Scribus
Editoração eletrônica no Linux

AdBlock
O fim da propaganda na Internet

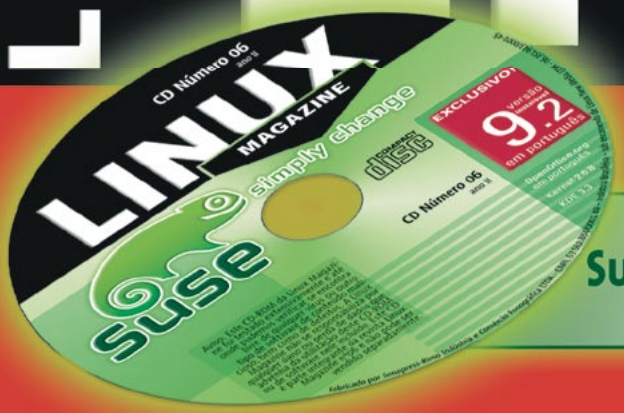
LINUX MAGAZINE NÚMERO 6

EXCELÊNCIA EM MATÉRIA DE LINUX

LINUX MAGAZINE

exemplar de cortesia **não contém CD**

NÚMERO 6



CD EXCLUSIVO: SuSE 9.2 INSTALÁVEL EM PORTUGUÊS!

Mantenha-se a salvo!

SEGURANÇA

Implemente uma bridge

Gerencie regras de firewall

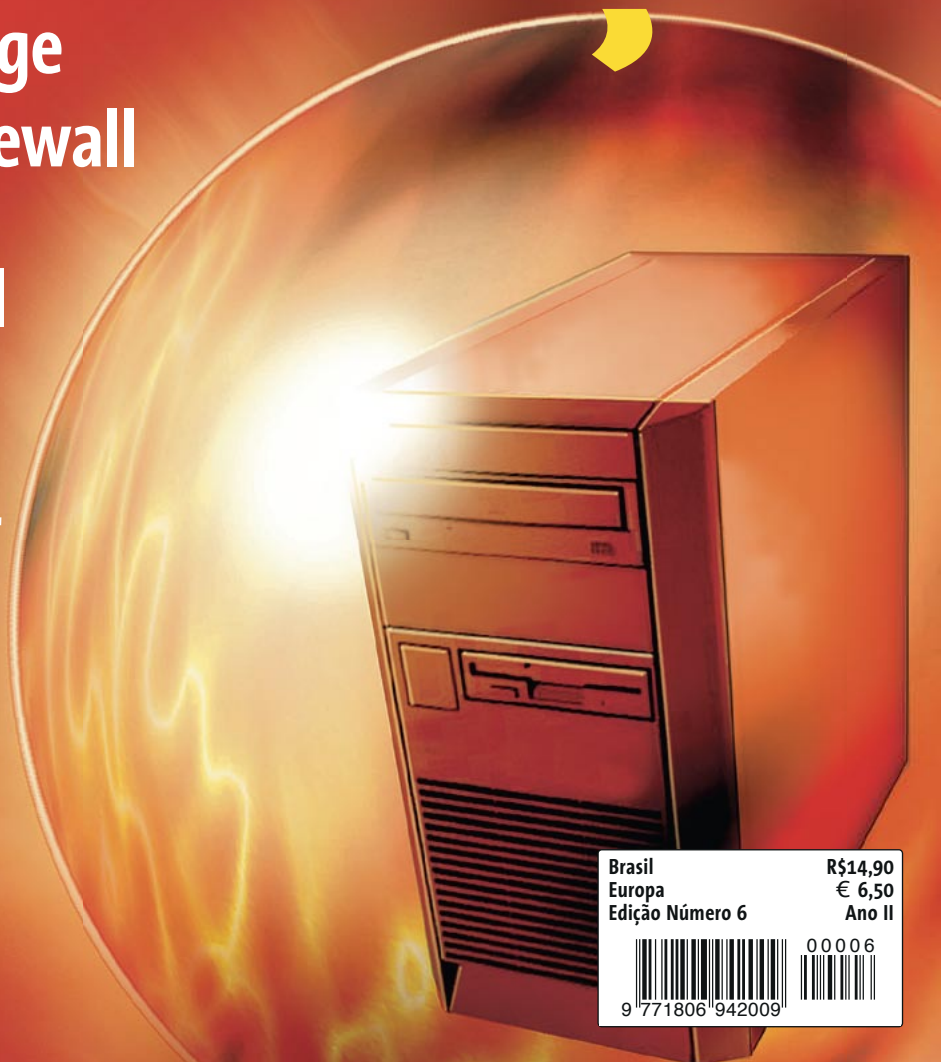
Firewall fácil com Guarddog e Shorewall

Firefox

Saiba mais sobre o navegador que está agitando a Internet

Veja também:

GUIs para Firewalls
Proteja a memória com PaX
Linux em um "chaveiro" USB
Fluxbox: leve, rápido, prático
Tutorial de compilação cruzada



Brasil R$14,90
Europa € 6,50
Edição Número 6 Ano II
00006
9 771806 942009

Segurança Bridgewall Firewall GUIs Guarddog Shorewall Firefox PaX Linux de bolso Fluxbox SuSE 9.2 BR

Segurança e Linux

Prezado leitor, prezada leitora da Linux Magazine,

no final do mês de Agosto de 2004, durante uma coletiva de imprensa realizada no âmbito da 9ª Conferência Anual sobre o Futuro da TI, o grupo Gartner anunciou o resultado de uma pesquisa de opinião sobre qual área de TI de empresas no Brasil deveria receber atenção (e investimentos) em especial em 2005. Desenvolvida junto aos participantes do evento que ocupam cargos de direção, a resposta da grande maioria foi **segurança**.

Quando se fala em segurança, sistemas baseados em tecnologia de código aberto, como o Linux, são, via de regra, a escolha apropriada. Mas por que o Linux é tão seguro? Ou melhor, por que o Software Livre é, por natureza, normalmente mais seguro que suas contrapartes proprietárias? Não deveria ser o contrário? Afinal, se todo mundo pode ler o código fonte dos programas, é mais fácil encontrar as falhas que podem ser exploradas, não? A resposta a essa última pergunta é um sonoro NÃO! Muito pelo contrário: é justamente porque o código pode ser lido por milhares de desenvolvedores no mundo todo que pode ser auditado e provado em segurança por quem quer que deseje e disponha de conhecimento técnico para fazê-lo. Enquanto alguém está implementando um novo recurso, outros estão corrigindo erros nessa implementação, empresas a estão testando sob cenários de missão crítica e devolvendo "patches" (correções – literalmente, "remendos") que tornem a execução do programa produzido menos passível de erros e mais eficiente. É, em resumo, por essa razão que o desenvolvimento de Software Livre torna essa modalidade de programas mais eficiente, rápida e segura.

Será que isso é verdade? Ou será que só se trata de retórica? Será que o Linux não é atingido por problemas de segurança com a mesma freqüência que, por exemplo, o Windows®, sistema operacional da Microsoft, porque o sistema criado por Linus Torvalds ainda não está tão disseminado quanto o da empresa de Bill Gates & Co.? Afinal, segundo pesquisas, a participação do Linux no mercado de desktops é de menos de 5%. Não haveria interesse de nenhum "cracker" em criar um vírus para Linux, ou mesmo um worm ou cavalo de Tróia, já que o sistema não seria um alvo realmente apetitoso, certo? Mais uma vez, a resposta é NÃO. O Linux, e outros sistemas de código aberto como o servidor web Apache, constituem hoje o fundamento da Internet. A maioria maciça de servidores web no mundo lá fora é baseada no quarteto de acrônimo LAMP – Linux, Apache, MySQL e PHP, todos projetos livres. Eles são alvos muito mais interessantes que o desktop do João, do José etc., uma vez que guardam informações corporativas que podem valer muito dinheiro. De outro lado, modificar o conteúdo de tais servidores é um excelente modo de se mostrar – coisas que "crackers" sempre apreciam. Aliás, tanto isso é verdade, que basta o administrador de sistemas não ficar atento e deixar de aplicar com freqüência as correções de segurança que a sua distribuição Linux torna disponíveis diariamente para que os sistemas sob sua supervisão sejam atacados – infelizmente com sucesso. Há inúmeros casos em que isso ocorreu no passado.

Mas para agradar os céticos, vamos embasar nossa retórica com alguns números: na edição anterior da Linux Magazine, noticiamos que, segundo um estudo realizado desde o ano 2000 no Centro de Pesquisas em Ciência da Computação da Universidade de Stanford, o kernel 2.6 do Linux, em suas 5,6 milhões de linhas, apresentou uma taxa média de erros de implementação da ordem de 0,17 a cada 1000 linhas de código. Estudo semelhante, realizado pela Universidade Carnegie Melon com softwares proprietários, identificou uma média variando entre 20 e 30 bugs a cada 1000 linhas, o que coloca o kernel do Linux em uma posição absolutamente vantajosa em relação à média de sistemas semelhantes, mas de código fechado, embora comparações específicas devam ser feitas caso a caso.

Para encerrar, gostaríamos de lembrar mais uma característica do Linux – na verdade do Unix – que também contribui para que tais sistemas sejam por padrão mais seguros: a sua arquitetura. No Unix, como no Linux, há uma clara separação de privilégios. Usuários são, normalmente, incapazes de instalar qualquer programa ou mesmo apagar arquivos fora do seu diretório pessoal. Para tanto, é necessário obter privilégios de administrador. Além disso, não basta um arquivo qualquer ter a extensão EXE ou COM para automaticamente ser executado com um clique do mouse. Ele precisa ter atributos especiais que digam ao sistema: "Ei, eu sou executável, mas só o usuário 'fulano' é que pode me executar!" Isso dificulta a vida do programador de vírus. Aliado a isso some-se a velocidade com que correções de segurança aparecem para Software Livre – em contraste com o que acontece com o software de código fechado, para o qual às vezes espera-se por meses até que uma correção apareça – e temos um cenário muito mais vantajoso de utilização, o que estimula a adoção de soluções abertas por governos, empresas e vai, inevitavelmente, acabar por chamar a atenção do usuário doméstico.

Não é à toa que o projeto OpenBSD anuncia com orgulho a ocorrência de apenas uma falha de segurança explorável remotamente na instalação padrão em mais de 8 anos! Projetos como Adamantix (Trusted Debian) e Security-Enhanced Linux (SELinux) seguem a mesma linha. Nesta edição, você vai conhecer algumas das tecnologias de segurança disponíveis em sistemas abertos que, apesar de não serem visíveis, podem constituir a diferença entre a vida e a morte de sua estrutura de TI.

Rafael Peregrino da Silva
Editor

19 Firewall para leigos

Seu computador permite que você tenha uma visão do mundo, mas ninguém quer ter o mundo olhando seu computador. Os invasores estão ficando cada vez mais ousados e tecnicamente mais "afiados". Não é mais aceitável esperar que tais delinqüentes simplesmente não notem sua estação de trabalho "dando sopa". Se você está conectado à Internet, é melhor colocar-se atrás de algum tipo de firewall.

Obviamente, os firewalls mais exóticos são produtos dedicados, baseados em hardware e destinados a redes gigantescas e configurações complexas. Estamos mais interessados no que se pode fazer com hardware comum, uma distribuição Linux e alguns programas fáceis de encontrar. Este mês, trazemos uma seleção de ferramentas para construção de firewalls, além de utilitários poderosos que simplificam sua configuração e manutenção. Com elas, você não precisa ser um expert em redes ou segurança para administrar um firewall.

40 Liga pro suporte!

Ficar atento aos pedidos de suporte dos clientes pode fazer a diferença quando se trata de voltarem – ou não – a contratá-lo. Felizmente, há diversos aplicativos baseados em Linux que oferecem soluções para o gerenciamento de seu departamento de suporte técnico.

45 Tipografia e texto

Continuando nossa série de artigos sobre editoração eletrônica no Linux com o Scribus, nesse mês mostramos como lidar com imagens em CMYK e trabalhar com texto.

