Footprint: a arte Cracker de levantamento de informações de um alvo

Sandro Melo
Diretor de Tecnologia - 4Linux
sandro@4linux.com.br

Resumo. Este artigo fala sobre como crackers e desconfiguradores de páginas necessitam da coleta de informações do servidor alvo da organização. Usam scan para identificar o Sistema Operacional ativo, as portas de serviços ativos e vulnerabilidades. Todas as técnicas de scanner são necessárias para obter descrições sobre o respectivo sistema. Aqui, faz-se uma explanação das principais técnicas utilizadas para realização do levantamento de informações em redes manipulando recursos da pilha TCP/IP.

1. Introdução

Notadamente, o levantamento de informação é a fase mais relevante para o invasor de sistemas para que obtenha êxito em uma tentativa de intrusão. Por essa razão, é uma das partes mais trabalhadas e que recebe maior atenção pelos 'crackers'. É a partir dela que o invasor consegue informações essenciais para ser bem sucedido em uma invasão, como a topologia da sua rede, nomes de domínio e sistema operacional (fingerprint) usado, ou seja, tem por intuito criar um perfil do seu possível host alvo, para tentar descobrir falhas e eventuais brechas que possam ser exploradas, configurações e senhas padrão.

Faz-se necessário também destacar que para uma atividade de desconfiguração de páginas em servidores Web que têm como uma característica peculiar a atividade de 'scriptkiddies', quase sempre o levantamento de dado não requer tanta engenhosidade como um ataque 'cracker'.

Esse cenário trouxe como conseqüência a motivação para o desenvolvimento de ferramentas que facilitem o levantamento da informação, criando a possibilidade de, através delas, mesmo invasores com pouco conhecimento terem facilidade na execução do levantamento de informação conhecido técnicamente como Footprint [4] [10].

2. O Footprinting

Nada mais é do que a busca detalhada da maior quantidade de informações possíveis do alvo da invasão, tentando burlar, se possível, ferramentas IDS's ou Firewalls, ou seja, ninguém que queira invadir sua rede vai ficar tentando uma invasão sem ter uma estratégia definida. A partir do resultado obtido pelo Footprint é traçado o plano/estratégia de invasão. Há casos em que essa busca de informações chega a durar meses, mas, como sabemos, para um cracker o tempo não é problema.

Muitos Footprints iniciam-se com a velha, mas ainda muito usada, engenharia social. Os alvos comuns de footprinting [2] [3] [4] [5] [9] [10] [17] [21]:

- Nomes de domínio;
- Dados da empresa, como endereço, telefones;
- Responsáveis pelos domínios;
- Sistema Operacional do host alvo;
- Serviços TCP e UDP disponíveis;
- Topologia da rede;
- Nomes de usuários e de grupos;
- Possibilidades de Firewall, IDS;
- Endereços de email (principalmente de administradores);
- Informações de serviços SNMP mal configurados;
- Domínio da organização;
- Responsáveis pelo domínio;
- Possibilidade de acesso remoto;
- Mecanismos de autenticação.

2.1 A finalidade do levantamento de informação

Depois de obter o máximo de informações possíveis com o "Footprint", é traçado o plano de invasão. É muito importante que o responsável pela segurança de uma rede de computadores esteja informado das vulnerabilidades e correções respectivas ao seu sistema operacional.

Conhecer o alvo é o objetivo dessa fase, necessariamente um script kiddie é capaz de executar essa tarefa, embora faça-a de uma maneira mecânica. Por outro lado, um hacker ou um cracker nesse momento colocam sua imaginação para funcionar, pois essa é a hora que determina o quanto o seu alvo é vulnerável ou não. Mesmo o mais habilidoso dos hackers ou crackers vai despender dias e dias pesquisando e levantando informações do seu alvo, elaborando uma lista de todas as possibilidades de invasão. E a quantificação do alvo, como já foi mencionado, pode começar por uma ligação telefônica [2] [3] [4] [5] [21].

