SANDRO MELO

COM CAPÍTULOS ESPECIAIS DOS PROFESSORES:
ALLAN PRESSI, CARLOS MARCELO, JOSÉ LUTIANO E ROGÉRIO CHOLA

EXPLORAÇÃO DE VULNERABILIDADES EM REDES TCP/IP

3ª EDIÇÃO
REVISADA E AMPLIADA

ALTA BOOKS
EDITORA
Rio de Janeiro, 2017

SUMÁRIO

SUMÁRIO

INTRODUÇÃO

xxxiv Técnicas de invasão vs conceito de pentest

CAPÍTULO 1

3 Anatomia de um ataque

6 O verdadeiro perigo é maior do que imaginamos!

CAPÍTULO 2

11 O footprint

14 Engenharia social

15 Footprint usando a base whois

18 Scanners de banners

19 O Grabbb

22 Contramedidas

CAPÍTULO 3

29 Footprint em SMTP

35 Contramedidas

CAPÍTULO 4

39 O fingerprint

41 Detecção de sistema operacional via Nmap

51 SO de sites populares

CAPÍTULO 5

63 Levantamento de dados por meio de port scanners
64 Técnicas de varreduras
65 Técnicas de varreduras clássicas
68 Varreduras TCP Connect
71 Varreduras TCP SYN
74 Varreduras baseadas na RFC 793
79 Microsoft e a RFC 793
80 Varreduras UDP
83 Varreduras para detecção de firewalls
90 Formas furtivas de varreduras
92 Varreduras temporizadas
100 Varreduras RPC
106 NMAP NSE – scripts arrojados
108 Sintaxe de uso do Nmap Scripting Engine
126 Conclusão sobre o recurso de scripts NSE
126 Contramedidas

CAPÍTULO 6

131 Sniffers – Captura de informações
131 Explorando redes com sniffers
133 Ferramentas
136 Outros sniffers
138 Contramedidas

CAPÍTULO 7

141 Backdoors – Conceitos básicos
144 Exemplificação de backdoor com Suidbit
158 Backdoor com Netcat e suas variantes
162 Usando o NSSL
163 Backdoor por meio de SMTP
169 Contramedidas

CAPÍTULO 8

175 Ataques de força bruta (bruteforce)
179 Crackeando senhas (ataques de dicionário)
187 Hydra e Medusa
188 Opções especiais para alguns módulos
195 Medusa
209 Bruteforce em HTTP
210 Conclusão sobre ataques de força bruta
210 Enumeração via bruteforce em SMTP

CAPÍTULO 9

220 Deny of Service
223 Ataque DoS 1234
241 Conceituando o DDoS
242 Sobre o ataque DDoS
246 Ferramentas de DDoS
249 Contramedidas

CAPÍTULO 10

253 O projeto Ettercap
258 Captura de informações
259 Detectando portas abertas
263 Contramedidas

CAPÍTULO 11

267 Análise de vulnerabilidade
268 Como surge uma vulnerabilidade?
268 Registro de segurança em relatórios de análise de vulnerabilidade
271 Ferramentas para análise de vulnerabilidade
271 Nessus

281 Execução do Nessus modo CLI
292 Nikto – Varreduras de vulnerabilidades web

CAPÍTULO 12

297 Exploração de vulnerabilidades
302 Metasploitable
304 Exploração do serviço FTP
306 Análise com Nessus CLI
307 Exploração via Metasploit Framework
310 Exploração do serviço SSH
313 Análise de vulnerabilidade com Nessuscmd
346 Exploração do recurso JAVA RMI
360 Conclusão pontual
367 Armitage
370 Módulo (exploit)

PROF. JOSÉ LUTIANO I

376 Pentest em redes VoIP
390 Information Gathering (Coleta de informações)
399 SVWAR
400 Enumiax
402 Monitoramento de tráfego de chamadas e escutas (*Monitoring traffic and eavesdropping phone calls*)
404 Vomit
405 UCSniff
407 Modo de aprendizagem MITM
410 Ataque de autenticação (Attacking Authentication)
412 Dictionary attack
413 DoS – Denial of Service (Negação de serviço)
416 Scanning SIP Enabled Devices