50 Linux é chave!

Todo mundo já está careca de saber que dá pra iniciar o Linux em um computador a partir de um Live CD. Mas quem já pensou em instalar seu sistema operacional favorito em um minúsculo chaveiro USB?

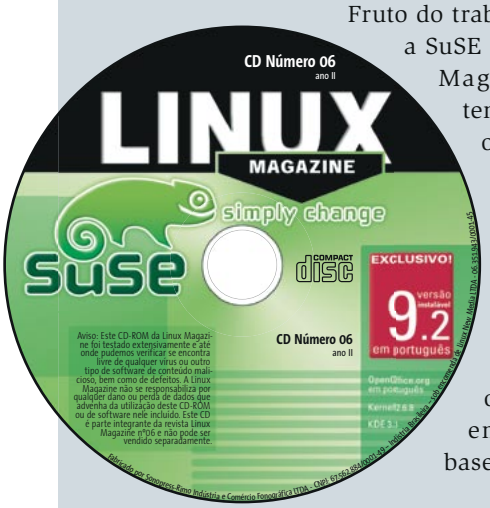
54 Regras para todos

A tecnologia de firewall presente no kernel do Linux já há algum tempo atende às exigências profissionais. Agora, os configuradores gráficos chegaram para diminuir a distância entre as soluções comerciais e livres.

61 PaX Romana

Conheça o patch para o kernel que oferece proteção contra corrupção da memória usada pelos processos e traz um pouco de paz de espírito aos paranóicos por segurança.

70 SuSE Linux 9.2



Fruto do trabalho conjunto entre a SuSE Linux AG e a Linux Magazine, este mês temos o prazer de lhe oferecer um CD com a versão nacionalizada de uma das distribuições mais cobiçadas do momento: o SuSE Linux 9.2. Instale em seu HD e aprecie o que há de melhor entre os desktops baseados em Linux.

80 A porta de entrada

Tanto o KDE quanto o GNOME possuem gerenciadores de login. Vamos olhar o que o KDM e o GDM têm a oferecer e aprender a configurar a tela de boas-vindas de seu sistema Linux.

83 Surfando sem banners

Que nos perdoem os publicitários, mas anúncios em websites podem dificultar sua navegação e atrapalhar a concentração do visitante, além de consumir muita banda. Neste artigo, conheceremos ferramentas que permitem bloquear banners, janelas pop-up, "overlays" em Flash e outras formas de propaganda, seja apenas em uma máquina ou em sua rede local inteira.

Cartas para o editor

Permissão de escrita

Sobre o Debian

» Em seu artigo sobre distribuições a Sulamita Garcia, como sempre, "mandou bem". Só gostaria de fazer uma observação quanto ao seguinte trecho:

> "Os pacotes disponíveis para a versão stable são conhecidos por serem quase pré-históricos. Se você quiser usar versões mais atuais, é instruído a usar o unstable. Espera aí, mas não é instável? Sim, o próprio nome está dizendo. E se você estiver usando o unstable e acontecer algum problema... bom, você estava usando o unstable. Quer dizer que ou uso pacotes pré-históricos ou uso instáveis? Pois é."

Ela esqueceu de dizer que a versão Sarge (*testing*) é uma boa opção para desktop, já que seus pacotes não são tão "arcaicos" quanto a Woody (*stable*) e nem tão instáveis quanto a Sid (*unstable*). Na verdade, quando um pacote passa da *unstable* para a *testing* ele já foi bem testado e o risco do sistema "quebrar" é muito baixo.

Grande parte das distribuições derivadas do Debian usam como base a *testing*, como por exemplo o Ubuntu e a brasileira Debian-BR-CDD. No canal *#debian-br* (*irc.freenode.net*) geralmente recomendamos a versão Sarge (*testing*) para uso em ambiente desktop ou em servidores não-críticos, a Woody (*stable*) para servidores críticos e a Sid (*unstable*) para quem está disposto a assumir os riscos.

Marco Carvalho

Sulamita responde: Muitos detalhes das distribuições acabam não cabendo nos artigos, que são uma visão mais geral. Só me parece que os leitores não entenderam minha crítica: ouço pessoas defendendo tanto o testing *quanto o* unstable*, dizendo que ambas são bastante estáveis. Se são tão estáveis, porque não são chamadas assim? Isto me soa como um empurra-empurra, e acho que uma comunidade desse tamanho deveria ter algo mais declaradamente estável, atual e assumir a responsabilidade por isso. Ficar eternamente "testando" não me parece uma boa idéia.* ■

SlaX

» Informo que o meu monitor, um "manjado" LG SW 500G *plug and play*, também não é reconhecido pelo Slax desde versões anteriores à distribuída pela revista. Como a ajuda não documenta as opções *xrefresh* e *vrefresh*, tive que editar na unha os arquivos de configuração de vídeo para poder rodar o sistema.

Com relação à sugestão para o leitor usar a opção *vrefresh = 60*, num monitor igual ao meu ela não irá funcionar, pois a implicância do Slax com ele se concentra no refresh horizontal. Quem tem um desses monitores e quer rodar o Slax deve usar o parâmetro *xrefresh=54*.

Flavio Lopes

Figura 1: Racer, um dos jogos distribuídos em nossa edição passada.

Mais Jogos

» Me surpreendi ao ver um CD repleto de jogos nessa nova edição. Positivamente, claro. Mas o que mais me decepcionou foi não ter encontrado na lista um jogo que, com certeza, seria de grande agrado para os gamemaníacos: *Return to Castle Wolfenstein – Enemy Territory*, produzido pela ID Software (a mesma de Doom3 e Return to Castle Wolfenstein), com o código aberto e totalmente gratuito.

Esse jogo é bastante famoso nos EUA e Europa, possui algumas centenas de jogadores aqui no Brasil e diversos cam-

peonatos são organizados durante o ano. Caso seja feita mais uma edição com jogos, gostaria que ele fosse incluso.

Caio

Valeu pela dica. Conhecemos o Enemy Territory, mas o nosso problema foi outro: espaço. Tentamos diversas composições diferentes, mas apenas o FlightGear com alguns mapas e aeroportos do Brasil já deu trabalho para acomodar junto aos outros jogos no disco (Racer, BZFlag, VegaStrike, GLTron e GL117), entre outros.

Figura 2: *Return to Castle Wolfenstein – Enemy Territory* pode ser baixado gratuitamente de http://games.activision.com/games/wolfenstein/

Fotografia Digital

» Estou aprendendo um pouco sobre edição de imagens para melhorar as fotos tiradas com uma câmera digital e notei que elas sempre precisam de um ajuste de sharpen, levels e coisas assim. Para isso uso o Gimp, e gostaria de um artigo mostrando como fazer isso: como corrigir as curvas, reduzir ruído, criar imagens panorâmicas e coisas do tipo.

Para organização e aquisição de imagens sugiro uma matéria sobre o digiKam (*http://digikam.sourceforge.net*), software para o KDE que supera até programas semelhantes para Windows. Ele não só baixa as imagens da câmera (usando a *libgphoto*) como também pode organizá-las por álbuns e "tags", que são comentários que eu mesmo adiciono às imagens. Dessa forma, uma foto pode, ao mesmo tempo, pertencer ao grupo "férias" e ao grupo "macro", sem necessidade de duplicá-la em dois lugares diferentes.

Ed Carlos de Paula

Obrigado pelas sugestões, Ed. Está programado para as próximas edições um artigo sobre ferramentas como o Hugin, o Enblend e o Autopano-SIFT, que facilitam, e muito, a criação de fotos panorâmicas. Aguarde!

Fax via Web

» Acompanhei com atenção o artigo sobre WebFax publicado na quinta edição (matéria de capa, página 38). Achei-o interessante e instrutivo; porém, como vocês mesmos disseram em nota, infelizmente não existe um serviço similar disponível no Brasil.

Daí minha sugestão: por que não publicar um tutorial mostrando como montar um provedor de Fax com Linux? Talvez com isso surjam novos provedores para essa solução, bem como empresas criadas para explorar tal mercado. Até as empresas de telefonia se beneficiariam com isso, já que de uma forma ou de outra ainda existe a necessidade das linhas telefônicas disponíveis para oferecer o serviço adequadamente.

Carlos Wagner

Agradecemos a sugestão e vamos estudar um futuro artigo sobre o assunto. Por enquanto, deixamos uma dica: uma das possíveis soluções para o problema seria combinar um servidor de Fax convencional, como o HylaFAX (www.hylafax.org) com uma central telefônica VoIP como o Asterisk, que foi tema da matéria de capa de nossa terceira edição. Alguém se habilita a colocá-la em prática?

Falha nossa!

Cadê os jogos?

Vários leitores nos escreveram, confusos com a aparente falta em nosso CD dos jogos mencionados na matéria "Os pingüins se divertem". Os jogos estão lá, mas infelizmente deixamos escapar o "detalhe" de como acessá-los: basta reiniciar o computador com o CD no drive. O disco é um Live-CD, com um ambiente gráfico similar ao Windows em que você poderá escolher entre os vários jogos inclusos

ESCREVA PRA GENTE

Se você tem dúvidas sobre o mundo Linux, críticas ou sugestões que possam ajudar a melhorar nossa revista, escreva para *cartas@linuxmagazine.com.br*. Devido ao volume de correspondência, é impossível responder a todas as mensagens, mas garantimos que elas são lidas e analisadas. As mais interessantes são publicadas nesta seção. Para dúvidas ou críticas referentes à sua assinatura da Linux Magazine, use o endereço: *assinaturas@linuxmagazine.com.br*.

Figura 3: Existem ferramentas de Software Livre para criar panoramas como este.

Dicas de [In]segurança

Kernel

O kernel do Linux desempenha as funções básicas do sistema operacional.

Foi descoberta uma falha provocada por falta de serialização na função *unix_dgram_recvmsg*, afetando kernels anteriores à versão 2.4.28. Um usuário local poderia potencialmente fazer uso de uma condição de disputa (race condition) para conseguir mais privilégios. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-1068.

Paul Starzetz do iSEC descobriu inúmeras falhas no carregador de binários ELF afetando kernels anteriores à versão 2.4.28. Um usuário local poderia usar essas falhas para conseguir acesso a binários que tenham apenas permissão de execução ou possivelmente ganhar mais privilégios. (CAN-2004-1070, CAN-2004-1071, CAN-2004-1072, CAN-2004-1073)

Foi descoberta uma falha na definição de limites para TSS que afeta os kernels das arquiteturas AMD, AMD64 e Intel EM64T anteriores à versão 2.4.23. Um usuário local poderia usar essa falha para causar uma negação de serviço (travamento do programa) ou possivelmente ganhar mais privilégios. (CAN-2004-0812)

Foi descoberta uma falha de estouro de inteiros na função *ubsec_keysetup* do driver Broadcom 5820 cryptonet. Em sistemas usando esse driver, um usuário local poderia causar uma negação de serviço (travamento do programa) ou possivelmente ganhar privilégios de administrador do sistema. (CAN-2004-0619)

Stefan Esser descobriu inúmeras falhas incluindo estouros de pilha no driver smbfs, afetando kernels anteriores à versão 2.4.28. Um usuário local poderia causar uma negação de serviço (travamento do programa) ou possivelmente ganhar mais privilégios. Para poder explorar essas falhas o usuário precisa ter controle sobre um servidor Samba conectado à rede. (CAN-2004-0883, CAN-2004-0949)

A SGI descobriu uma falha no carregador ELF que afeta kernels anteriores à versão 2.4.25. A falha pode ser explorada por um binário (executável) mal formado. Em arquiteturas diferentes da x86, um usuário local poderia criar um binário malévolo que causaria uma negação de serviço (travamento do programa). (CAN-2004-0136)

