

Scanner de Portas

Por: Jonas Monteiro Carreira (Consultor em Segurança da Informação)

E-mail: J3115@hotmail.com (para MSN) ou j3115_21@yahoo.com.br (para dúvidas críticas e sugestões)

1-Introdução

2-Portas de comunicação

3-Pacotes de comunicação

4-SYN e ACK

5-Scanner de Portas

6-N-map (Network Mapper)

7-Conclusão

1-Introdução

Quando um intruso tenta invadir sua rede para fins financeiros, deface, curiosidade e etc, ele tentará saber se sua rede roda um firewall, e se rodar, com quais tipos de restrições ela está configurada, ou seja, quais serviços a sua máquina esta rodando.

Se a rede está configurada para “ouvir” comunicações em portas, existem softwares que possuem uma visão de raios-X para expor quais portas estão rodando na máquina alvo. Uma porta aberta significa um serviço rodando e, como errar é humano, logicamente, este serviço estará vulnerável, se não estivermos atentos aos patches de segurança dos mesmos. Pronto, o invasor (principalmente um deface) correrá atrás de um exploits e você estará em complicações.

È sobre Scanners de portas que falaremos neste artigo. Como as comunicações destes programas agem para descobrir se a nossa rede está com alguma porta aberta, também vamos ver como evitar coisas do tipo.

Neste artigo, usaremos para estudos um Scanner de portas muito conhecido chamado N-map.

2-Portas de comunicação

As portas de comunicação servem para diferenciar protocolos que você usa na internet. Ao acessar um site no www, você está fazendo uso de um tipo de serviço (protocolo) através de uma porta. Quando você está transferindo arquivos por FTP, você está usando outro protocolo, provido por outra porta, e assim a vida caminha no mundo da WEB...

Um exemplo de porta que acessamos diariamente é a 80, para o protocolo http (www). Outro esquema, poderia ser no protocolo FTP de transferência de arquivos, que atua pela porta 21 (sendo que este usa duas portas: uma receber os dados e outra para enviar), mas o uso de portas não é um caso obrigatório, podemos configurar nosso servidor para atuar em qualquer outra porta. Mas ao digitarmos nosso endereço no browser, teremos que especificar em qual porta nosso site se encontra. Veja:

www.portas.com.br:Porta, onde porta estabelece a porta que configuramos no nosso servidor e seus usuários (internautas) também teriam que saber que a configuração de porta do seu site é específica.

Para estudos, podemos experimentar o comando netstat onde veremos algumas portas em que estamos trabalhando em nossa máquina. Não me estenderei muito sobre este comando, senão fugiremos do objetivo do artigo, mas em caso de dúvida, digite netstat help para obter mais informações.

Então podemos dizer que o uso de portas é para facilitar que serviços em nossa rede sejam executados, tudo ao mesmo tempo e na mesma máquina, devido ao fato de que cada serviço possui uma porta diferente para comunicação. E é sobre comunicação que falaremos agora.

3-Pacotes de comunicação

A arquitetura do protocolo TCP/IP trabalha utilizando 4 camadas: Host, inter-redes, transportes e aplicação. As camadas inter-redes e transportes são onde se localizam os protocolos IP e TCP. No nosso artigo, daremos atenção ao protocolo que roda na camada de transporte: O TCP. O TCP se encarrega de checar seu pacote de rede e determina a rota que seguirá, ou seja, o protocolo (http, ftp, telnet, etc) e a porta que o pacote chegará.

O pacote TCP possui uma estrutura interna bastante interessante, à qual resumirei para que leitor entenda como funciona, mas apenas um resumo:

Porta de origem;

Porta de destino;

Número seqüencial;

Tamanho;

Reservado;

Flags.

Bom, aí está um pouco da estrutura interna de um pacote TCP. O importante para o uso de um Scanner de Porta é os Flags. Flags são campos que fazem com que o TCP trate de como deve agir com determinado pacote na rede.