2.3 Basicamente, podemos dividir em três partes:

- Footprint - Organização dos dados levantados objetivando de forma coesa o melhor e mais completo perfil do alvo [5] [21];
- FingerPrint - Parte do Footprint que tem como finalidade identificar o S.O. do alvo [3];
- Enumeração - Basicamente, extrai informações do ambiente alvo, como contas de usuários, recursos compartilhados mal protegidos e principais serviços disponíveis [2] [3] [4] [6].

3. Conhecendo o Fingerprint da Pilha TCP/IP

O texto de Fyodor que acompanha a documentação do nmap é uma forte fonte de conhecimento sobre o assunto, nos tópicos a seguir, será feita uma explanação bem específica e valiosa de como extrair informações de uma máquina através das características implementadas em sua pilha TCP/IP [2].

É importante lembrar alguns métodos "clássicos" para determinar o S.O. de uma máquina sem envolver o fingerprinting. É obvio o por quê de determinar qual sistema operacional está rodando em uma máquina para o invasor. Muitas falhas de segurança estão vinculados à versão do sistema operacional. Vamos imaginar que o invasor vá realizar uma tentativa de intrusão, o "bug" é DNS e ele irá identificar se a porta 53 está aberta. Caso seja uma versão do BIND vulnerável, ele terá a oportunidade de explorá-la, mas sabe que tentativas falhas podem matar o *daemon* ou até mesmo chamar a atenção do administrador [2].

3.1 Técnicas Clássicas

A análise de *fingerprinting* resolve o problema de identificação de S.O. de forma única, normalmente procurando particularidades de implementação da Pilha TCP/IP. Acreditamos que esta é uma das técnicas mais promissoras, mas atualmente existem outras soluções para descobrir o S.O.. Uma das mais utilizadas é tentar um telnet para um determinado host para identificar o banner do serviço e, quase sempre, o sistema operacional.

3.2 Os scanners de FingerPrinting

Um "scanners" capaz de realizar um Fingreprint o faz quase sempre enviando uma sequência de datagramas pré-definidos e, em seguida, comparando a resposta do resultado com uma tabela na qual encontra-se relacionado o perfil de resposta de cada Sistema já conhecido. Essa lógica é a base dos scanners de Fingerprint. Esses programas podem ser divididos de forma didática em três categorias [2]:

- Passive OS Fingerprint
- Active OS Fingerprint
- Deamon OS Fingerprint

3.3 Técnicas de Exploração da Pilha TCP/IP que possibilitam o Fingerpring

Existem muitas técnicas que podem ser utilizadas para identificação da *fingerprint* da pilha de rede. Basicamente, a técnica consiste em identificar diferenças entre sistemas operacionais e desenvolver um código que classifique-as. Uma ferramenta capaz de combinar estes dois processos poderá chegar a um nível de refinamento da informação muito bom [2] [3].

3.3.1 Investigação de FIN (FIN probe)

Nessa técnica, enviamos um pacote FIN (ou qualquer pacote sem o flag de ACK ou SYN) para uma porta aberta e esperamos por uma resposta. A implementação correta da RFC 793 diz para NÃO responder*****, mas muitas implementações (pelo fato de não seguirem a RFC 793) como MS Windows, BSDI, CISCO, HP/UX, MVS e IRIX respondem com um RESET. A *RFC793*, especifica que portas *FECHADAS* devem responder com *RST (RESET)* e portas *ABERTAS* devem ignorar o pacote, mas muitos não seguem as especificações, e respondem com *RST* aos pacotes *FIN* enviados para portas abertas [2] [3].

3.3.2 Investigação FALSA (BOGUS flag)

Queso foi o primeiro *scanner* a utilizar esse tipo de teste. A idéia consiste em enviar um *flag* TCP indefinido (64 ou 128) no cabeçalho TCP de um pacote SYN. O Linux anterior ao 2.0.35 mantém este *flag* setado em sua resposta. Porém, outros sistemas operacionais parecem cancelar a conexão (resposta com RST) quando recebem um pacote SYN+BOGUS. Esse comportamento pode ser útil para identificar o sistema [2] [3].