A Conectiva descobriu diversas falhas em certos drivers USB afetando kernels anteriores à versão 2.4.27. Os drivers vulneráveis usam a função *copy_to_user* em estruturas não inicializadas. Essas falhas poderiam permitir que usuários locais tivessem acesso a pequenas porções da memória usada pelo kernel. (CAN-2004-0685)

Referência no Red Hat: RHSA-2004:549-10
Referência no SuSE: SUSE-SA:2004:042

Cyrus-imapd

O servidor Cyrus IMAP é um eficiente servidor de emails IMAP, altamente escalável. Múltiplas vulnerabilidades

Postura das principais distribuições Linux quanto à segurança

Distribuição	Referência de Segurança	Comentários
Conectiva	Info: *http://distro2.conectiva.com.br/* Lista: *seguranca-admin@distro.conectiva.com.br* e *http://distro2.conectiva.com.br/lista/* Referência: CLSA-... [1]	Possui uma página específica; não há link para ela na página principal. Os alertas são sobre segurança, mas distribuídos através de emails assinados com a chave PGP da empresa para assegurar sua autenticidade. Contém também links para os pacotes atualizados e para fontes de referência sobre o problema sendo corrigido.
Debian	Info: *http://www.debian.org/security/* Lista: *http://lists.debian.org/debian-security-announce/* Referência: DSA-... [1]	Alertas de segurança recentes são colocados na homepage e distribuídos como arquivos HTML com links para os patches. O anúncio também contém uma referência à lista de discussão.
Gentoo	Info: *http://www.gentoo.org/security/en/gsla/index.html* Fórum: *http://forums.gentoo.org/* Lista: *http://www.gentoo.org/main/en/lists.xml* Referência: GLSA: ... [1]	Os alertas de segurança são listados no site de segurança da distribuição, com link na homepage. São distribuídos como páginas HTML e mostram os comandos necessários para baixar versões corrigidas dos softwares afetados.
Mandrake	Info: *http://www.mandrakesecure.net* Lista: *http://www.mandrakesecure.net/en/mlist.php* Referência: MDKSA-... [1]	A MandrakeSoft tem seu próprio site sobre segurança. Entre outras coisas, inclui alertas e referência a listas de discussão. Os alertas são arquivos HTML, mas não há links para os patches.
Red Hat	Info: *http://www.redhat.com/errata/* Lista: *http://www.redhat.com/mailing-lists/* Referência: RHSA-... [1]	A Red Hat classifica os alertas de segurança como "Erratas". Problemas com cada versão do Red Hat Linux são agrupados. Os alertas são distribuídos na forma de páginas HTML com links para os patches.
Slackware	Info: *http://www.slackware.com/ security/* Lista: *http://www.slackware.com/lists/* (slackware-security) Referência: [slackware-security] ... [1]	A página principal contém links para os arquivos da lista de discussão sobre segurança. Nenhuma informação adicional sobre segurança no Slackware está disponível.
SuSE	Info: *http://www.suse.de/uk/private/support/security/* Lista: *http://www.suse.de/uk/private/download/updates/* Referência: suse-security-announce Referência: SUSE-SA ... [1]	Após mudanças no site, não há mais um link para a página sobre segurança, que contém informações sobre a lista de discussão e os alertas. Patches de segurança para cada versão do SuSE Linux são mostrados em vermelho na página de atualizações. Uma curta descrição da vulnerabilidade corrigida pelo patch é fornecida.

[1] Todas as distribuições indicam, no assunto da mensagem, que o tema é segurança.

foram encontradas nos interpretadores de argumentos dos comandos *partial* e *fetch* do Cyrus IMAP Server (CAN-2004-1012, CAN-2004-1013). Há ainda estouros de buffer no código do 'imap magic plus' que também podem ser explorados. (CAN-2004-1011, CAN-2004-1015).

Um invasor poderia explorar essas vulnerabilidades para executar código arbitrário com os direitos do usuário no qual o Cyrus IMAP Server roda. Não há solução conhecida (mesmo provisória) até o presente momento. Todos os usuários do Cyrus-IMAP Server devem atualizar o programa para a versão mais nova. ■

Referência no Debian: DSA-597-1 cyrus-imapd
Referência no Gentoo: GLSA 200411-34 / cyrus-imapd
Referência no SuSE: SUSE-SA:2004:043

■ Sun Java Plugin

A segurança do plug-in Java fornecido pela Sun e Blackdown.org pode ser contornada, permitindo acesso a pacotes arbitrários. Isso permitiria que applets Java não autorizados executassem ações sem restrição no sistema. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CVE CAN-2004-1029.

A Sun e a Blackdown oferecem, cada uma, sua implementação de kits de desenvolvimento em java (Java Development Kits – JDK) e máquinas virtuais para execução de programas compilados em Java (Java Runtime Environments – JRE). Todas essas implementações possuem um plug-in que pode ser usado para executar applets Java em ambientes restritos – especialmente navegadores. Todos os plug-ins Java estão sujeitos a uma vulnerabilidade que permite acesso irrestrito a qualquer outro pacote Java.

Um invasor remoto poderia embutir um applet Java malicioso e convencer a vítima a executá-lo. O applet pode, então, contornar as restrições de segurança e executar qualquer comando ou acessar qualquer arquivo ou diretório legível pelo usuário usando o navegador.

Como paliativo, desative a execução de applets Java em seu navegador. Todos os usuários do Sun JDK devem atualizar o programa para a versão mais nova. ■

Referência no Gentoo: GLSA 200411-38 / Java

■ Samba

O Samba é uma implementação livre e gratuita de servidores de arquivos baseados nos protocolos SMB/CIFS, permitindo interoperabilidade com serviços de impressão e compartilhamento de arquivos de outros clientes SMB/CIFS – por exemplo, sistemas operacionais Windows®. Foram identificadas diversas vulnerabilidades no Samba.

Há um problema no daemon de compartilhamento de arquivos do Samba, permitindo que um usuário remoto consuma uma quantidade impressionante de recursos e potencialmente possa derrubar o serviço se usar nomes de arquivo com coringas ("*" e "?").

Para que o ataque seja um sucesso, basta que o daemon do Samba esteja rodando e um usuário remoto tenha acesso a um compartilhamento qualquer (mesmo que apenas de leitura). O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-0930.

Stefan Esser encontrou um problema na manipulação de cadeias de caracteres Unicode do gerenciador de arquivos do Samba que leva a um estouro de buffer baseado no segmento de dados (heap) que pode ser provocado remotamente, permitindo que atacantes remotos injetem código no processo smbd. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-0882.

Todos os usuários do Samba devem atualizar o programa para a versão indicada, que contém patches de segurança trazidos de versões mais recentes que não são vulneráveis a esses incidentes. ■

Referência no Gentoo: GLSA 200411-21 / samba
Referência no Mandrake: MDKSA-2004:136
Referência no Red Hat: RHSA-2004:632-17
Referência no SuSE: SUSE-SA:2004:040

■ Apache

O Apache é um dos servidores web (HTTP) mais populares da Internet.

Chintan Trivedi descobriu uma vulnerabilidade no Apache httpd 2.0 causada por uma limitação no comprimento de campos de dados. Essa imposição está presente no código que interpreta o cabeçalho HTTP. Ao enviar uma grande quantidade de requisições *HTTP GET*, um atacante remoto pode causar uma negação de serviço no sistema alvo. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-0942.

"Crazy Einstein" descobriu uma vulnerabilidade no módulo *mod_include* que pode causar um estouro de buffer e levar à execução de código arbitrário. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-0940.

Larry Cashdollar descobriu um estouro de pilha em potencial no utilitário *htpasswd* que pode ser explorado quando dados do usuário são passados a ele através de um CGI (ou PHP ou Perl...). Todos os usuários do Apache devem atualizá-lo para a versão mais nova. ■

Referência no Debian: DSA-594-1 apache
Referência no Gentoo: GLSA 200411-18 / apache
Referência no Mandrake: MDKSA-2004:134
Referência no Red Hat: RHSA-2004:562-11

sudo

O *sudo* é um programa que concede poderes limitados de superusuário a usuários específicos. Liam Helmer descobriu, entretanto, que o *sudo* não "limpa" o ambiente como deveria. Funções do shell Bash e a variável de ambiente *CDPATH* ainda são repassados para o programa rodando com permissões privilegiadas, permitindo que rotinas do sistema sejam sobrecarregadas. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-1051.

Essas vulnerabilidades só podem ser exploradas por usuários que estão cadastrados no arquivo sudoers e, portanto, possuem privilégios limitados de superusuário. Recomendamos que o pacote sudo seja atualizado o mais rápido possível. ■

Referência no Debian: DSA-596-2 sudo
Referência no Mandrake: MDKSA-2004:133

Ruby

Ruby é uma linguagem interpretada para scripts que permite desenvolvimento rápido e orientado a objetos. O módulo de CGI do Ruby pode ser usado para construir sistemas web.

Os desenvolvedores do Ruby encontraram um problema no módulo de CGI que pode ser disparado remotamente e que causa um laço de execução (loop) infinito. Um atacante remoto poderia disparar a vulnerabilidade através de um sistema web em Ruby que esteja vulnerável e causar o uso desnecessário de tempo de CPU por parte do servidor. Um certo número desses loops rodando simultaneamente pode provocar uma negação de serviço por esgotamento de recursos. O projeto "Common Vulnerabilities and Exposures" deu a essa falha o código CAN-2004-0983.

Não há solução conhecida até o presente momento. Todos os usuários do Ruby até a versão 1.6.x devem atualizar o programa para a versão mais nova. ■

Referência no Mandrake: MDKSA-2004:128

Openssl e Groff

O OpenSSL é um subsistema que implementa os protocolos SSL (Secure Socket Layer) e TLS (Transport Layer Security), bem como uma biblioteca de criptografia para uso geral. Inclui também o script *der_chop* para converter certificados codificados no padrão DER para o formato PEM. O Groff (GNU Troff) é um pacote de formatação de textos que lê arquivos de texto contendo comandos de formatação e produz um documento formatado. No pacote há o comando *groffer*, que mostra arquivos groff e páginas de manual (man pages) no X e no console (tty).

Tanto o *groffer* quanto o *der_chop* criam arquivos temporários nos diretórios em que as permissões sejam as mais universais possíveis (como o /tmp, por exemplo). Esses arquivos possuem nomes fáceis de prever. Um usuário local malicioso poderia criar links simbólicos no diretório temporário que apontem para um arquivo válido em outra parte do sistema de arquivos. Como resultado, rodar os scripts *groffer* e *der_chop* pode sobrescrever os arquivos para os quais os links apontam. Como é possível (e bem provável) que o próprio root execute esses scripts, nenhum arquivo do sistema está a salvo. Potencialmente, é possível comprometer de forma severa toda a segurança do sistema. Para mais informações sobre esse problema, consulte os documentos CVE CAN-2004-0969 e CAN-2004-0975 do Mitre

Não há solução conhecida (mesmo provisória) até o presente momento. ■

Referência no Debian: DSA-603-1 openssl
Referência no Gentoo: GLSA 200411-15 / OpenSSL
Referência no Mandrake: MDKSA-2004:147

BNC

O BNC é um proxy que "ricocheteia" sessões IRC. Leon Juranic descobriu que o BNC nem sempre protege os buffers, evitando que sejam estourados. Um servidor IRC malicioso poderia explorar a falha e fazer o limitado buffer transbordar, permitindo a execução de código arbitrário. O projeto "Common Vulnerabilities and Exposures" (cve.mitre.org) deu a essa falha o código CAN-2004-1052. ■

Referência no Debian: DSA-595-1 bnc
Referência no Gentoo: GLSA 200411-24 / BNC

ez-ipupdate

O ez-ipupdate é um utilitário para a atualização de informações de nome de hosts para serviços de DNS. Ulf Harnhammar, do Debian Security Audit Project, descobriu uma vulnerabilidade de formatação em cadeias de caracteres no ez-ipupdate. Um invasor poderia explorar a falha para executar código arbitrário com as permissões do usuário rodando o ez-ipupdate, que poderia ser o próprio root. O projeto "Common Vulnerabilities and Exposures" deu a essa falha o código CAN-2004-0980.