Os campos Flags possuem informações que designam a ação do pacote dentro da rede. As quais podemos citar:

PSH: Este tipo, chamado Push, determina ao TCP, que serão enviados todos os pacotes ao seu destino.

URG: Conhecido Flag Urgent, faz com este pacote tenha prioridade ao chegar ao seu destino.

SYN: Este flag é um dos flags mais importantes para um TCP. Chamado Synchronise, ele determina a comunicação entre destino e origem, agindo junto ao flag ACK.

ACK: Informa ao destino qual o próximo número de seqüência a máquina origem deseja receber.

FIN: Esta flag é utilizada para finalizar uma conexão.

Logo veremos como estes flags são utilizadas para técnicas de Scanner de Portas.

4-SYN e ACK

Uma comunicação entre dois Hosts na internet se dá com pacotes TCP com Flag SYN e ACK ligados. Quando nos conectamos a um site na internet, nós enviamos a um servidor um pacote TCP com a flag SYN ligado. O servidor web recebe a nossa requisição (TCP com SYN ligado) e, por sua vez, envia um pacote (TCP) com SYN-ACK ligado, em resposta. Quando o nosso host recebe a seqüência SYN-ACK, reconhecemos como se estivéssemos aptos a concluir a comunicação e, sendo assim, nosso host envia a resposta com o Flag ACK ligado. Um servidor Web permanece cerca de 40 segundos à espera de um pacote com Flag ACK para concluir a conexão. Se esse tempo expirar a conexão é abortada. A conexão com o servidor também só pode ser feita com apenas 8 máquinas ao mesmo tempo, mas como tudo não é perfeito, existe uma invasão denominada DOS (negação de serviço), em que consiste enviar vários pacotes SYN com endereços imaginários para o alvo. Ele, em contrapartida, responde, mas como não existe nenhum endereço, o servidor não obtém respostas e pára a conexão. Se, por exemplo, a cada 5 segundos, ele receber vários pacotes SYN de oito máquinas, provavelmente, ele interromperia seus serviços (este assunto ficará para o próximo artigo).

Infelizmente os pacotes TCP não garantem a confiabilidade das conexões. Os Flags têm a função de controlar os dados de forma confiável, mas eu acho que na época que o TCP/IP foi projetado, não se pensava que fosse se tornar tão útil para todos e também por invasores potenciais.

Bom, para entendermos melhor as flags vamos nos aprofundar mais na comunicação desses dados. Vimos que ao nos conectarmos com alguma máquina, enviamos a pacote TCP com uma Flad SYN ligada. Por sua vez, a máquina servidora nos envia o pacote SYN/ACK e recebe um ACK e conclui a conexão. Porém, existe algo a mais sobre esta comunicação. O que chamaremos de números de seqüência e números de reconhecimento. Exemplo:

```

                SYN
                (número de seqüência=6)
                ACK (número de reconhecimento=0)
Cliente----->Servidor

                SYN/ACK
                (número de seqüência=8)
                ACK 6+1(número de reconhecimento=7)
Cliente<-----Servidor

                ACK
                (número de seqüência=7)
                ACK=8+1(número de reconhecimento=9)
Cliente----->Servidor
```

Bem, vejamos neste esquema como funciona “os apertos de mãos” na internet, os números de seqüência são números aleatórios que são enviados para o destino. Podemos observar que, no primeiro instante, o número de reconhecimento está vazio. Ao receber o valor SYN, o servidor envia o número

de seqüência aleatória e faz um número de reconhecimento adicionando +1 ao número de seqüência do cliente e fazendo o número de reconhecimento e assim ocorre o processo de aperto de mão. Podemos reparar a importância da ACK para fazer números de seqüência entre dois hosts, lembrando que é neste esquema que ocorre Spoofing, ou seja, o invasor usa Sniffer para farejar o esquema números de seqüência e reconhecimento entre hosts dentro de uma rede e tentar determinar como são gerados tais números.