PROF. JOSÉ LUTIANO II

420 Introdução ao PTaaS (Pentesting as a Service) na nuvem
438 Teste de penetração na AWS (Amazon)

PROF. ROGÉRIO CHOLA

442 WLAN – Exploração de vulnerabilidades e provas de conceito
444 Subvertendo uma rede WLAN
446 Descoberta de redes WLAN
457 Ataques de negação de serviço
466 Ataques de força bruta em WPS
467 Método 1 – Push Button Connect ou PBC
468 Método 2 – Personal Identification Number ou PIN – Interno
469 Método 3 – Personal Identification Number ou PIN – Externo
470 Falha de arquitetura WPS – #1
471 Arquitetura de autenticação IEEE802.11/EAP Expanded Type – WPS
471 Falha de Arquitetura WPS – #2
476 Quebra de criptografia WEP
480 Quebra de criptografia WPA/WPA2
484 Segurança pela obscuridade

PROF. CARLOS MARCELO

488 Fundamentação teórica
489 O navegador web
492 Interface do usuário
495 Plugins
497 Extensões
497 Segurança
498 Armazenamento
501 Abordagem do capítulo
508 Resultado obtido da QP 2

509 Ataque 1: Cross-Site Scripting (XSS) – A1
516 Ataque 2: Injeção SQL – A2
518 Ataque 3: Clickjacking – A3
520 Ataque 4: Redirecionamentos e encaminhamentos inválidos – A4
523 Ataque 5: Complementos maliciosos – A5
526 Ataque 6: Exposição de dados sensíveis – A6
528 Ataque 7: Utilização de componentes vulneráveis conhecidos – A7
530 Ataque 8: Cross-Site Request Forgery (CSRF) – A8
532 Ataque 9: Falhas nas políticas de mesma origem – A9
535 Considerações importantes
537 Considerações finais

PROF. ALLAN PRESSI

540 Montagem do laboratório de análise de vulnerabilidades e pentest
540 Criando seu laboratório
541 Instalando o Oracle VirtualBox
546 Criando nosso ambiente de ataque com o Kali Linux
565 Instalando o Metasploitable Vulnerable Machine

APÊNDICE

573 Definições de termos
576 A diferença entre hackers e crackers
582 Mundo underground

583 ÍNDICE

Um pouco de história

Inicialmente, fiz parte de um grupo de usuários iniciantes em Linux que, em sua busca pelo saber, encontrava-se em constante reciclagem de conhecimentos sobre tecnologia de redes. Éramos sempre motivados a estudar muito, tudo relacionado a esse universo maravilhoso que é a tecnologia de redes, internet e software livre, o que foi o principal estímulo no início dessa história.

Quase no final do ano de 1997, ainda como estudante na Universidade Mackenzie de São Paulo, nosso grupo buscava apoio de parceiros para eventos e atividades acadêmicas relacionadas ao sistema operacional (SO) Linux. Na época, ainda éramos estudantes responsáveis pelo Diretório Acadêmico de Tecnologia (DATEM/DACOMPI), onde realizamos, entre tantas outras coisas, o primeiro Install Fest Linux da cidade de São Paulo. O evento, embora tenha sofrido uma tentativa de boicote, foi realizado com sucesso. Também construímos o primeiro laboratório para os alunos com dual boot: na época, o Linux ainda era uma novidade e não fazia parte dos temas abordados nos cursos, o que mudou no ano seguinte.

Não se pode afirmar que as ações realizadas pelo DACOMPI foram motivadores exclusivos para as mudanças, pois a alteração na grade curricular com adição do Linux e outras tecnologias livres seria uma consequência. No entanto, é fato que as ações do diretório acadêmico contribuíram de alguma forma.

Nesse mesmo ano foram realizadas diversas atividades a favor do software livre. Dessa forma, é importante registrar o agradecimento pelo apoio recebido de algumas entidades, como a Conectiva, representada na época por Sandro Henrique e Maruen, os quais vinham de Curitiba para ministrar palestras à comunidade mackenzista, e a Unicamp, representada por Rubens Queiroz. Bons e áureos tempos!

No ano seguinte, em 1998, o trabalho continuou com a formação de um grupo de estudos de docentes do Senac SP, onde percebemos que existiam várias ideais em comum e que a maioria dos envolvidos era entusiasta do software livre e também que podiam fazer muita coisa sozinhos, mas era possível fazer muito mais e melhor somando-se forças. Foi então que, com o apoio do Senac, o grupo de professores passou a se reunir todos os finais de semana e a estudar para aperfeiçoar ainda mais o conhecimento. Nesse meio tempo, durante um bate-papo, Sandro Melo sugeriu o nome 4Linuxmen (Os 4 Homens Linux) para o grupo, sem a pretensão de que a ideia fosse além em um primeiro momento, pois o objetivo inicial era o desenvolvimento pessoal do conhecimento.