3.3.3 Padrão TCP de ISN (TCP ISN Sampling)

A idéia por traz dessa técnica é a identificação de padrões do Número Inicial de Seqüencia (*ISN - Initial Sequence Number*)

escolhido pelo TCP ao responder a um pedido de conexão. Esses números podem ser classificados em vários grupos como o tradicional 64K (utilizado em muitas versões antigas de UNIX). O sistema Windows e outros sistemas utilizam um modelo que depende do horário onde o ISN é incrementado por um valor fixo a cada período de tempo [2] [3].

3.3.4 Bit de não fragmentação (Don't Fragment bit)

Muitos sistemas operacionais começaram a setar o bit de não fragmentação em alguns pacotes enviados. Com isso, temos vários benefícios de desempenho (mas pode também ser chato - é por isso que o scanner de fragmentação do nmap não funciona com Solaris). De qualquer forma, nem todos os sistemas operacionais fazem isso e alguns fazem em diferentes casos, sendo assim, se analisarmos estes bits podemos obter mais informações sobre o nosso alvo [2] [3].

3.3.5 Janela deslizante inicial do TCP (TCP Initial Window)

Técnica que envolve a análise do tamanho da janela devolvida pelos pacotes de retorno. Os *scanners* antigos classificam o sistema operacional como BSD 4.4, quando o pacote RST possui uma janela com valor diferente de zero. Os *scanners* mais novos como queso e nmap, através das janelas, conseguem deterninar o tipo de sistema operacional. Esse teste nos dá uma série de informações, uma vez que alguns sistemas operacionais podem ser identificados unicamente pela janela [2] [3].

3.3.6 Valor do ACK (ACK Value)

Embora possamos pensar que este valor seja um padrão, as implementações usam valores diferentes para o bit ACK em alguns casos. Por exemplo, vamos supor que seja enviado um FIN|PSH|URG para uma porta TCP fechada. Muitas implementações vão setar o ACK com o mesmo ISN inicial enviado, entretanto o Windows e algumas impressoras enviarão "seq+1". Se você enviar SYN|FIN|URG|PSH para uma porta aberta, o Microsoft Windows comporta-se de forma inconcistente. Em algumas vezes, ele devolve o "seq", em outras devolve S++. Há ainda situações em que ele devolve um valor randômico [2] [3].

3.3.7 Diminuindo as mensagens de erro ICMP (ICMP Error Message Quenching)

Alguns sistemas operacionais (inteligentes) seguem as sugestões da RFC 1812 no sentido de limitar a taxa de envio de mensagens de erro. Por exemplo, o kernel do Linux (net/ipv4/icmp.h) limita a geração de mensagens *destination unreachable* de 80 para 4 segundos, com uma penalidade de ¼ de segundo se o tempo for excedido [2] [3].

3.3.8 Mensagem ICMP de erro (ICMP Message Quoting)

As RFCs especificam que as mensagens ICMP de erro devem conter uma pequena parte da mensagem ICMP que causou o erro. Por exemplo, para uma mensagem de porta *unreachable*, quase todas as implementações enviam somente um cabeçalho IP + 8 bytes. Porém, o Solaris retorna um pouco mais e o Linux também. O bom disso é que a ferramenta "nmap" pode identificar se um sistema é Solaris ou Linux, mesmo que eles não tenham portas abertas, desde que devolvam resposta ICMP 3 [2] [3].

3.3.9 Tipo de serviço (Type of Service)

Verifica-se o valor do tipo de serviço (*TOS - type of service*) retornado pelas mensagens de ICMP *port unreachable*. Quase todas as implementações setam esse tipo de erro ICMP com o valor 0, mas o Linux usa 0cC0 [2] [3].

3.3.10 Controle de fragmentação (Fragmentation Handling)

Essa técnica tira proveito do fato de que as várias implementações freqüentemente fazem a remontagem dos pacotes de forma diferente. Algumas escrevem a porção antiga juntamente com a nova e, em outros casos, a porção antiga tem precedência [2] [3].