Não há solução conhecida até o presente momento. Mesmo assim, todos os usuários do ez-ipupdate são aconselhados atualizar o programa para a versão mais nova. ■

Referência no Debian: DSA-592-1 ez-ipupdate
Referência no Gentoo: GLSA 200411-20 / ez-ipupdate

Sergio Ianni: www.sxc.hu

Notícias do Kernel

O BitKeeper será substituído?

O desenvolvimento do Kernel ainda é mantido e controlado, basicamente, com a ajuda do software proprietário BitKeeper, algo que não deve mudar tão cedo. Mas as alternativas livres estão ficando cada vez melhores. O programa *tla*, também chamado de *arch*, provavelmente possui um projeto mais bem elaborado que o do BitKeeper e recursos bem mais poderosos, mas ainda não é capaz de trabalhar na escala de produção em que o kernel do Linux é desenvolvido, cuja velocidade e quantidade são de tirar o fôlego.

Andrea Arcangeli tem sido o mais fervoroso defensor do *tla* entre os desenvolvedores do kernel, mas Linus Torvalds ultimamente tem berrado cobras e lagartos sempre que alternativas ao BitKeeper são mencionadas na lista. David Roundy, autor original do sistema *darcs* de controle de versão, evitou um derramamento de sangue ao anunciar um repositório dos fontes do Linux baseado em seu "filho". Isso é um marco para qualquer gerenciador de versões, pois o tamanho descomunal de todo o histórico de alterações do kernel, sem contar a velocidade em que novos patches são adicionados, torna a tarefa digna de Hércules. O fato do *darcs* conseguir gerenciar esse mastodonte já é um feito considerável. O problema da escalabilidade não foi resolvido, entretanto: David afirma que há alguns problemas de desempenho. Mas a existência no BitKeeper de uma "porta de conexão" (o chamado *gateway*) para o *darcs* semelhante às já existentes para CVS e Subversion permitem que David obtenha realimentação importante vinda dos desenvolvedores – e, com isso, possa melhorar o *darcs*.

Esses gateways pavimentarão o caminho pelo qual passará aquele que substituirá o BitKeeper no futuro. Só quando a vasta maioria de desenvolvedores do kernel passar a usar *tla*, *darcs* ou outra alternativa qualquer de desenvolvimento distribuído é que Linus vai, finalmente, jogar a toalha e bradar "ok, vocês venceram!", abandonando de vez o BitKeeper. É importante ter em mente que uma grande parte das idéias que vêm sendo introduzidas nas ferramentas livres de gestão de desenvolvimento foram motivadas e inspiradas na decisão de Linus de adotar um software proprietário para a tarefa. A teimosia do criador pode ser combustível para a ira de seus pares, mas também o é para o aperfeiçoamento das ferramentas usadas por eles.

Violações da GPL

Uma penca de aparentes violações à licença GPL, tentativas de encontrar brechas em sua aplicação e mal-entendidos em geral vêm pipocando por toda parte. É difícil determinar se é um fenômeno sazonal, uma tendência temporária ou a maneira como as coisas serão daqui para a frente. Entre outubro e novembro, cerca de 6 empresas foram acusadas de violação à GPL na lista de desenvolvimento do kernel, a famosa *lkml*. Algumas dessas acusações foram produto de simples mal-entendidos, como quando Adrian Bunk sugeriu que o driver *drivers/char/rocket.c* continha duas licenças conflitantes: a GPL e a licença não-livre da Control Corporation. Theodore Ts'o, num comentário sobre o autor do código, afirmou que era apenas um engano e rapidamente conseguiu um documento da Comtrol que permitia a remoção da licença conflitante.

Os casos mais graves envolvem empresas procurando por brechas no código e na própria licença. Uma delas, por exemplo, distribui binários em uma versão e código-fonte em outra para os drivers de seus produtos, alegando que as versões mais novas são idênticas às antigas. Mentira deslavada: as versões novas contêm uma certa quantidade de código proprietário. Ela também se recusa a liberar atualizações para qualquer um que exercite o saudável direito – assegurado na GPL – de redistribuir seu software. Não são exatamente violações à GPL mas, se a veracidade das denúncias for comprovada, certamente trata-se de uma prática bastante grosseira e vulgar, para não dizer imoral e desonesta. Embora seja raro uma empresa usar código GPL e ao mesmo tempo ignorar o fato, pelo menos uma empresa – segundo Marcus Metzler - está se negando a distribuir o código fonte de seu produto baseado em código GPL.

Revisitando a querela das versões

Linus Torvalds, além de rebatizar a versão 2.6.10-rc1 do kernel Linux – era Zonked Quokka, agora chama-se Woozy Numbat – está tentando aplacar os ânimos de todos os que vêm reclamando do sistema de numeração de versões nos últimos tempos. Para reduzir a doideira das variações do tipo x.y.z.w-pre-rc-final que vem provocando o não-funcionamento de scripts de usuários ao redor do mundo, Linus decidiu que a série 2.6 terá apenas lançamentos *-rc* entre as versões pontuais. Sem *-pre*, sem *-final*, sem caos.

Ou, pelo menos, até amanhã. O sistema de numeração pode até se estabilizar, mas o código do 2.6 continua mutante, com alterações ocorrendo a velocidades cada vez mais estupefacientes. A versão 2.6.9 teve 3549 patches, mais do que qualquer outra versão exceto a 2.6.0. O número de desenvolvedores também alcançou o impressionante patamar de 400 colaboradores para as últimas duas versões, tendo ficado numa média de 250 para as anteriores. Um "fork" para a versão 2.7 não desponta no horizonte, e mesmo a idéia de se dividir as versões entre *estável* e *em desenvolvimento* parece ter sido defenestrada. Para o futuro, tenho a impressão de que a estabilidade será encontrada apenas nos principados governados por desenvolvedores como Alan Cox – que não tem medo de fazer "forks" importantes em sua série *-ac* – e distribuições engajadas como Red Hat, Debian e Tinfoil Hat Linux. Na outra ponta da corda, novos recursos e características vão continuar sendo adicionados ao kernel principal a uma velocidade de quebrar pescoços e chacoalhar bochechas. Se a coisa ficar assim, uma mudança da série de 2.6 para 2.7 (ou de 2.7 para 2.8) estará relacionada apenas ao cumprimento de metas de projeto (os famosos *milestones*) e nada terá a ver com estabilidade. Para mais detalhes, consulte as notícias do kernel nos números anteriores da Linux Magazine – esse parece ser um assunto recorrente.

Entrevista com Jaison Patrocínio

HP quer manter liderança em servidores no Brasil

Figura 1: Jaison Patrocínio, gerente de marketing de servidores da HP.

A lista de grandes corporações que ajudam a mover o Linux não pode deixar de incluir a Hewlett-Packard. Para se ter uma idéia do volume de negócios com o ambiente de código aberto, as receitas com Linux ultrapassaram US$ 2,5 bilhões em todo o mundo em 2003 (o número de 2004 ainda está sendo fechado). POR ALEXANDRE BARBOSA

Boa parte desses resultados é oriunda das vendas de servidores equipados com Linux e serviços associados, impulsionadas pelas razões de praxe que interessam aos grandes clientes corporativos: o baixo custo e a estabilidade.

A empresa também é um gigante de vendas, contando com mais de 6 mil profissionais de suporte e serviços nos segmentos de pré e pós-venda para o mercado Linux em todo o mundo. No Brasil, a HP tem como grandes destaques os servidores da linha Integrity e conta com um leque variado, que inclui até mesmo terminais thin client como o HP Compaq t5515 – proposta de computação de baixo custo da empresa que começou a ser oferecido ao mercado brasileiro no final do ano passado. É claro que essa oferta nem chega aos pés do que outros mercados, como os EUA, têm à sua disposição – ali, é possível encontrar notebooks e até media centers, PCs especializados em conteúdo multimídia, tudo com Linux.

Para falar sobre as ofertas no Brasil e a estratégia de Linux para o nosso mercado, a Linux Magazine entrevistou Jaison Patrocínio, gerente de marketing de servidores da HP e um dos cérebros por trás do pingüim na gigante de tecnologia.

Linux Magazine » ***A HP tem uma estratégia de Linux definida para o Brasil?***

Jaison Patrocínio » Existe uma estratégia local de oferta de soluções em Linux, mas estas não são montadas em cima de adaptações das estratégias mundiais da HP. Acontece que nem todas as ofertas da empresa em outros países são aplicáveis ou interessantes ao nosso mercado, por conta do tamanho dele e mesmo das características do país. Mesmo assim, nossa estratégia nos deixa numa posição de liderança no mercado; nosso plano é aumentar a diferença que nos separa das demais concorrentes em 2005. Uma das ferramentas é a participação em eventos internacionais, como o Linux Road, que acontece entre 14 e 18 de fevereiro em Boston. Ali, a HP terá um dia inteiro só de palestras. Levaremos 10 clientes do Brasil e mais 60 clientes de outros países da América Latina para conhecer os mais recentes desenvolvimentos na plataforma. Ou seja, temos uma estratégia de levar parceiros e clientes aos eventos mais representativos, ajudando assim a desenvolver o mercado. Continuaremos investindo em nossa fábrica de software em Porto Alegre e mantendo a equipe de serviços, que é capaz de prestar suporte do primeiro ao último nível, seja em sistemas operacionais, aplicações, integração ou resolução de problemas.

LM » ***Quais são as distribuições suportadas por vocês? Há algum destaque nos negócios em Linux neste ano?***

JP » Damos suporte a duas distribuições no Brasil, que já vêm de fábrica em nossos servidores: Red Hat e Novell SuSE. Por que apenas essas e não outras distribuições? É que, como temos por política a obrigação de dar suporte ao cliente em todo o ciclo de uso da solução, não teríamos condições de arcar com os custos de oferecer esse nível de serviço para todas as distribuições; assim, nos concentramos nas que achamos mais importantes para atender à demanda dos clientes. Este ano a nossa grande aposta está na migração do ambiente Sun para Linux. Temos uma ferramenta, desenvolvida em nossos laboratórios, que possibilita uma migração ágil de ambientes Solaris para Linux em um único dia, o que será muito atraente para várias corporações. Essa solução já foi adotada em outros lugares do mundo. Achamos que a redução de despesas propiciada pelas plataformas abertas atrairá a atenção de clientes potenciais para o baixo TCO (custo de propriedade) de nossos servidores Proliant ou a linha Integrity. Outro destaque é que o Linux já provou, e confirmará neste ano, a sua capacidade de lidar com aplicações de missão crítica sem atritos, o que é extremamente importante para o mercado corporativo.

LM » *Qual é a dimensão do crescimento dos negócios da HP com Linux no país?*
JP » Temos obtido um crescimento muito significativo no Brasil com o Linux e isso está nos levando a uma posição de liderança, com grande destaque. Para dar uma idéia desse crescimento, a HP foi, no terceiro trimestre do ano passado, a líder em vendas de sistemas em servidores, com uma fatia de mercado de 38,5% em faturamento (de acordo com dados do IDC). Foi um crescimento expressivo, mesmo em comparação com os 32% do segundo trimestre de 2004.

LM » *Como isso afeta a política de canais? Com que tipo de empresa a HP firma parcerias em Linux no Brasil?*
JP » Os nossos canais de distribuição e revenda estão todos preparados para vender, suportar e desenvolver soluções em Linux. O foco é a resolução dos problemas apresentados pelo cliente e, nesse sentido, todo o nosso canal está pronto e certificado em nossas soluções para trabalhar com Linux. Além de nossos revendedores e integradores, temos grandes parcerias com outros provedores de soluções, cujos produtos complementam a oferta da HP em Linux, como a SAP (para sistemas de gestão empresarial) ou a Oracle (para bancos de dados). Os parceiros são muito importantes, pois são eles que trazem mais valor ao Linux como alternativa corporativa. Não temos parcerias com empresas menores, se é essa a pergunta, mas estamos sempre abertos para fechar novas parcerias que agreguem valor às ofertas para clientes corporativos. Existe um formulário na página da HP Brasil para que qualquer empresa proponha sua oferta de solução, que será analisada pela HP (O endereço *http://www.hp.com.br/go/jornada/progisv.htm* tem um foco em soluções para handhelds, mas é genérico e pode ser usado por fornecedores de soluções para Linux em geral).

