪 Aumenta o modo de depuração. O modo verbose lhe fornece dados suficientes, a depuração está disponível para lhe imunda com muito mais! Seu nível vai de **0** a **9**, e quanto mais alto mais detalhes.
* **--reason** 🡪 Mostra o estado do host alvo e suas portas.
* **--packet-trace** 🡪 Traça os pacotes e os dados enviados e recebidos.
* **--open** 🡪 Mostra apenas as portas abertas, ou possivelmente abertas.
* **--iflist** 🡪 Lista as interfaces e rotas.
* **--log-erros** 🡪 Registra erros e advertências no arquivo de saída do modo normal.
* **--append-output** 🡪 Apenda, ao invés de sobrepor, arquivos de saída.
* **--resume <nome\_arquivo>** 🡪 Em caso do exame ser abortado antes do fim (a rede pode cair, você deve interromper devido o horário, etc.). Com esta opção você retoma o exame a partir de onde parou.
* **--stylesheet <caminho|URL>** 🡪 Ajusta a folha de estilo XSL para transformação da saída XML.
* **--webxml** 🡪 Carrega a folha de estilo do Nmap.Org.
* **--no-stylesheet** 🡪 Omite a declaração da folha de estilos XSL do XML.

**11 – Referências**

**Referências Bibliográficas**

**[1]** MELO, Sandro – Estudo de Técnicas para Exploração de Vulnerabilidades em Redes TCP/IP, 2º Ed, Rio de Janeiro, 2006, Editora Alta Books Ltda.

**[2]** Packetwatch Research. Disponível em <http://www.packetwatch.net/netcatfix.php>. Acessado em 28/03/2017.

**[3]** Fontes Netcat. Disponível em <http://download.insecure.org/stf/nc110.tgz>. Acessado em 28/03/2017.