4. Enumeração

Essa fase do Footprint consiste na manipulação de datagramas TCP e UDP objetivando identificar tanto os serviços ativos, através de scanners de portas, como também as características especiais de uma topologia como, por exemplo, se respectivos servidores encontram-se em "Load Balance". Pode também enumerar roteadores, ou ainda a identificar sistemas de "Firewalls". Através de manipulações sutis do cabeçalho TCP e UDP, um bom scanner é capaz de

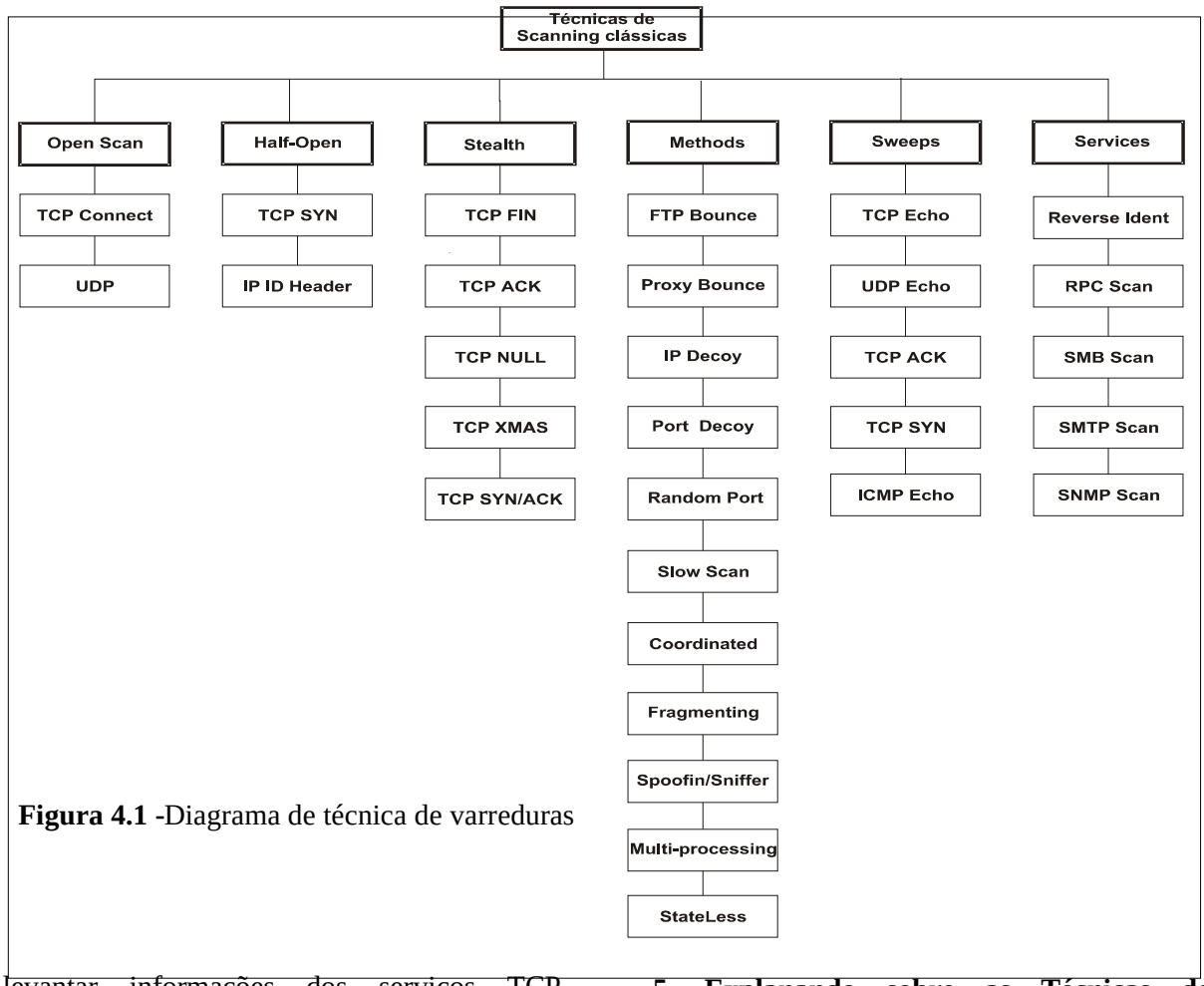

Figura 4.1 -Diagrama de técnica de varreduras

levantar informações dos serviços TCP disponíveis. Classificando as técnicas utilizadas: [2] [4] [5] [9] [20] [19] [21]