Contudo, um grupo de estudos, por melhores que fossem seus ideais, ainda não atendia aos nossos anseios e motivações. Então, mais uma vez, a vontade de mudança nos motivou a seguir um novo desafio. O grupo de estudos esboçou ideias e modelos de negócio, ainda que iniciais, e partiu em busca de recursos e investidores para criar uma empresa.

Com o crescimento do uso do Linux no mundo corporativo, demandas por consultoria começaram a surgir, e a oportunidade de prestar serviços nesse sistema para clientes consolidou a cada dia o sonho de uma empresa de tecnologia de código aberto, a qual, na época, tinha como grande exemplo no Brasil a Conectiva.

No início, a consultoria era somente de serviços básicos, mas atualmente a demanda é bem mais exigente, incluindo soluções de infraestrutura escaláveis, desenvolvimento e treinamento especializado.

Estamos dia a dia desbravando o mercado, ministrando cursos e palestras em eventos importantes, como FISL (RS), Linux 2000 (SP), Comdex (SP), Sepai (PA), Linux Day Enterprise (PR), ExpoSalt (RJ) e Linux Expo Brasil (SP). Também estamos em grandes universidades, como a Universidade Federal de Itajubá (MG), Universidade Federal de Lavras (MG), Universidade de Alfenas (MG), Universidade Federal Fluminense (RJ), Instituto Metodista Bennett (RJ), Faculdades Integradas Maria Thereza (RJ), Universidade Mackenzie (SP), Faculdade Sumaré (SP), Universidade Anhembi-Morumbi (SP), Faculdade de Tecnologia de São Paulo (SP), Universidade Nove de Julho (SP), entre tantas outras de igual importância para a comunidade brasileira. Dessa forma, foi criada uma identidade particular na comunidade de software livre, o que trouxe importantes realizações e também um pouco de saudosismo, pois tudo começou em uma universidade, em um ambiente realmente propício para novas ideias.

Foi então criada uma empresa para trabalhar com soluções de código aberto para segurança de redes de interconectividade, além de treinamento personalizado para o cliente de acordo com o que foi implantado, ou seja, Linux de distribuição Red Hat, Debian, Slackware, TurboLinux e Conectiva. Em laboratório, são usadas as principais distribuições, como SUSE, Mandriva (oriunda da junção da distribuição Mandrake com Conectiva), Fedora, Cent OS, Ubuntu, Slackware, Yellow Dog e PPC (ambas para Macintosh), entre outras, e também sistemas operacionais Unix da família BSD (Net, Free e Open), Mac OS e Darwin.

Essa é uma empresa formada por entusiastas por tecnologia, mas conscientes como profissionais no que tange à realidade das empresas. Temos a missão de agregar valores a cada projeto, tendo como princípio e meta o sucesso do negócio dos clientes. Mesmo com tanta atividade na empresa, nunca foi abandonada a ótima relação com o lado acadêmico. Ainda são realizadas palestras e também peças teatrais em entidades acadêmicas, e, sempre que possível, grupos de usuários e eventos de software livre são patrocinados.

Falar desse começo é lembrar de todas as pessoas que, direta ou indiretamente, contribuíram e continuam contribuindo de alguma forma para essa história, como Massaro (Senac-SP), Fernando Lozano (na época atuava na LPI-Brasil), Michele Rangel (na época atuava na DTCOM), Rubens Queiroz (Unicamp), Ronaldo Lages, Mario Teza e Marcelo Branco (todos do FISL), Sandro Henrique e Maruem (Conectiva), entre tantos outros nomes.

O porquê deste livro

Com a quantidade cada vez maior de dados armazenados, a internet está se tornando cada vez mais "das coisas" e dos usuários, com o Byod e os recursos providos pela computação nas nuvens ratificando que o perímetro da rede está nas pessoas, e, por fim, com o número crescente de ameaças cibernéticas também se fazendo presente, a segurança computacional é obrigada a evoluir para responder à altura.

Hoje, projetar segurança e ser capaz de testá-la são requisitos obrigatórios, bem como saber responder a um incidente de segurança. Dessa forma, projetar segurança, testá-la e responder a incidentes são habilidades requisitadas aos profissionais de segurança dos novos tempos.