LM » *Como é o relacionamento com clientes governamentais? Há diferenças entre as esferas federal, estadual e municipal?*
JP » O mercado governamental é um dos mais maduros. Há diferenças de escala, é evidente, mas no geral toda a área de governo já tem uma visão consistente sobre as possibilidades da plataforma. Para mostrar as diferenças de escala, temos um grande cliente, cujo nome ainda não podemos revelar, que possui um cluster de mil servidores rodando com Linux e fazendo processamento de alta performance. E também temos, como clientes, pequenas prefeituras que possuem um servidor com dois processadores rodando Linux. Ou ainda casos como a cidade de Maringá, que está investindo fortemente em soluções baseadas em código aberto.

LM » *Como a estratégia da empresa afeta o treinamento e a certificação?*
JP » Sentimos que nossa estratégia já afetou esse mercado no passado. Houve um tempo em que precisávamos treinar nossos canais especificamente para as nossas soluções. Naquela época, o mercado de treinamento acompanhava nossos passos. Hoje o mercado está mais desenvolvido e já existem empresas especializadas e capacitadas para dar treinamento em todos os níveis. É claro que as certificações continuam sendo muito importantes para a HP, pois servem para qualificar os parceiros e profissionais que vão lidar com nossos produtos de maneira profissional e eficiente, oferecendo serviços e atendimento de qualidade aos clientes finais.

LM » *Qual é o perfil do profissional de Linux que as empresas buscam?*
JP » Com a grande penetração do Linux no meio universitário, existe muita gente boa e com um perfil bem diversificado chegando ao mercado. Há pessoas com conhecimento técnico profundo e que sabem compor soluções – esses profissionais compreendem as necessidades dos clientes e das empresas, até porque esse tipo de postura é algo que já vem sendo trabalhado dentro das universidades com o intuito de atender bem às demandas corporativas. É claro que existem posições que valorizam o perfil mais técnico, mas o que mercado procura é o tipo de profissional que tenha uma visão do todo.

LM » *O fato de propor o Linux a partir da perspectiva de uma grande empresa facilita o acesso aos grandes clientes?*
JP » Não necessariamente. Não adianta só a HP falar de Linux. Sem parceiros como uma Oracle ou uma SAP, apenas para citar algumas, não teríamos as ferramentas para compor uma solução de valor para o cliente. Não é o porte da empresa que abre as portas dos grandes clientes, mas sim a oferta de um conjunto consistente de soluções que permitam mostrar ao cliente uma solução estável, que traz um valor real para as suas operações.

LM » *Em que segmentos de atividade o Linux tem sido adotado de forma mais rápida?*
JP » Dois segmentos do mercado têm mostrado maior destaque: o setor governamental, pela postura e investimento feitos em Software Livre, e o setor de manufatura (que, para a HP, envolve serviços, indústria e o médio e grande varejo). Um exemplo é o Expresso Araçatuba, que usa sistemas Itanium com Linux. Também há uma penetração menor no setor de finanças, que é mais conservador quanto à adoção do Linux.

LM » *Lá fora a HP já vende media centers e notebooks baseados em Linux. É o tamanho do mercado brasileiro que faz com que não vejamos essas novidades por aqui?*
JP » O que determina a oferta local de produtos é a demanda. As soluções dentro da HP são sempre disponibilizadas mundialmente. O que acontece com freqüência é não existir demanda para um determinado tipo de produto ou solução, até por falta de interesse do mercado local. A maturidade do mercado dos EUA lhe dá certa vantagem; as coisas acontecem lá mais rápido do que na América Latina. Claro que isso não quer dizer que será assim para sempre: estamos preparados para oferecer mais produtos conforme a necessidade dos clientes. ■

Mundo Livre em Revista

Aberto o código do Distributed Computing Environment (DCE)

O código fonte do Distributed Computing Environment (DCE) do Open Group foi aberto sob uma licença livre. O DCE é um ambiente para aplicativos distribuídos rodando em diversas plataformas. Ele é utilizado especialmente em ambientes de hardware e software heterogêneos, compostos de diferentes sistemas operacionais, para administrar os recursos de processamento disponibilizados para os aplicativos, bem como para controlar o acesso a esses recursos. O sistema dispõe ainda de serviços de segurança para proteger e controlar o acesso a dados, serviços de atribuição de nomes, que facilitam a procura por recursos distribuídos e um modelo de organização de contas de usuários, serviços, aplicativos e dados que estejam largamente espalhados pelas máquinas da rede.

THE Open GROUP

Com a liberação do código fonte do DCE, o Open Group almeja a disseminação de conceitos e componentes dessa tecnologia, de modo que cada vez mais aplicativos façam uso dela. A organização, que detém inclusive os direitos da marca UNIX, ressalta ainda que o DCE torna disponível uma infraestrutura multiplataforma distribuída em múltiplos sistemas que é independente de fabricante. O código fonte do sistema será distribuído sob a GNU Lesser General Public License (LGPL). A versão 1.2.2 já se encontra disponível para download no site do Open Group. Devido às restrições norte-americanas de exportação de software, o código da versão do programa que contém o módulo de criptografia DES não está disponível para cidadãos de Cuba, Irã, Iraque, Líbia, Coréia do Norte, Sudão e Síria, para os quais uma versão especial sem suporte à criptografia foi liberada para download. ■

http://www.opengroup.org/dce/
http://www.opengroup.org/
http://www.fsf.org/licenses/licenses.html#LGPL
http://www.opengroup.org/dce/download/dl-pg2.htm

Polícia francesa (Gendarmerie) migra para OpenOffice.org

A polícia francesa (Gendarmerie) planeja a migração do Microsoft Office para o OpenOffice.org, conforme noticiou o site francês Toolinux, especializado em Software Livre. Ainda em janeiro, aproximadamente 35.000 desktops devem ser instalados com o pacote de aplicativos para escritório de código aberto e até o início do verão europeu esse número deve subir para 80.000. A Gendarmerie prevê uma economia de recursos da ordem de mais de 2 milhões de euros com a migração. ■

http://www.toolinux.com/news/logiciels/le_gendarme_et_openoffice_ar5768.html

Lançado o Xfce 4.2.0

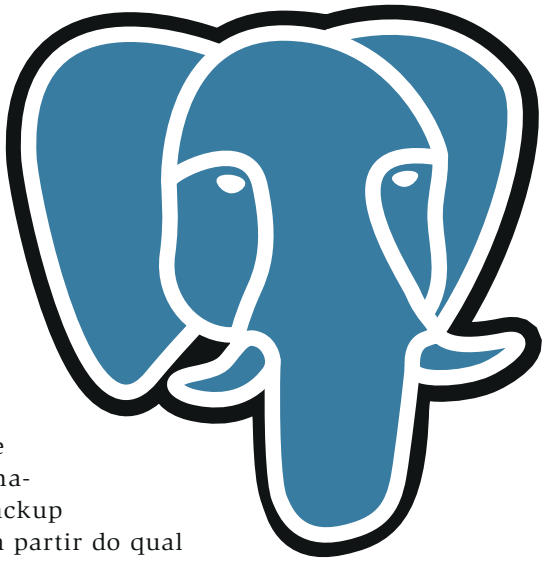
A versão 4.2.0 do ambiente desktop leve para Unix e Linux alcançou os conhecidos desktops KDE e Gnome em termos de recursos, mas continua a exigir muito menos do hardware do computador. Além de um painel, através do qual é possível iniciar aplicativos, e de uma barra de tarefas, o Xfce contém também um gerenciador de arquivos com uma interface que torna confortável o acesso a arquivos compartilhados do Windows. Vários aplicativos desktop, de editor a navegador, no mesmo estilo do KDE e do Gnome, são incluídos no Xfce. Aplicativos Gtk+ e Gnome se integram totalmente ao ambiente desktop, mas programas do KDE também são suportados.

Entre as melhorias em relação à versão anterior (a 4.0) estão a compatibilidade com as especificações do projeto Freedesktop e a utilização de recursos do X11 como o Xinerama, além da integração com aplicativos do KDE e Gnome. Da mesma forma que os dois outros projetos, cujo conceito básico da interface tem como origem o Common Desktop Environment (CDE), o Xfce traz também um menu "Iniciar". Para incluir aplicativos nesse menu há um editor específico e o novo "Application Finder", que interpreta os arquivos *.desktop* do mesmo modo que o KDE e o Gnome. Entre outras novidades há um gerenciador de sessões, que restaura o ambiente desktop e todos os aplicativos em execução na próxima vez que o usuário iniciar sua sessão.

Tanto o código fonte da versão 4.2.0 do Xfce quanto binários para diversas distribuições estão disponíveis para download. No site os-cillation há também um instalador gráfico. Aqueles que desejarem experimentar o ambiente desktop sem riscos podem fazê-lo usando o Xfld, uma versão do Knoppix que tem o Xfce como ambiente desktop padrão. ■

http://www.xfce.org/
http://www.kde.org/
http://www.gnome.org/
http://www.freedesktop.org/
http://www.opengroup.org/cde/
http://www.xfce.org/index.php?lang=en
http://www.os-cillation.com/article.php?sid=42
http://www.xfld.org/Xfld/en/index.html

Lançada versão 8.0 do PostgreSQL



O time de desenvolvimento do PostgreSQL acaba de lançar a versão 8.0 do seu sistema gerenciador de banco de dados objeto-relacional de código aberto, que a partir dessa versão roda nativamente no Windows NT, 2000, XP e Server 2003: o uso do ambiente Cygwin não é mais necessário. Além disso, entre outras novidades o PostgreSQL oferece suporte a transações aninhadas e pode realizar um backup contínuo da base de dados, a partir do qual é possível retornar à situação de um ponto determinado no passado ("Point-in-Time Recovery"). Do ponto de vista do desempenho há um novo recurso interessante: o administrador pode determinar exatamente qual sistema de arquivos deve ser usado para tabelas, índices etc. O desempenho do programa de manutenção do sistema, o *Vacuum*, também está visivelmente melhor.

O PostgreSQL é distribuído sob a licença BSD.

http://www.postgresql.org/
http://www.cygwin.com/
http://www.postgresql.org/docs/whatsnew
http://www.postgresql.org/about/licence

Bluefish chega à versão 1.0

No final de Janeiro o editor HTML Bluefish chegou à sua versão 1.0. Entre as novidades estão um manual mais abrangente, melhor integração com os ambientes desktop Gnome e KDE, melhor desempenho, melhores padrões para destaque de código, melhor detecção da codificação de caracteres (charset) usada no documento e várias correções de pequenos bugs.

O programa tem, entre outros recursos, capacidade de edição remota de documentos, suporte completo a 16 idiomas, incluindo o Português Brasileiro, destaque de código para 15 linguagens de programação, capacidade de edição de mais de 500 arquivos simultâneamente, um visualizador para documentação de referência e muito mais.

O Bluefish pode ser baixado no site oficial, onde estão disponíveis pacotes com o software para as distribuições Fedora (Core 1 e 2) Mandrake, Slackware e Debian, além de seu código-fonte.

http://bluefish.openoffice.nl

A união faz a força

Os usuários do Linux tem mais uma forma de conseguir os CDs de suas distribuições favoritas. O site *Linux Tracker* concentra "torrents" com imagens ISO dos CDs de instalação das principais distribuições e de projetos como o *The Open CD*.