Open Scan - Classificadas quando a forma de manipulação do datagrama, varreduras baseadas no conceito de início de conexão.

Half Scan - Classificadas quando a forma de manipulação do datagrama, varreduras baseadas no conceito de fim de conexão e pacote nulo.

Stealh - Desenvolvidas para serem sutis no levantamento de dados.

Methods - Desenvolvidas para tentar enganar sistemas de IDS, motivando falsos positivos ou sutilmente passando sem ser detectada.

Sweep - Varreduras simples que têm como único objetivo comprovar que o host está ativo.

Service - Desenvolvidas para levantar informações a partir de serviços específicos.

5. Explanando sobre as Técnicas de Varreduras mais utilizadas

5.1 Varreduras TCP Connect

Essa é a forma mais básica da exploração do TCP, e praticamente todos os scanners de portas têm esse recurso. Na prática, seria um “hand shake” para cada porta definida na varredura. Em sistemas que usam a API Socket essa camada de sistema é conhecida como “connect()”, usada para abrir uma conexão a cada porta definida [7] [16].

Por ser a varredura mais comum, e pelo fato de realizar um “hand shake”, o que demanda dois datagramas por porta, essa varredura é facilmente detectável porque os registros do Host do alvo mostrarão um grupo da conexão que falharam pelo simples fato de a porta não estar em estado de conexão [2] [4] [5] [6] [8] [9] [10] [11] [18] [19] [20] [21].

5.2 Varreduras TCP SYN

Essa técnica é outra freqüentemente usada, também conhecida como "conexão semi-aberta", na qual a exploração não demanda um “hand shake” completo. O scanner envia um datagrama TCP SYN como

se estivesse abrindo uma conexão real e espera uma resposta [7] [16].

Por serem técnicas baseadas em início de conexão, muitos sistemas básicos de detecção de intruso identificam varreduras partindo da lógica que se um host envia um datagrama de início de conexão à uma porta que não está nesse estado, significa um datagrama arbitrário e o sistema assume-o como uma tentativa de varredura. O que, na pratica, é interessante, mas o problema é que as varreduras não se limitam ao conceito de "hand shake" [2] [4] [5] [6] [8] [9] [10] [11] [18] [19] [20] [21].

5.3 Varreduras basedas na RFC 793 (TCP FIN, TCP XMAS, TCP NULL)

Em épocas remotas até mesmo a exploração de FIN, XMAS ou Null não eram furtivas o bastante para não chamar a atenção. E pilhas TCP/IP respondem RST somente para portas abertas quando são padrão Unix [7] [16].

Essa técnica de varredura não funcionará para exploração em sistemas Microsoft, pois a pilha TCP/IP do Windows responde tanto em porta fechada quanto em aberta [2]. Embora possa parecer muito positivo, essa característica da pilha TCP/IP da Microsoft permite a um invasor definir que aquele host remoto é um sistema Microsoft. Segundo a própria documentação do NMAP [2] [8], outros sistemas seguem essa linha de raciocínio, pois possuem uma implementação similar a da Microsoft. Nessa lista, incluem-se o Cisco, o BSDI, o HP/UX, o MVS e o IRIX.

5.4 Varreduras do UDP

A técnica consite em emitir datagramas de 0 bytes UDP a cada porto no alvo máquina. Quando um datagrama UDP chega a uma porta fechada, é devolvida uma mensagem de erro ICMP 3 (unreachable). Caso não retorne nada, supostamente seria uma porta aberta [7] [16].