Muitas vezes, pensa-se em soluções sem ao menos avaliar o que realmente se deseja defender ou até mesmo sem definir o que se quer defender. É praticada a "segurança de olhos vendados", quando, erroneamente, acredita-se que a ferramenta resolve todo os problemas, sendo que, na realidade, a segurança é algo muito mais amplo, que abrange processos, pessoas e ferramentas, as quais são o meio, e não o fim.

Este livro tem o objetivo de ser uma contribuição para ajudar na capacitação de administradores de redes dos novos tempos e profissionais que iniciam seus estudos sobre segurança computacional, demonstrando de maneira prática e direta algumas das técnicas utilizadas por grupos de script kiddies e crackers comumente conhecidas no mundo underground e que se tornaram populares pela internet em várias comunidades de usuários, mas que também podem ser usadas para avaliar a segurança de uma empresa. Assim sendo, este livro trata da temática de técnicas e ferramentas para testes de segurança, com foco em infraestrutura e serviços de rede.

Ter a capacidade de realizar testes de segurança, em especial pentest (do inglês *Penetration Test*), é importante, uma vez que possibilita ter uma visão de quanto de vulnerabilidade tem a infraestrutura do sistema e como essas vulnerabilidades podem se tornar uma ameaça. Dessa forma, pode-se desfazer muitos mitos e mostrar que a solução é ou não a segurança, além de ser capaz de quantificar e avaliar o risco assumido.

Quando a suíte TCP/IP foi inicialmente projetada, o cenário era de uma rede privada militar, na qual cada estação era uma base militar, fazendo do roteamento uma necessidade. Essa rede militar saiu da estrutura de teia e subiu aos satélites, tornando a internet disponível a órgãos de pesquisa americanos, como o MIT (Instituto de Tecnologia de Massachusetts), entre outros, e depois a todo o mundo.

Naquele cenário inicial da pós-Arpanet, o problema de segurança de redes de computadores podia ser resolvido inicialmente com simples filtros de pacotes. Embora isso fosse realmente simples para a realidade da época, era o suficiente na maioria dos casos.

Atualmente, a segurança de informação é algo extremamente necessário, mas, infelizmente, como qualquer investimento em TI (Tecnologia da Informação), pode não ser aparamente tangível em um primeiro momento. No entanto, por outro lado, ter o nome de sua empresa manchado certamente não será agradável para nenhum acionista ou investidor. Em outras palavras, pode parecer inicialmente difícil mensurar a necessidade de um investimento em segurança, mas com certeza não será difícil quantificar o prejuízo da marca após ter um site alterado por script kiddies.

Quando falamos em segurança de informação, temos que pensar na empresa como um todo, em todos seus ativos. Entretanto, quando pensamos em soluções computacionais, temos alguns paradigmas a escolher no que tange à escolha de ferramentas. Podemos deixar toda a segurança de nossa empresa à mercê do melhor produto de segurança baseado em soluções caixa-preta, que, segundo a maioria dos vendedores, são autossuficientes e nem precisam de um profissional para utilizá-las, bastando ligá-las na tomada para que sua empresa esteja segura. Também podemos pensar em soluções baseadas em software livre ou código aberto, situação em que o profissional tem que saber o que está fazendo, pois a tecnologia é confiável, transparente e totalmente auditável.

O mais preocupante é que um cracker é justamente o oposto do perfil de uma solução caixa-preta: ele vai usá-la e se atentar a detalhes que você, com uma solução caixa-preta, nem imagina que existam.

É difícil, como profissional de segurança, aceitar esse ideal de que a ferramenta (software/hardware) baseada na obscuridade do código fechado é tudo. Se isso fosse verda-

deiro, não teríamos sistemas de código fechado entre os mais invadidos, prova concreta e incontestável de que problemas de segurança não se resumem em ferramental.

A falsa propaganda que algumas empresas de produtos proprietários fazem é que o código aberto é mais fácil de ser invadido, visto que o cracker tem acesso a como ele foi implementado. Isso não é justificativa, pois existem códigos mal elaborados tanto como software livre (ou Open Source2) quanto como caixa-preta. Da mesma forma, também encontramos em ambos ferramentas bem elaboradas. Todavia, é relevante lembrar que, no mundo do código aberto, um código malfeito não fica muito tempo sem que a comunidade de programadores questione sua validade e qualidade.