Bem, na vida real os números de seqüência e reconhecimento não são tão fáceis assim. Ao usarmos um Sniffer nós podemos observar os números de seqüência e reconhecimento da seguinte maneira:

SYN

Seq.number-->0x8ba42f85/0x0000000 ---- > Número de reconhecimento vazio, como visto antes.

SYN/ACK

Seq.number-->0x6ba55f23/0x8ba42f86 ----> Número de reconhecimento adicionado +1 ao anterior.

ACK

Seq.number-->0x8ba42f86/0x6ba55f24 ----> Número de seqüência sendo usado como o ACK anterior e número de reconhecimento sendo adicionado +1 no número de seqüência anterior.

Lembrando, também, que neste quadro, podemos usar técnicas para uma IDS implementada para atuar como farejadora de tráfego de rede.

Bem, vimos algo necessário no estudo de scanner de portas, porém isso foi apenas o básico no estudo de monitoração de rede, por isso recomendo ao leitor estudar sobre Sniffer ou, se ficou algum ponto obscuro ao leitor, me envie e-mail.

5-Scanner de Portas

A inspiração para escrever um artigo que relate sobre Scanner de Portas originou-se no fato de que a primeira ação de um invasor é determinar quais serviços rodam na sua rede para depois descobrir vulnerabilidades que se encontram à mostra, dependendo da astúcia do invasor, mas para isso, o invasor necessita da ferramenta de Scanner de Portas.

Vimos anteriormente, como funciona o esquema de portas no protocolo TCP/IP e pacotes TCP, (vale lembrar que não falamos de UDP, pois iria fugir do escopo do artigo). Para mais informações

sobre portas e seus determinados serviços, façam uma consulta na internet à procura de bons materiais, o que está cheio.

Para nossos estudos práticos, podemos usar o Nmap que pode ser encontrado em www.insecure.org. O Nmap é um Scanner de portas completo que possui muitas funções. Uma dessas funções são as de descobrir qual sistema operacional roda em determinada máquina. Técnica conhecida como Fingerprinting.

Uma das vantagens do Nmap é a de que usamos linhas de comando para fazemos nossos estudos. O que nos dá total segurança de sabermos o que estamos fazendo.

Mas, antes de falarmos do Nmap vamos ver alguns métodos onde podemos usar de malícia para Scanear uma determinada porta dentro de uma rede.

No início dos primeiros Scanners de Portas, eles eram projetados basicamente para concluir o aperto de mão e, com isso, descobria-se quais portas estavam abertas dentro da sua máquina, mas a segurança foi evoluindo e, com ela, os métodos de ataque também. Logo, os firewalls detectavam Scanners de Portas com métodos de apertos de mão. Com isso, foram se desenvolvendo malícias nos envios de pacotes TCP para descobrir uma porta, por isso vimos como funciona tudo para termos uma noção.

Um método muito usado para um Scanner de Portas bem sucedido é quando o Scanner envia um pacote TCP com a flag SYN ligada e o alvo, por sua vez, envia um SYN/ACK. Quando nosso host recebe esse aviso, significa que a porta está aberta, ou seja, não precisa concluir a conexão, sendo que já temos o que precisamos. Então o Scanner envia um RST, resetando a conexão. Quando a porta está fechada o alvo envia um RST/ACK ao invés de SYN/ACK.

Outro método conhecido é o envio de pacotes SYN/ACK para o alvo. Caso o host alvo não responda, significa que a porta está aberta. Outro esquema que espera uma resposta ignorada do alvo é o envio da flag FIN ou ACK. Bem, note que estes tipos de esquemas fogem do padrão de aperto de mão conhecido na comunicação entre hosts, estes tipos de escanners também são muitos usados para fugir de firewalls que trabalham com filtros de pacotes.

6-N-map(Network Mapper)

O N-map é um Scanner de portas muito usado por invasores e por profissionais na área de segurança. Tem a capacidade de fazer um Scanner na rede e determinar quais serviços, aplicações e versões a rede alvo está rodando, também é capaz de ser útil na técnica de fingerprint, onde ele irá descobrir o sistema operacional que roda na máquina alvo.