Um dado importante é que infelizmente a exploração do UDP é, às vezes, dolorosamente lenta, uma vez que a maioria das implementações TCP/IP levam à risca as definições da RFC 1812 (seção 4,3,2,8) [2] que indicam limitar a taxa da mensagem de erro do ICMP.

É observado que sistemas Microsoft respondem muito mais rápido a varreduras, o que indica o não seguimento da RFC 1812 na sua implementação. Assim sendo, é possivel fazer a varredura de todas as portas (até 65535) de uma máquina com sistema Microsoft muito mais rapidamente que em qualquer outra plataforma [2] [4] [5] [6] [8] [9] [10] [11] [18] [19] [20] [21].

5.6 Varreduras baseadas apenas no cabeçalho IP Protocol

Varreduras do protocolo do IP são um método usado para determinar quais protocolos IP são suportados em uma técnica de host. Consiste em emitir datagramas básicos do IP sem nenhum encabeçamento, atendendo apenas a definição do campo do tipo de protocolo. Caso seja recebido uma mensagem unreachable do protocolo do ICMP, o protocolo não está em uso. Do contrário, supõe-se que sim. Alguns bons exemplos de sistemas operacionais nos quais essa técnica é facilmente aplicada são: AIX, HP-UX, Digital UNIX. Todavia, os sistemas de "Firewalls" não emitem mensagens unreachable (ICMP 3), criando um cenário ideal para um scanner se enganar, pois isso faz com que todas as portas sejam definidas como abertas e os protocolos apareçam "ativos", gerando "falsos positivos" [2] [4].

A técnica executada é muito similar à exploração de portas do UDP, limite da taxa do ICMP mightapply demasiado. Mas o campo do protocolo IP tem somente 8 Bits, permitindo que 256 protocolos possam ser sondados [2] [4] [9] [10].

5.7 Varreduras para detecção de Firewalls

São varreduras destinadas à detectar a presença de um firewall em um domínio, possibilitando ao invasor desenhar a topologia domínio, mapeando os possíveis firewalls.

5.7.1 Varredura ACK

Esse método avançado é usado para identificar Firewalls Stateful ou um Packet Filter (firewalls que atuam na camada 3, camada IP, denominados filtros de pacotes) que no máximo, fazem apenas um controle com pacotes TCP com a flag SYN ativadas. A idéia dessa técnica é aproveitar que um datagrama TCP/ACK órfão (que não pertença a nenhuma conexão estabelecida) tenha como resposta RST, tanto em uma porta aberta, como em uma fechada. Dessa forma, se um RST voltar, as portas estarão classificadas como "não filtradas". Se nada voltar (ou se um ICMP unreachable for retornado), a porta é classificada como "filtrada" [2] [4] [9] [10].

5.7.2 Varredura TCP Window

Varredura da janela do interruptor: é avançada e muito similar à varredura do ACK, a não ser naquela em que ele possa portas abertas do sometimesdetect as.well.as filtered/nonfiltered devido a uma anomalia no tamanho da janela do TCP que relata por alguns sistemas se operando. Os sistemas vulneráveis a este incluem ao menos algumas versões de AIX, Amiga, BeOS, BSDI, Cray, Tru64 UNIX, DG/UX, OpenVMS, Digital UNIX, FreeBSD, Cavalo-força-ux, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.X, Ultrix, VAX, e VxWorks.

5.8 Varredura Bounce

Definido de uma forma didática como uma técnica que consiste em utilizar um serviço de um determinado host para levantar informações em outros. Um exemplo, foi a falha do comando "PORT" de um servidor FTP definida na RFC 959, na qual o servidor atua como um "proxy" para a varredura[2] [4] [9] [10]. Todavia, através de servidores proxy de cache mal configurados, a varredura bounce torna-se possível para levantamento de vulnerabilidades web com scanners com "Nikto" [15].

5.9 Varreduras ICMP

O ICMP não possibilita fazer sondagem em serviços, limitando as varreduras ICMP ao levantamento de informações como: hora do servidor, subnet, netmask . Por outro lado, possibilta a realização de algumas técnicas de fingerprint [2] [4] [9] [10] [14].