O download dos arquivos é feito através da rede BitTorrent, que tem como característica o fato de que quanto mais gente baixa um arquivo, maior a taxa de transferência, já que o "upload" dos dados baixados para outros membros da rede é feito de forma automática. Com isso, os servidores contendo o arquivo não são soterrados com tantos pedidos de conexão logo após o lançamento de uma nova versão de um produto.

Para ter acesso ao software disponível no site basta fazer um cadastro gratuito e instalar um cliente BitTorrent. Há várias escolhas, desde o cliente oficial, que roda na linha de comando, até alternativas gráficas como o Azureus.

http://www.linuxtracker.org
http://azureus.sf.net

Primeiras partes do código fonte do Solaris disponíveis

A Sun Microsystems começou a liberar as primeiras partes do código fonte do sistema operacional da empresa, o Solaris 10, sob uma licença de código aberto. A primeira porção de código já aberta é a que contém o Software Dynamic Tracing (DTrace) da nova versão do Solaris. O Solaris 10 já está disponível para download para as arquiteturas x86 e Sparc na forma de imagens de CD e DVD. Uma versão em DVD deverá estar à venda a partir do dia 7 de março nos fornecedores especializados.

O DTrace é um programa que analisa as características de execução de aplicativos e do ambiente em que eles rodam em tempo de execução, procurando por gargalos de desempenho no sistema ou na rede que possam ser minimizados. Esse recurso precisa de suporte no kernel do sistema operacional e funciona atualmente apenas no Solaris. A licença sob a qual o DTrace foi liberado é a Common Development and Distribution License (CDDL), desenvolvida dentro da própria Sun e reconhecida recentemente como licença Open Source pela Open Source Initiative (OSI).

Em paralelo com a abertura do código do DTrace, a Sun inaugurou o seu novo portal OpenSolaris.org, que deverá servir de referência para a comunidade Open Source no que concerne ao desenvolvimento do OpenSolaris. Nos próximos meses a Sun irá paulatinamente abrir outras porções do código fonte do Solaris 10, mas ainda precisa se certificar de que trechos de código desenvolvidos em parceria com outras empresas não sejam abertos por engano, o que poderia infringir acordos de desenvolvimento. Essa análise demandaria tempo adicional. ■

http://www.sun.com/
http://www.sun.com/software/solaris/
http://www.sun.com/bigadmin/content/dtrace/
http://www.sun.com/cddl/
http://www.opensource.org/

Sun abre patentes do OpenSolaris

A Sun Microsystems liberou 1.600 patentes do OpenSolaris sob a Common Development and Distribution License (CDDL), reconhecida recentemente como licença Open Source pela Open Source Initiative (OSI). A medida deve proteger desenvolvedores que utilizem o código fonte do Solaris contra processos de infração de patentes. A informação no site da Sun não informa entretanto como a liberação das patentes irá influir na implementação de recursos do OpenSolaris em outros projetos de código aberto. É provável que isso seja um problema no caso de as licenças serem incompatíveis umas com as outras, como é provavelmente o caso da CDDL e da GNU General Public License (GPL), que diferem, por exemplo, nas permissões de uso do código fonte em projetos de código fechado.

Recentemente a IBM também anunciou a liberação de 500 patentes para a comunidade de desenvolvedores de sistemas de código aberto. Entretanto, ao contrário da Sun, a IBM não indicou uma licença específica, mas aceitou todas as licenças reconhecidas pela Open Source Initiative (OSI) na liberação de suas patentes. Por outro lado, a política de escolha das patentes do portfólio de 40.000 patentes da IBM também tem sido motivo de controvérsia.

Em um roadmap a Sun disponibilizou mais detalhes sobre o plano para a abertura do restante do código fonte do Solaris 10 e do desenvolvimento do OpenSolaris. No máximo até março um conselho administrativo com cinco membros deve ser fundado; ele terá a função de coordenar o desenvolvimento da versão de código aberto do sistema operacional da empresa. Dois dos membros do conselho virão da própria Sun, dois virão da comunidade OpenSolaris e o último deverá ser um membro da comunidade Open Source em geral, nomeado pela empresa. ■

http://www.sun.com/cddl/
http://www.opensource.org/
http://www.sun.com/smi/Press/sunflash/2005-01/sunflash.20050125.2.html
http://www.gnu.org/copyleft/gpl.html
http://br-linux.org/main/noticia-ibm_compartilha_500_patentes_c.html
http://opensource.org/licenses/index.php
http://kwiki.ffii.org/IbmEp0501En
http://www.opensolaris.org/roadmap/index.html

Insigne Free Software Brasil vai fornecer distribuição Linux para o projeto PC Conectado

A Insigne Free Software Brasil, empresa que desenvolve o Insigne Linux, é uma das empresas que deverá fornecer uma distribuição Linux para o PC Conectado, projeto do governo que pretende levar ao varejo 1 milhão de microcomputadores com conexão à Internet a preços populares.

Com uma base instalada de 100 mil computadores, ambiente gráfico amigável e customizado para o projeto do governo, que requer que, além do sistema operacional Linux, as máquinas cheguem ao mercado com mais 26 aplicativos de código aberto – como editor de texto, editor de planilhas eletrônicas e programa para montagem de apresentações – o sistema da Insigne é totalmente compatível com o Red Hat/Fedora Core, mas possui seu próprio sistema de gerenciamento de pacotes, o IPAK, capaz de lidar também com pacotes RPM e de usar o APT.

A solução da Insigne não trará o CrossOver Office, que permite rodar algumas aplicações do Windows no Linux, já que a idéia é oferecer ao público um computador cujo sistema seja de código totalmente aberto, como é o caso da distribuição Linux da empresa. O suporte técnico para os compradores do PC Conectado está orçado em R$ 30,00 ao ano, período no qual o usuário também irá desfrutar de atualizações regulares. Usuários que optarem pelo suporte técnico receberão uma chave de ativação.

O Insigne Linux se encontra atualmente na versão "Zino 3.0" e pode ser baixado diretamente do site da Insigne Free Software Brasil. O sistema vem customizado e pré-instalado (OEM) em microcomputadores comercializados pela Semp Toshiba, Magazine Luiza, Positivo Informática e Wal Mart, entre outros. ■

http://www.insignesoftware.com/
http://www.insignesoftware.com/produtos/gnu.php
http://www.iti.br/twiki/bin/view/Main/PressRelease2005Jan20A
http://www.insignesoftware.com/docs/guia74.html
http://www.codeweavers.com/products/cxoffice/
http://www.insignesoftware.com/downloads/insigne/001.php
http://www.insignesoftware.com/recursos/recursos.php

Diretor do MIT Media Lab planeja computador barato para países emergentes

Fundador e diretor do Media Lab da renomada universidade americana MIT (Massachusetts Institute for Technology) e co-fundador da Wired (que ainda deve estar na memória de todos os envolvidos com a Internet pré-bolha .COM), Nicholas Negroponte planeja desenvolver, em conjunto com a AMD, um computador portátil com produção em massa para ser usado em países em desenvolvimento. Ele estima o preço do equipamento, que terá uma tela de 14 polegadas e rodará Linux sobre processadores AMD, em 100 dólares.

"O modelo já foi desenvolvido; agora precisamos conversar com parceiros adicionais. A chegada do equipamento ao mercado deverá acontecer entre 12 a 18 meses", informou Negroponte. Um porta-voz da AMD confirmou o trabalho de cooperação com o MIT e descreveu o plano como um "projeto altamente interessante". A própria AMD apresentou no ano passado um projeto semelhante: o Personal Internet Communicator (PIC), cujo desenvolvimento tinha por objetivo fornecer microcomputadores a preços acessíveis para países emergentes e em desenvolvimento – dentro da sua estratégia 50x15 a AMD almeja equipar 50% da população mundial com computadores pessoais. O AMD PIC – software inclusive – deverá custar, entretanto, 185 dólares (com monitor, 250 dólares)

O projeto de Negroponte conta com as promessas de apoio não somente do MIT e da AMD, mas também dos gigantes do mercado de TI mundial Google, Motorola, Samsung e News Corp – ou seja: tecnologia, hardware, conteúdo e difusão via satélite. A diferença para o projeto original da AMD (além do preço) é que, segundo os planos de Negroponte, o computador deverá revolucionar o ensino nos países onde for utilizado. Deve ser produzido em larga escala e vendido aos ministérios de educação dos países em desenvolvimento, que deverão distribuí-los convenientemente aos alunos das escolas. Uma empresa – na qual o MIT teria participação societária – deverá ser fundada para cuidar da produção e comercialização do computador.

Negroponte ainda informou que, com o intuito de possibilitar a compra do computador pelos governos dos países pobres, ele entrou em contato com o Banco Mundial. "Nós não vamos lançar o computador no mercado. O mercado potencial para o grupo alvo do produto é de cerca de 800 milhões de unidades", ressaltou. Em janeiro ele conversou com o ministro da educação chinês, que teria manifestado grande interesse pelo projeto. ■

http://www.media.mit.edu/
http://www.mit.edu/
http://web.media.mit.edu/~nicholas/
http://50x15.amd.com/

Lançada a versão 1.0 do Skype para Linux e Mac OS X

A versão 1.0 para Linux e Mac OS X do telefone via Internet (VoIP) e cliente de troca de mensagens instantâneas Skype já está disponível gratuitamente para download. O hardware mínimo para rodar o programa é um processador de 400 MHz e 128 MB de RAM.

O Skype é um sistema proprietário que permite realizar telefonemas gratuitos via Internet dentro de sua rede privada e ligações pagas para usuários da rede de telefonia fixa. O custo de ligações feitas com o Skype para telefones fixos depende da localidade: para aquelas onde há cobertura pela empresa, o custo é de 20 centavos de dólar por minuto, independente de onde se esteja ligando. Em lugares que não dispõem dessa cobertura – o Brasil é um deles – são cobradas taxas individuais cujos valores dependem, entre outras coisas, da proximidade de uma localidade com cobertura.

O programa funciona bem até via modem. Para ligações dentro de sua rede privada, a qualidade do áudio das ligações supera a da rede de telefonia fixa mesmo com uma conexão discada de baixa velocidade. O sistema Skype funciona segundo o modelo "peer to peer" e usa para telefonemas e mensagens instantâneas um sistema de codificação, que emprega um algoritmo AES com uma chave de 256 bits. A negociação das chaves simétricas ocorre via algoritmo RSA com uma chave de 1024 bits. ■

http://www.skype.com/

Novell vai portar o Evolution para o Windows

A Novell está planejando portar o Evolution, gerenciador de informações pessoais e cliente de email de código aberto da empresa, para o Windows®. A fonte da informação é nada menos que o blog de Nat Friedman, co-fundador da Ximian, empresa especialista em Linux desktop comprada pela Novell em 2004, e atual vice-presidente de pesquisa e desenvolvimento da companhia.

Friedman informa, entretanto, que o trabalho de criação de uma versão para Windows® do Evolution é enorme e que ainda não há nenhum cronograma para implementação do software. O Evolution é considerado atualmente um dos maiores concorrentes do MS Outlook. ■

http://nat.org/2005/january/#17-January-2005
http://www.novell.com/
http://www.novell.com/products/desktop/features/evolution.html

Edulinux será adotado por escolas chilenas

O Chile quer dar vida nova a PCs antigos em escolas do país usando o Edulinux. O sistema transforma computadores conectados em rede em terminais leves, que podem executar programas instalados em um servidor central. Isso deve permitir, inclusive, a utilização de softwares em suas versões mais atualizadas. De acordo com o jornal El Mercurio, o objetivo do projeto é disponibilizar um PC para cada 30 alunos.