Um Scaneamento executado pelo N-map é feito através de opções de comandos que você atribui para obter informações de determinada rede. Uma porta possui estados dentro do N-map, cujos mais importantes são: Aberto (open) – significa que serviços estão sendo executados nesta porta;

Filtrado (filtered) – este estado diz que uma determinada porta ou serviço está filtrada por um firewall, o que impede o N-map de descobrir se está open ou fechada (closed). Closed – significa que nenhuma aplicação ou porta está aberta neste momento.

Para se executar um comando no N-map, deve-se utilizar o seu nome seguido dos parâmetros e do endereço do alvo. Exemplo:

```
n-map <parâmetros> [www.alvo.com.Br ou 127.0.0.1]
```

Um exemplo prático seria usar como parâmetro -A (para determinar o sistema operacional) e -T4 para seguir um Scanner padrão. Então seria assim:

```
n-map -A -T4 www.alvo.com.br
```

Observe que podemos usar vários parâmetros para nosso Scanner.

Para você conhecer mais sobre os comandos do N-map digite Help na sua linha de comando para os devidos estudos.

Para um estudo completo do manual do N-map, acesse: www.insecure.org

7-Conclusão

O primeiro passo que um invasor irá tomar para entrar em uma determinada rede será fazer um mapa da rede em questão. Para isso, ele irá usar Scanners de portas, comando de tracers e pesquisas no registro Br. É surpreendente como numa simples varredura na internet podemos ver redes que não rodam firewalls e, o pior, quando rodam não possuem regras suficientes para impedir um simples ICMP na rede.

Registros Br são de fato impossíveis de esconder informações, mas Firewalls configurados podem impedir que invasores tentem determinar um perfil bem elaborado da sua rede. Darei exemplos do Iptables (incluído no Kernel do Linux). Farei este exemplo com principal personagem: O Iptables. Porque ele é uma ótima ferramenta de filtro de pacotes e para quem souber administrá-lo bem, saberá também administrar outras ferramentas de firewall.

Lembrando que o nosso estudo está voltado para uma rede que possui uma DMZ (Zona desmilitarizada) e não de uma voltada à tecnologia NAT (Network Address Translation), onde se escondem hosts para que o invasor não consiga determinar o perfil da rede alvo (em um próximo artigo estudaremos técnicas de NAT). Sendo que também não é muito difícil encontrar redes de DMZs .

Comandos de Tracers servem para mapear o caminho da rede alvo. Eles usam pacotes UDP para o envio ao alvo e esperam pacotes ICMP como resposta do mesmo. Existem regras no Iptables que podem impedir pacotes do tipo ICMP na rede. Exemplo:

```
[root@jelis /root]# iptables -A FORWARD -p icmp --icmp-type echo-request -j DROP
```

Obs: Existem meios, com comandos Tracerts, de se burlar as regras que impedem ICMP, mas para impedir Script kids, estas regras são bem sucedidas.

Para impedir Scanner de Portas de todos os tipos, basta criar regras como as seguintes:

```
[root@jelis /root]# iptables -A FORWARD -p tcp --tcp-flags SYN,ACK,FIN,RST
```

```
RST -m limit --limit 1/s -j ACCEPT
```

Com isso, espero ter convencido o amigo leitor a estudar sobre regras de firewall, pois as duas citadas acima foram apenas o básico que o iptables tem a oferecer. Em caso de dúvidas, sugestões e críticas, me mande e-mail.

Devido ao envio de muitos e-mails para o endereço j3115@hotmail.com, minha caixa de correio eletrônico ficou com problemas e deixei de receber mensagens. Caso não tenha respondido seu e-mail, peço desculpas. Envie-me outro agora para: j3115_21@yahoo.com.br. O j3115@hotmail.com ainda está ativo para o caso de você desejar me adicionar no messenger.

Abraços, e até a próxima.

Jonas Monteiro Carreira.