6. Varreduras a partir de serviços

São as técnicas utilizadas para o levantamento de informação a partir de um serviço específico. Entretanto, o cenário que possibilita esse tipo de exploração via Internet retrata também um domínio onde as políticas de segurança são extremamente fracas ou, o que é pior, nem existem. Do contrário, somente um invasor interno (insiders [19]) teria possibilidade de executá-la. Um exemplo disso é o serviço "finger" que possilibita identificar os usários ativos em um sistema Unix, normalmente ativo na porta 79, porém é pouco provável um host na Internet ou até mesmo um router ter esse serviço ativo, devido à popularidade de sua exploração. Mas ainda encontramos possibilidades de execução dessa técnica a partir de outros serviços, dos quais muitos são facilmente encontrados em host na Internet:

6.1 Varreduras Ident-based Port Querying

Através do serviço IDENT ativo no servidor alvo é possível levantar informações de usuários ativos.

6.2 Varreduras SMB

Informações sobre compartilhamentos em sistemas Windows e servidores Samba.

6.3 Varreduras SNMP

Informações em sistemas operacionais e dispositivos que utilizam esse procotolo para gerenciamento, mas com configurações padrões ou mal definidas.

6.4 Varreduras SMTP

Informações de contas de usuários através de consultas utilizando comandos como "vrfy, expn e rcpt".

6.5 Varreduras RPC

Informações sobre serviços com "portmap, nfs" e outros.

7. Formas Furtivas

São denomidas assim pois têm por objetivo permitirem uma varredura de forma discreta, desviando a atenção do administrador e seus aparatos de segurança, ou simplesmente não chamando a atenção. Uma forma básica seria uma varredura baseada no início da coneção para uma única porta. Por melhor que seja um sistema de detectação de intrusos, não poderia diferenciar uma varredura feita dessa forma numa porta de serviço de uma handshake que falhou.

Mas se o invasor deseja levantar informação do host por completo, pode apelar para outras técnicas, como por exemplo:

- Enviar junto com os pacotes de sua varredura vários pacotes com a origem forjada;
- Ataques coordernados nos quais são feitos scanners simultâneos de várias origens;
- Enviar pacotes de origem forjada para desviar a atenção;
- Manipular o campo de porta origem para enganar firewall (Packet Filter) e ids mal implementados;
- Enviar uma seqüência de vários pacotes forjados junto com a varredura, não permitindo identificar a real origem;
-Outra forma é a técnica conhecida como "Slow Scan" que possibilita temporizar as varreduras. Um bom exemplo são as opções disponíveis no Nmap.

- Varredura Paranoid (Paranoica) 5 minutos delay;
- Varredura Sneaky - 15 segundos delay;
- Varredura Polite (Educada) - 0.4 segundos delay;
- Varredura Normal (default);
- Varredura Aggressive (Agressiva) - 1.25 minutos por host;
- Varredura Insane (Insana) - 0.3 segundos.

8. Técnicas para varreduras em larga escala

8.1. Scanning com Multi-processing

São scanners capazes de abrir vários processos simultâneos, tornando o processo de levantamento das portas ativas mais dinâmico [12].

8.2. Fast Scanning em modo Stateless

Um dos problemas de trabalhar no modo stateless é a autenticidade dos dados que o scanner recebe, mas a possibilidade existe e permite varreduras com grande velocidade. Essa técnica também é conhecida como "Inverse SYN Cookies" [12].

9. Banner Grabbing

É uma técnica muito utilizada para o levantamento de informações, que consiste em ler banners dos serviços ativos. Normalmente, os banners trazem o nome e a versão do respectivo serviço. E, em muitos casos, também trazem a versão do sistema operacional.

10. Conclusão

As técnicas de scanners são métodos utilizados para testar e levantar informações de ambientes computacionais com o intuito de analisar e obter informações relevantes para fazer seu levantamento topológico, utilizando com sagacidade recursos do protocolo TCP/IP. Dessa forma, com o grande número de ferramentas disponíveis na Internet, a aplicação torna-se, independente da técnica, necessariamente simples.