Hoje, por meio das estatísticas do grande volume de ataques, vemos que, na grande maioria das pichações de sites, o que faz a diferença não é apenas a melhor solução de firewall, mas o profissional que está por trás dela. O peopleware, embora sempre tenha sido, agora é reconhecido como o diferencial. Não podemos assumir que o melhor ferramental é tudo o que se necessita para um projeto de segurança. Quando pensamos em segurança de sistemas, temos uma fórmula algébrica de recursos: capacitação (técnicos, usuários) + metodologia (normas, boas práticas, planejamento) + ferramental = bom projeto de segurança

Essa fórmula simples deve ser entendida como um exemplo ilustrativo em que cada variável explodirá em vários outros itens.

Este livro, portanto, é muito importante na medida em que focará o aspecto técnico do ferramental, ou seja, as vulnerabilidades e as técnicas de intrusão nos sistemas, objetivando ser um veículo de capacitação. Contudo, em momento algum temos a pretensão de ser uma literatura definitiva sobre a segurança de sistemas.

Não existe uma ferramenta autossuficiente que permita que o administrador não seja um especialista em tecnologia de redes de computadores. Talvez no futuro, ferramentas baseadas em inteligência artificial sejam capazes de identificar assinaturas de ataque, podendo até ser pró-ativas de forma objetiva e eficiente. Todavia, isso ainda não é possível, e é lamentável ouvir e ver profissionais oferecendo soluções milagrosas baseadas exclusivamente em ferramentas. Ainda mais triste é que muitos administradores que não têm método e nem a capacitação adequada assumem soluções mal implementadas, não fundamentadas em normas, criando um cenário de falsa segurança, que é justamente o que fará a diferença para um invasor.

Hoje temos que ter a consciência de como o invasor pode atuar em nosso site usando as técnicas mais furtivas. Já dizia Sun Tzu: “Se conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas”. Assim sendo, desconfie (primeiro do vendedor, depois do produto) quando um vendedor de firewall bater à

porta de sua empresa com uma solução miraculosa, dizendo que qualquer um seria capaz de administrar o produto devido ao fato de possuir uma interface intuitiva, tornando qualquer profissional um analista de segurança capacitado, desde que conheça a arquitetura "next, next, finish".

Saiba que os invasores que podem levar perigo à sua empresa não se contentam apenas em trocar a sua homepage, pois são capacitados e, na maioria dos casos, são ótimos em programação e/ou engenharia de redes. Quase sempre autodidatas, utilizam seus conhecimentos para causar danos a corporações. Dessa forma, para ser capaz de se defender, você terá que ter a mesma arma: o conhecimento tanto das vantagens das ferramentas que são utilizadas quanto de suas limitações. Entretanto, o que fará a diferença é justamente ter o conhecimento das técnicas comumente utilizadas pelos invasores, que, segundo eles mesmos, são as técnicas não éticas de seu submundo, o mundo underground, para que as elaborações de contramedidas sejam concisas e verdadeiramente funcionais.

Pensando dessa forma, a empresa 4NIX (www.4nix.com.br) inovou o mercado, desenvolvendo uma formação de segurança que a trata sem tabus, de forma verdadeira e com apoio jurídico especializado, já que a divulgação não ética e indiscriminada de técnicas de intrusão de sistemas pode caracterizar apologia a esse tipo de crime, o que também é crime.

Este livro é baseado em parte do material desenvolvido para um dos cursos que visam a formação consciente de administradores de sistemas. A metodologia é baseada na ideia de que, por meio de um forte embasamento teórico e de uma carga horária que permita exercícios práticos, é possível mostrar a um administrador as técnicas utilizadas por script kiddies e crackers, permitindo uma capacitação diferenciada para os administradores de redes dos novos tempos.

Dessa maneira, quando o aluno passa a estudar técnicas e ferramentas para segurança, é capaz de entender até onde elas podem atuar, pois poderá validá-las em laboratório usando o conhecimento adquirido. Além disso, quando um aluno desenvolver políticas para um firewall, ele será capaz de se defender de qualquer ataque que seu firewall seja capaz de tratar, pois terá consciência do que realmente estará sendo defendido.

Conhecendo, aprendendo e entendendo o risco de cada técnica, o aluno será capaz de aplicar esse conhecimento para validar um projeto de segurança em ambiente de laboratório, criando um cenário próximo ao da realidade dos ataques de invasores. O sucesso dessa proposta nos possibilitou treinar vários grupos de CyberCops, policiais de delegacias especializados em crimes digitais ou crimes por

meios eletrônicos, em São Paulo, no Distrito Federal, no Rio de Janeiro e em Belém do Pará, validando a qualidade do conhecimento que os cursos agregam.