Primeiramente, 25 escolas vão testar a arquitetura cliente-servidor do sistema, que depois deverá ser utilizada em 600 escolas. O projeto piloto tem por meta mostrar que terminais leves podem executar programas que, de outro modo, não poderiam rodar diretamente no hardware disponível. Além disso, espera-se que o uso de Internet e email através de um servidor central seja mais simples. Ainda segundo o jornal, caso os alunos não consigam usar o Edulinux, poderão voltar a adotar uma solução proprietária. O custo total para uma solução baseada no Edulinux com 5 desktops é estimado em R$7.000,00. Para 20 desktops estima-se um custo em torno de R$25.000. ■

http://www.edulinux.cl/
http://www.elmercurio.cl/

Proteção ao alcance do usuário comum

Firewall para Leigos

Os Firewalls, dispositivos que controlam o tráfego e bloqueiam acessos indesejados, estão se tornando cada vez mais sofisticados. Entretanto as ferramentas de criação e configuração de firewalls estão ficando cada vez mais simples e acessíveis ao usuário não técnico.

POR JOE CASAD E ACHIM LEITNER

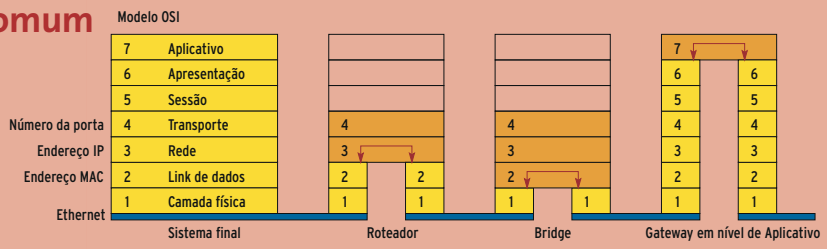


Figura 1: Os firewalls modernos podem operar como pontes ou *bridges* (à esquerda), roteadores (ao centro) ou *gateways* de aplicação (à direita).

Seu computador permite que você tenha uma visão do mundo, mas ninguém quer ter o mundo olhando seu computador. Os invasores estão ficando cada vez mais ousados e tecnicamente mais "afiados". Não é mais aceitável esperar que tais delinqüentes simplesmente não notem sua estação de trabalho "dando sopa". Se você está conectado à Internet, é melhor colocar-se atrás de algum tipo de firewall.

Os firewalls são oferecidos em vários tamanhos, formatos, preços e tecnologias. Um fenômeno curioso vem ocorrendo: o que antes era um simples filtro hoje é um conjunto completo de produtos de segurança. Tradicionalmente, um firewall é um tipo de roteador que fica na camada 3 do modelo de referência OSI. A camada 3 é, justamente, onde residem os protocolos de rede, mais precisamente o protocolo IP. O roteador lê os endereços IP e toma decisões quanto ao destino do datagrama– ou seja, em que direção deve roteá-lo. Um firewall também inspeciona a camada 4 (ou seja, os protocolos de transporte TCP e UDP) para identificar os serviços relevantes e interpretar os avisos presentes nas *flags*.

Produtos modernos de firewall podem operar em outras camadas da pilha de protocolos (figura 1). Esse enfoque multicamada pode ser estendido para baixo até a camada 2 – situação em que temos o famoso *bridgewall*. Da mesma forma como uma bridge (ou um switch) controla o tráfego baseado nos endereços MAC (camada 2), um bridgewall inspeciona os pacotes desde a camada 2 até a camada 4. O bridgewall é um filtro de pacotes tão flexível quanto um switch.

Um *gateway* de aplicação oferece um nível adicional de segurança em uma camada superior. Ele age na conexão TCP como um intermediário entre o cliente e o servidor – é o conhecido *proxy*. Isso permite que o firewall inspecione diretamente os protocolos da camada de aplicação e detecte pacotes ilegais que quebrem as regras estabelecidas pelo administrador de segurança.

Obviamente, os firewalls mais exóticos são produtos dedicados, baseados em hardware e destinados a redes gigantescas e configurações complexas. Estamos mais interessados no que se pode fazer com hardware comum, uma distribuição Linux e alguns programas fáceis de encontrar. Este mês, trazemos uma seleção de ferramentas para construção de firewalls, além de utilitários poderosos que simplificam sua configuração e manutenção. Com elas, você não precisa ser um expert em redes ou segurança para administrar um firewall.

Em um dos artigos, "Cão de Guarda", mostraremos como montar um firewall com IPtables ou IPchains usando o Guarddog. O artigo seguinte, "Ponte Levadiça", discorre sobre as ferramentas necessárias para se construir um firewall do tipo "bridge". Veremos como os "bridgewalls" funcionam e em que situações eles são interessantes.

Um maiores incômodos quando se administra um sistema de segurança é a quantidade de informação acumulada nos registros de eventos, os chamados *logs*. Nosso terceiro artigo mostra ferramentas para ler e analisar tais logs. Nossa matéria final, "Cumprindo Tabela", descreve o Shorewall, abreviação de *Shoreline Firewall* – outra ferramenta que não é um firewall em si, mas que abranda o inferno que é a vida do profissional de segurança.

Nenhum assunto é mais importante que a segurança; da mesma forma, poucos dispositivos são tão importantes na cadeia de segurança quanto os firewalls. O tema deste mês é apenas um lembrete de que firewalls não são apenas para experts: qualquer um conectado à Internet **precisa** de um firewall. Não é uma opção. ■

CAPA

Configure seu firewall com o Guarddog

Cão de guarda

O Guarddog, um programa do KDE, promete um "firewall fácil" no Linux com uns poucos cliques. Mais importante: auxilia usuários inexperientes a proteger seus computadores – ou mesmo redes inteiras – contra os perigos da Internet. POR HOLGER JUNGE

Os subsistemas ipchains (kernel Linux 2.2) e iptables (kernel Linux 2.4 e 2.6) configuram o kernel do Linux para agir como firewall. Entretanto, sua operação por linha de comando pode parecer um tanto "cifrada" para neófitos do Linux. Em meio a essa confusão, Simon Edwards desenvolveu o Guarddog [1] para tornar as coisas mais fáceis. O Guarddog é uma ferramenta gráfica de configuração e manutenção de firewalls. Protegida sob a licença GPL, roda tanto no KDE 2 como no KDE 3.

A versão estável mais atual (2.4.0) saiu em dezembro de 2004 e pode ser obtida em [2]. Como os nossos testes foram feitos antes dessa data, experimentamos com as versões 2.2.0 (estável) e 2.3.2 (desenvolvimento). Além dos códigos fonte, o site possui binários prontos para o Mandrake e o SuSE. Há binários para o Red Hat também, mas apenas para a versão 2.2.0. Há também a indicação de um repositório APT para o Debian Woody com a versão 2.2. Se você usa Debian Sarge ou Sid, os repositórios oficiais da distribuição possuem, respectivamente, as versões 2.3.2 e 2.4.0. Usuários do Sarge que preferirem instalar a versão instável em lugar da disponível para o Sid devem estar cientes dos prós e contras mencionados no quadro "Aos Corajosos".

O Guarddog foi projetado para o usuário doméstico, seja em uma única máquina ou em uma pequena rede local. Embora as distribuições mais importantes como Conectiva, Red Hat, Mandrake e SuSE possuam suas próprias ferramentas gráficas para montagem de firewalls, tais ferramentas padecem de uma certa dose de granularidade – em outras palavras, costumam ser "oito ou oitenta". Alguns usuários precisam de uma configuração simples de fazer, mas um pouco mais detalhada. O Guarddog pode ser a ferramenta ideal para isso.

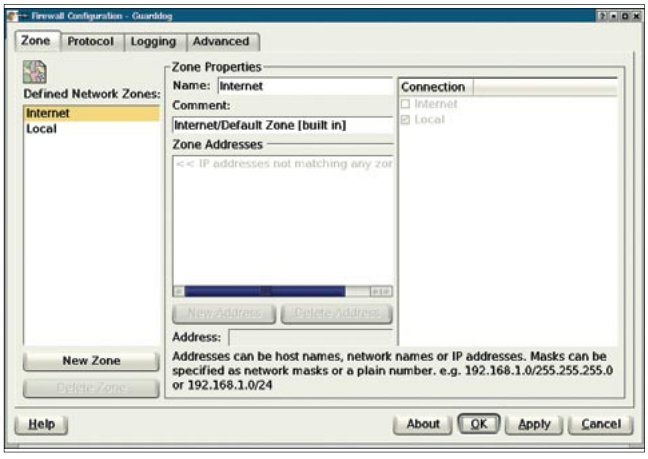


Figura 1: A interface gráfica do Guarddog depois de iniciado. Observem as duas zonas pré-configuradas, *Internet* e *Local*.

Figura 2: A aba *Protocol* permite informar ao firewall quais protocolos deve liberar e quais bloquear. O usuário não precisa se preocupar com números de porta e outros bichos.

Armadilhas de Segurança

Os usuários inexperientes precisam ter bastante cuidado ao montar um firewall. A facilidade com que a "empurração de mouse" habilita e desabilita opções pode estimular – erroneamente! – o usuário a deixar mais portas abertas do que o necessário. No outro extremo, é bem fácil criar uma muralha tão intransponível que deixaria alguns serviços importantes inacessíveis.

Além disso, o Guarddog é um programa do KDE e não deve ser rodado em uma máquina servidora, que muitas vezes roda apenas em modo texto. É preferível usar o Guarddog em uma máquina cliente para gerar a configuração do firewall e, depois, copiar o script resultante para seu servidor dedicado.

Como o Guarddog é baseado em ipchains ou iptables, os usuários precisam certificar-se de que os módulos apropriados do kernel estão disponíveis no servidor. Muitas distribuições encarregam-se automaticamente disso. Na remotíssima possibilidade de a sua não tomar essas providências, será necessário recompilar o kernel para incluir os módulos de ipchains ou iptables.

O Guarddog usa comandos de filtragem que levam em conta os protocolos dos pacotes em trânsito. os usuários não precisam se preocupar com os números de porta, o que evita erros de configuração. É possível ainda determinar grupos de máquinas – as chamadas zonas – o que permite, entre outras coisas, criar redes periféricas, as chamadas Zonas Desmilitarizadas (DMZ).

A interface

O Guarddog deve ser executado com privilégios de superusuário (root) para que o programa possa colocar imediatamente as novas regras de filtragem em ação. A figura 1 mostra o Guarddog no momento em que é chamado. Infelizmente, a GUI não é muito intuitiva em alguns pontos. O Guarddog possui quatro abas: a guia *Zone* permite que os usuários agrupem máquinas em zonas.

Em *Zone properties* (propriedades da zona) deve-se digitar os endereços IP (únicos ou em faixas) para a zona. Há duas já configuradas, uma delas chamada de *Internet* e a outra de *Local*. Nenhuma delas pode ser apagada. A zona *Internet* automaticamente inclui qualquer endereço IP que não faça parte de nenhuma outra zona. Já a zona *Local* compreende os endereços da rede interna. Uma máquina sozinha ficará contente apenas com essas duas zonas.

Pode-se usar a aba *Protocol* (ver figura 2) para permitir ou bloquear protocolos específicos. A estrutura em árvore à direita organiza os protocolos por categoria. Sem sombra de dúvida, o DNS é o primeiro serviço que você deverá permitir; ele está na categoria *Network*. Ao ativar a opção *DNS - Domain Name Server* aparecerá uma marca para indicar a liberação do serviço. Para aplicar as alterações, pressione o botão *Apply* (aplicar). Um segundo clique na opção mostra um X, indicando que o firewall irá, explicitamente, rejeitar qualquer conexão que use o protocolo. Se nenhuma marca estiver aparecendo, o firewall simplesmente bloqueará o tráfego baseado nesse protocolo.

Além do DNS, você pode precisar (ou, melhor dizendo, certamente precisará) de HTTP, HTTPS (HTTP seguro), FTP (presente na categoria *File transfer*) e os protocolos de email SMTP e POP3 (na categoria *Email*).