Possibilitar que administradores possam testar seus próprios sistemas computacionais seria a finalidade ideal das técnicas de scanners. Apesar disso, elas são, notoriamente, parte do ferramental utilizado por um invasor para levantar dados, sendo um fator determinante para o sucesso de uma futura intrusão reunir o maior número possiível de informações de seu alvo.

O conhecimento dessa metodologia permite aos administradores de sistemas computacionais serem mais incisivos no desenvolvimento de soluções e contramedidas concisas.

Ao utilizar esse conhecimento, os administradores podem validar a segurança implementada, podendo avaliá-la num contexto muito próximo do real perigo e comprovar a validade das políticas definidas nos sistemas de detectação de intrusos e firewalls.

12. Referências Bibliográficas

[1] LONGMAN, Group Ltda. *Dictionary of contemporary English.* Longman, Third Edition, 1995.

[2] FYODOR. *The Art of Port Scanning, 1997.* Disponível em http://www.insecure.org, em 01/2003.

[3] FYODOR. *Remote OS Detection via TCP/IP stack Fingerprinting, 1998.* Disponível em http://www.insecure.org, em 01/2003.

[4] DETHY. *Examinig Ports Scan Methods.* Disponível em http://www.insecure.org, em 02/2003.

[5] SHEMA, Mike e SCAMBRAY, Joel. *Hacking Exposed.* Mc Graw Hill / Osborne, 1 rd edition, 2000.

[6] WOLFGANG, Mark. *Host Discovery with Nmap*. Novembro de 2002, disponível em http://www.moonpie.org, em 02/2003.

[7] TANENBAUM, Andrew S. *Redes de Computadores.* Editora Ccampus, 3rd edition, 1997.

[8] NORTHCUTT, Stephen. *Como detectar invasão em rede*. Editora Ciência Moderna, 1 rd edition, 2000.

[9] MATTA. *IP Network Scanning & Reconnaissance.* Technical Primer, 2002, disponível em http://www.trustmatta.com, em 02/2003.

[10] RUFINO, Nelson Murilo. *Segurança Nacional.* Editora Novatec, 1 rd edition, 2001.

[11] ARKIN, Ofir. *Network Scanning Techniques, Understanding how it is done.*

PubliCom (Communications Solutions), novembro de 1999, disponível em http://www.sys-security.com,em 01/2003.

[12] SANTOS, André. *Tópicos Avançados em TCP/IP: Parte 1*. Dezembro de 2002, disponível em: http://www.secforum.com.br/textos/, 02/2003.

[13] GOLDSMITH, David e SCHIFFMAN, Firewalking. *A Traceroute-like Analysis of IP Packet Responses to Determine Gateway Access Control list.* Outubro de 1998.

[14] ARKIN, Ofir. *ICMP Usage In Scanning, The Complete Know How.* Junho de 2001, disponível em http://www.sys-security.com , em 01/2003.

[15] SHEMA, Mike e SCAMBRAY, Joel. *Hacking Exposed Web Applications.* Mc Graw Hill / Osborne, 1 rd edition, 2003.

[16] KUROSE, James e ROSS, Keith. *Redes de Computadores e a Internet: Uma Nova Abordagem.* Editora Addilson Wesley, 1 rd edition, 2003.

[17] McCLURE, Stuart. *Web Hacking Attack and Defense.* Editora Addilson Wesley, 1 rd edition, 2003.

[18] SCHIFFMAN, Mike. *Hacker´s Chanllenge.* Mc Graw Hill / Osborne, 1 rd edition, 2003.

[19] GEUS, Paulo Licio de, e NAKAMURA, Emilio. *Segurança de Redes.* Editora Berkeley, 1a Edição, 2002.

[20] STREBE, Matthew e PERKINS, Charles. *Firewalls*. Editora Makron Books, 1a edição, 2002.

[21] SHEMA, Mike. *Anti-Hackers - Toolkit.* Mc Graw Hill / Osborne, 1 rd edition, 2003.