É notório que um bom administrador de redes dos novos tempos tem que conhecer bem o que faz, conhecer bem o protocolo TCP/IP, inclusive suas limitações, para que seja capaz de configurar corretamente um firewall e um sistema de IDS e até reconhecer uma assinatura de ataque. Por melhor que seja a ferramenta, a capacitação do profissional será o diferencial, além de sua atividade em busca de atualizações constantes. Sempre que possível, o administrador de redes deve testar as vulnerabilidades dos sistemas por ele utilizados para buscar erros de configurações padrões, bem como submeter seu sistema a um teste de um possível exploit, ou seja, à realização de pentest3, que também é um mecanismo muito importante para confrontarmos a segurança empregada.

Se observarmos as estatísticas, veremos que o Brasil é o país número um em invasões e em invasores. Além disso, grande parte das invasões explora vulnerabilidades conhecidas em sistemas de código fechado, e a maioria dos administradores das empresas invadidas nem imagina como a invasão ocorreu. Todavia, invasões em sistemas abertos também ocorrem, mas as alterações são muito dinâmicas, pois, ao contrário do modelo proprietário, a correção não está exclusivamente na mão do fabricante. O desenvolvimento de ferramentas destinadas a aumentar a segurança de soluções de código aberto é contínuo, deixando claro que essa estatística está ligada, na maioria das vezes, a falhas de procedimento ou capacitação, e não à tecnologia empregada.

Não podemos deixar de citar que muitas das soluções de segurança abertas são base para soluções proprietárias, provando que, em momento algum, o fato de não existir custo na aquisição da ferramenta ou tecnologia faz de sua tecnologia proprietária algo inferior. Além de padrões abertos como IPSEC, que são a base para soluções fechadas e abertas, temos muitos sistemas de firewall caixa-preta, que, por exemplo, rodam em cima de kernel baseado em sistemas Linux ou BSD.

Existem profissionais que, erroneamente, elaboram projetos de segurança com um NAT em um firewall de fronteira e os colocam como solução salvadora, acreditando e vendendo a ideia de que um servidor atrás de um firewall via NAT estaria seguro, sem levar em consideração o que faz ou não um firewall. Crackers com bons conhecimentos facilmente tirariam do ar um servidor nessas condições, levando em consideração o tipo de firewall e o serviço disponível via NAT. Para termos um bom entendimento, vamos fazer um estudo de caso.

Imagine que está sendo usado um Firewall Packet Filter ou um Stateful Firewall com configurações básicas e um servidor de correio Exchange. Um cracker não pen-

saria duas vezes e faria um ataque SYN Flood na porta 25 em loop, tirando o serviço do ar se ele fosse vulnerável a tal investida ou se houvesse um IIS vulnerável à manipulação arbitrária de Unicode. Isso seria possível pelo simples fato de que, mesmo estando atrás de um firewall, a conexão por meio da respectiva porta do serviço estaria disponível no NAT, ou seja, o NAT por si só não faria diferença caso aquela conexão e seus dados não fossem tratados, fossem simplesmente redirecionados.

Indo um pouco mais longe, imagine quantos servidores IIS foram literalmente detonados pelos últimos ataques do CodeRed, CodeBlue, Nimda e outras pérolas digitais! E o mais interessante: muitos estavam nessa topologia de NAT, e, o que é pior, os administradores se questionavam como o ataque obteve sucesso em seus servidores sem ao menos lembrarem do conceito básico de uma conexão. O mais triste é o fato de o NAT estar dentro da LAN: o ataque, nesses casos, prejudica a performance da LAN. Apesar de o NAT fazer referência ao IP do servidor, o enlace físico é o mesmo. A falta de conhecimento dos administradores em relação ao mundo real é um dos fatos mais preocupantes, ainda mais se somada às atitudes de empresas e pseudoprofissionais que vendem soluções apenas mostrando as vantagens, mas não suas limitações, vendendo uma doce ilusão de segurança, esquecendo ou simplesmente omitindo que são exatamente essas limitações que os crackers exploram.

Tudo isso mostra o quanto o fato de não conhecermos o lado não ético da rede nos limita. Quando se conhece o inimigo, o combate é muito mais fácil. Embora a imaginação dos script kiddies, insiders e crackers seja algo formidável, nós, administradores, podemos lutar em igualdade desde que conheçamos o terreno onde pisamos, para que possamos desenvolver uma solução ideal. Todavia, por melhor que seja a solução, ela não será completa, e sempre haverá um risco assumido.

Por que Linux? Por que software livre?

Embora este livro seja focado em segurança computacional, o Linux utilizado para compor distribuições como Backtrack e Kali, que reúnem um grande número de ferramentas para pentest, é também uma grande fonte de referência e inspiração. Hoje o Linux é notavelmente um sistema operacional poderoso, sendo o preferido de muitos hackers, crackers e script kiddies. Ele tem como concorrente direto em capacidade os Unix BSD*, que também são softwares livres. Eles são regidos pela GPL e pela licença BSD, respectivamente. Então por que não usar esse poder ao nosso favor? O Linux e as ferramentas FOSS de segurança podem e devem ser usados como meio para avaliar a segurança da infraestrutura de serviços de redes de uma determinada empresa.

O Linux é um ótimo exemplo de ferramenta livre fruto de um desenvolvimento coletivo de vários programadores ao redor do planeta. O Linux é como a matemática: não tem dono, é de todos. Agora imagine se tivéssemos que pagar licença para cada cálculo que fizéssemos: será que a humanidade evoluiria tanto? Imagine se a teoria da relatividade, desenvolvida pelo gênio Einstein, fosse um código fechado: será que a física teria evoluído da forma que evoluiu? O Linux é muito mais que um SO, pois não tem precedentes na história da TI; é a liberdade tecnológica.

Vários países estão utilizando o Linux e montando suas próprias distribuições. Imagine o quanto seria economizado se todas as repartições públicas, unidades militares e escolas utilizassem softwares livres. Embora, para muitos, o Linux não tenha interfaces tão intuitivas e agradáveis quanto as do Mac OS X (cujo kernel é baseado em BSD), como sistema operacional nos servidores, é uma solução já consagrada e supera a maioria dos outros sistemas em performance e qualidade computacional. Nas estações de trabalho, a cada dia é mais capaz de atender às necessidades básicas de qualquer usuário. No entanto, a batalha nos desktops ainda será muito longa!

Agora acontece diferente do que ocorreu nos anos 1980, quando a reserva de mercado tentou trazer para o nosso país essa liberdade, o que fracassou, pois acabou brevemente, destruindo o sonho tecnológico de grandes empresas que desenvolviam a base da tecnologia tupiniquim, pois, em muitos casos, a tão sonhada tecnologia nacional é fruto da engenharia reversa de projetos europeus e americanos.

O novo capítulo escrito pelo software livre, em especial o Linux, traz a oportunidade de termos a tecnologia nacional somada a tantas outras boas ideias oriundas dos quatro cantos do planeta. O Linux não nos isola do mundo; pelo contrário, ele nos deixa mais perto de todos, devido ao fato de que no mundo do software livre existem profissionais de todo o planeta, além de ser livre, ou seja, há a liberdade de saber como foi feito, a liberdade de poder modificar e ajustar as necessidades e a liberdade de distribuir e utilizar. Outro ponto importante do código aberto é que ter acesso ao código-fonte admite uma real e total auditoria sobre ele, permitindo ao nosso país, um país de terceiro mundo, a liberdade tecnológica que tanto desejamos e da qual precisamos no segmento da computação.

Traçando um outro paralelo, imagine se a grandiosa IBM tivesse patenteado o PC: será que a evolução que ocorreu a partir do momento em que ele passou a ser usado por todo o mundo teria ocorrido da mesma forma? É difícil saber, mas, por outro lado, ainda assim um dia os computadores estariam dentro dos lares, mesmo que isso não ocorresse na mesma velocidade em que ocorreu. Ninguém pode negar a

evolução tecnológica que o PC trouxe à humanidade no final do século XX. Veja só: o PC tem seu código aberto (projeto!), e não vemos nenhuma empresa de software comercial reclamar disso.

Também devemos levar em conta que a segurança que o código aberto traz é admirável, pois, ao estar à mostra para qualquer um que queira auditá-lo, o código pode ser questionado quanto a sua eficiência e, por consequência, estará sempre se aperfeiçoando. Além disso, um projeto de código aberto não tem necessariamente sua qualidade vinculada a um fluxo de caixa. Por outro lado, no código fechado, só o fabricante tem a real certeza do que consta em seu produto, e, dessa forma, qualquer correção que seja necessária depende somente dele. Um exemplo da ineficácia desse modelo são os históricos ataques do CodeRed, CodeBlue e Nimda: quantos servidores pelo mundo não foram aniquilados devido ao fato do path do fabricante não ter vindo em tempo hábil para evitar o volume de prejuízos e transtornos causados por essas pragas digitais? No modelo de código fechado, o tempo da janela de exposição de vulnerabilidade depende da velocidade do fabricante em disponibilizar a correção.

O código fechado não é 100% auditável. Todo profissional de computação sabe que, sem o código, mesmo fazendo uma engenharia reversa, um software de código fechado não pode ser 100% auditável, já que essa não é uma tarefa tão simples quando a aplicação tem um tamanho considerável. Além disso, devemos levar em consideração que um procedimento desses também não poderia ser feito devido às leis autorais, restando apenas confiar cegamente na utópica bondade humana. Será que é correto para uma unidade militar ou governamental ficar dependente tecnologicamente de um fabricante, principalmente se o fabricante for de outro país? Assim, acabamos entrando em questões de soberania nacional.

Outro importante ponto de vista é relativo a um bom administrador Linux ou de outro sistema Like Unix, como Solaris, FreeBSD ou OpenBSD. Ele realmente deve saber o que está fazendo, para, dessa forma, ter acesso ao código e fazer ajustes finos em uma determinada solução.

Existe um grande número de ferramentas de redes livres que permitem a um bom administrador fazer uma análise detalhada de seus ambientes e do tráfego de sua rede. Uma ferramenta de código aberto permite ao administrador, desde que haja capacitação, personalizar o sistema de acordo com a sua solução, e não o contrário. Ele não tem tela azul (Blue Screen) e não possui suporte a plugins especiais, como Melissa, ILOVEYOU, SirCam, CodeRed, CodeBlue, Nimda, entre tantos outros.

A possibilidade de vírus ou mesmo outros tipos de Malware também é uma realidade para sistema Like Unix, mas é muito mais difícil um vírus se propagar em

um sistema Unix, devido ao fato de ter sido gravado no sistema de arquivo, pois em sistema Unix não há extensão do nome do arquivo para considerar o arquivo como executável. Demanda-se dar direitos de execução qualquer a um arquivo antes de executá-lo, e mesmo que seja gravado com o máximo de direitos possíveis, no momento da escrita no sistema de arquivo, jamais um arquivo recebe direito de execução, devido a uma barreira chamada umask.

Isto é apenas um exemplo simplista da qualidade e segurança que o sistema Unix traz consigo. Dessa forma, deve-se dar os méritos a sistemas que seguem esse modelo, como o Linux e Unix BSD (Free, Open, Net entre outros). Sendo isto mais um argumento de defesa de que esses sistemas são ótimas alternativas.

Devemos também considerar o contexto deste livro, pois esses sistemas também podem propiciar um ambiente adequado para teste de segurança como Pentest, devivo à grande quantidade de ferramentas de segurança disponíveis e distribuições Linux customizadas com Kali Linux.

O uso dos termos "hacker", "cracker" e "script kiddie"

Durante o livro, utilizaremos o termo "hacker" em sua essência. Nos primórdios da internet, "hackear" era o ato de forçar a porta do CPD para ter acesso aos computadores no MIT. Com o tempo, o termo passou a ser associado a usuários avançados, e hoje, erroneamente, a mídia o associa a criminosos cibernéticos. Segundo um dos hackers mais famosos do mundo, Richard Stallman:

> *O uso da palavra hacker para se referir ao violador de segurança é uma conclusão que vem por parte dos meios de comunicação de massa. Nós, hackers, nos recusamos a reconhecer este significado, e continuamos usando a palavra para indicar alguém que ama programar e que gosta de ser hábil e engenhoso.*

Será que é justo denominar Stallman, Linus Torvalds, Arnaldo Melo, Marcelo Tossati, Alan Cox e tantos outros excelentes hackers da mesma forma que denominamos um jovem que, na emoção da internet, comente o delito de invadir e danificar o sistema de uma empresa? Claro que não! Hackers constroem e tornam o mundo melhor; hackers não são fúteis desconfiguradores de páginas. A ética hacker é questionada pela sociedade, visto que, muitas vezes, hackers são ativistas que confrontam conceitos e governantes. No entanto, é justamente esse questionamento, esse confrontamento, que a evolução humana necessita para que haja um mundo melhor.