"Fichando" protocolos

A aba *Logging* (ver figura 3) permite que configuremos o Guarddog para registrar eventos no sistema *syslog*. Isso pode ser usado para, por exemplo, detectar "crackers" varrendo suas portas em busca de falhas. O Guarddog pode ajustar a taxa de verificação (*logging rate*) para limitar a bagunça criada por eventos do firewall nos logs do sistema. É importante impor um limite para isso; do contrário, seu computador pode ser derrubado por Negação de Serviço (DoS

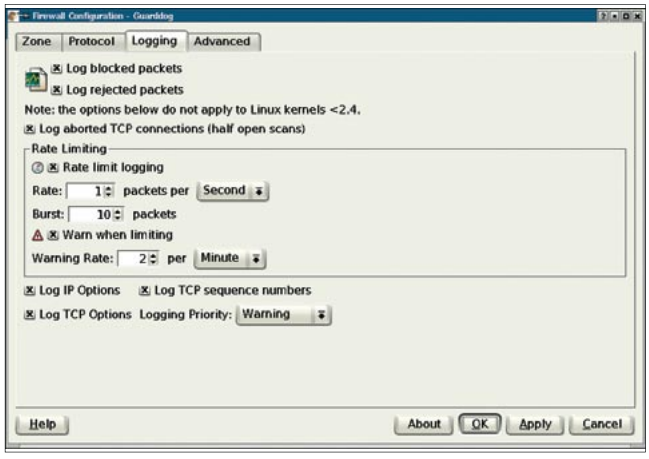


Figura 3: Os administradores podem usar a aba *Logging* para especificar o tipo de registro de eventos (log) que o firewall deve fazer.

– Denial of Service). Uma enxurrada de pacotes IP malformados poderia rapidamente abarrotar os arquivos do syslog e sobrecarregar seu disco rígido.

Se, por algum motivo qualquer, for preciso um nível maior de detalhe sobre os pacotes IP e TCP que entram, é possível ativar a opção de mostrar o fluxo de dados na parte de baixo da tela.

A aba *Advanced* (avançado – ver figura 4) oferece opções avançadas para ajuste fino do firewall. É de especial interesse para administradores experientes. Se algo der errado, não entre em pânico: clique em *Restore to factory defaults...* (restaurar valores iniciais) para usar os padrões do software. Os padrões para *Local Dynamic Port Range* também são suficientes na maioria dos casos. Elas especificam a faixa de portas que o Linux pode usar para iniciar conexões de dentro para fora da rede.

Se algum protocolo não estiver listado na aba *Protocol*, é possível clicar em *New Protocol* (novo protocolo) e digitar o nome, o transporte usado (TCP ou UDP) e as portas usadas por ele.

O Guarddog possui um útil recurso de importação e exportação dos scripts de firewall. É possível, por exemplo, exportar as regras já criadas para um script simples em shell e armazená-lo em */etc/rc.firewall*. Como não é comum servidores rodarem o ambiente KDE, os administradores podem, simplesmente, pressionar o botão *Export* para criar o script. Depois, basta copiá-lo para o servidor e rodá-lo.

Uma porta para o mundo

Nem sempre os firewalls Linux são usados para proteger apenas a máquina em questão. Pelo contrário, é comum usar sistemas Linux como parte da estrutura de segurança de redes inteiras. Nesse caso, o computador rodando Linux funciona como um "porteiro" – o chamado *gateway* – e possui duas interfaces (placas) de rede: uma voltada para a Internet, a outra conectada à rede interna. (ver figura 5). É bastante simples configurar o Guarddog para o papel. A única ressalva: isso só funciona com iptables, por isso é necessário kernel 2.4 ou 2.6. Será preciso configurar o mascaramento de IP *antes* de usar o Guarddog para gerar as regras do firewall. Embora o Guarddog não possa auxiliá-lo nesse passo, seu primo Guidedog pode [4].

O primeiro passo é criar uma nova zona no Guarddog para a rede local. Para isso, clique em *New Zone* na aba *Zone*. Qualquer nome serve – LAN, por exemplo. Depois clique em *New Address* para configurar os endereços IP, que podem ser únicos, faixas ou redes inteiras (como, por exemplo, *192.168.1.0/24*).

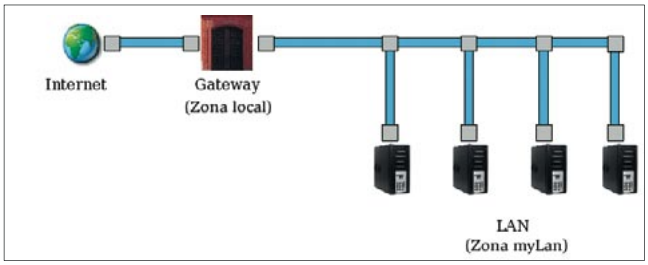


Figura 5: O computador rodando Linux e atuando como firewall também é um gateway para a rede interna acessar a Internet.

Agora clique em *Internet* e *Local* sob o ramo *Connection* para certificar-se de que a zona LAN está conectada à Internet e à máquina local. Na aba *Protocol* escolha *Internet* e habilite (ou bloqueie) os protocolos apropriados na coluna LAN. Finalmente, clique em *Apply* para armazenar as regras no script */etc/rc.firewall* e ativar o firewall. ■

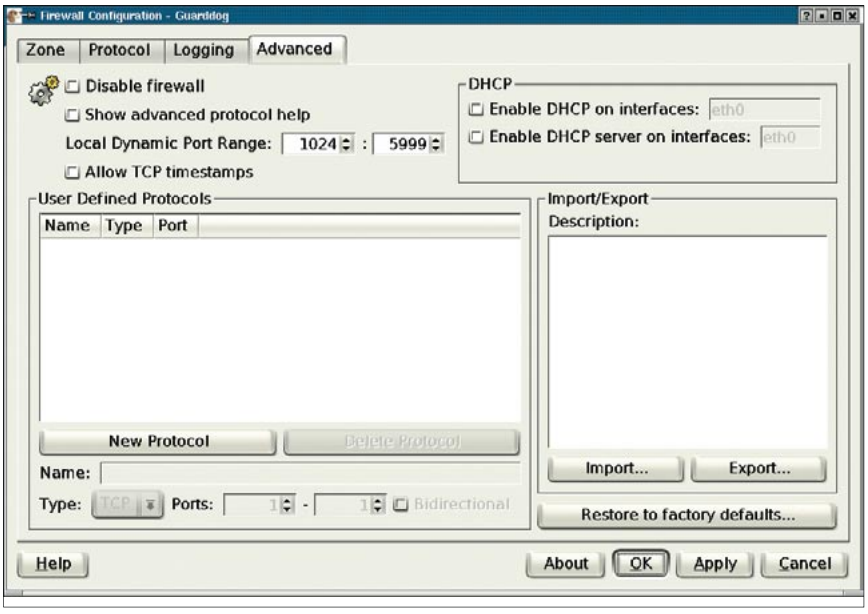


Figura 4: O Guarddog permite que muitos detalhes do firewall sejam configurados. Por exemplo, é possível definir novos protocolos e importar ou exportar scripts.

Aos Corajosos

Apesar da versão estável ser a 2.4.0, muitas distribuições (como a Debian Sarge) ainda disponibilizam apenas a versão instável anterior, a 2.3.2. Há alguns problemas em usar essa versão em ambientes de produção. Há, é claro, vantagens em relação à 2.2, como a definição, pelo usuário, de protocolos desconhecidos. Há também o suporte ao kernel 2.6, recurso ausente no Guarddog 2.2, e a adição de muitos protocolos novos como RSync, Distcc, GKRellm, Bittorrent, Servidor de Chaves PGP, Jabber sobre SSL e o Microsoft Media Server. Todas essas vantagens podem ser usufruídas também com o Guarddog 2.4.0, portanto desaconselhamos o uso da versão instável.

INFORMAÇÕES

[1] Página oficial do Guarddog: *http://www.simonzone.com/software/guarddog*

[2] Download do programa: *http://www.simonzone.com/software/guarddog/#download*

[3] Manual Online do Guarddog: *http://www.simonzone.com/software/guarddog/#manual*

[4] Guidedog: *http://www.simonzone.com/software/guidedog/*

SOBRE O AUTOR

Holger Junge trabalha para a Lifemedien (*www.lifemedien.de*) onde, como bom pastor, apascenta servidores de domínio e de web além de bancos de dados MySQL e Oracle. Obviamente, todos rodando Linux.

Mergulhe fundo nos logs do seu firewall

Abrindo a caixa preta

Firewalls baseados em Netfilter produzem registros de eventos (os famosos *logs*) tão monstruosos e gigantescos que ninguém em sã consciência quer (ou consegue) digeri-los manualmente. Para nos tirar desse lodaçal lançamos mão das ferramentas de análise de logs. Este mês testamos o IPtables Log Analyzer, o Wallfire wflogs e o FWlogwatch, que pretendem auxiliar os administradores no desenvolvimento e manutenção de suas regras de firewall. Será que dão conta do recado? POR RALF SPENNEBERG

Em ambientes protegidos por firewalls, o administrador precisa manter um controle rígido sobre como estão definidas as regras de filtragem e a quantas anda o tráfego dos clientes. Entretanto, megabytes de arquivos de registro (*logfiles*) podem afogar até os profissionais mais competentes, arrastando-os numa fenomenal enxurrada de informações que eles enfrentam sem proteção apenas para ter certeza de que não deixaram nenhuma pista para trás. É uma vida dura...

Os amigos do syslog

Ferramentas de análise de protocolo podem ser de grande ajuda para sair desse espinhoso dilema. Felizmente, os usuários de Linux possuem muitas opções de programas para análise de firewalls. Neste artigo analisaremos três alternativas: IPtables Log Analyzer [1], wflogs do projeto Wallfire [2] e fwlogwatch [3]. Todos os três reconhecem uma variedade bastante grande de protocolos e apresentam os resultados em relatórios HTML muito bonitos. O wflogs e o fwlogwatch possuem, além disso, modos de visualização em tempo real. O IPtables Log Analyzer é a única ferramenta que usa um banco de dados para armazenar os resultados.

Correndo em direção contrária, o IPtables Log Analyzer é calcado no subsistema Ulogd [4] de Harald Weltes, que substitui o subsistema padrão do kernel, o syslog. Infelizmente, ferramentas de análise gratuitas que reconheçam o banco de dados do Ulog são raríssimas. Uma delas é o Ulogd-php [5]. Ao contrário de todos os outros sistemas de registro de eventos, o Ulogd pode registrar incidentes em seu banco de dados.

IPtables Log Analyzer

O IPtables Log Analyzer é um servidor que gera logs do IPtables (kernels 2.4 e 2.6) artisticamente formatados em lindas páginas HTML (ver figura 1). A ferramenta tem três componentes. O gerenciador de saída (*database feeder*) armazena cada evento registrado nos logs em um banco de dados MySQL. Os administradores podem consultar os resultados em uma interface web. O gerenciador, o banco de dados e a interface web podem rodar tanto na mesma máquina como em máquinas separadas. Nesse segundo caso, o banco de dados pode recolher logs de múltiplos firewalls.

Depois de decidir-se sobre a arquitetura a ser implementada, o administrador precisa criar um banco de dados MySQL chamado *iptables*, liberando acesso para que os usuários *iptables_admin* e *iptables_user* possam manipulá-lo. Depois, é preciso gerar as tabelas dentro do banco de dados (veja a listagem 1). Obviamente, depois de tudo isso deve-se escrever as regras para o IPtables. Duas cadeias definidas pelo usuário nos parece ser a melhor abordagem (veja a listagem 2).

Criando cadeias

Em vez do tradicional *-j ACCEPT*, o IPtables usará, agora, *-j LOG_ACCEPT*. Essas modificações não são necessárias se usarmos o Shorewall [6] ou o SuSE Firewall em CD [7]. Isso posto, temos que informar também que a SuSE não vai mais oferecer suporte ao seu firewall comercial. Essa é mais uma razão para que os administradores reflitam e passem a usar *apenas* ferramentas e atualizações provenientes do universo do Software Livre. Ficar na mão de fornecedores comerciais é isso mesmo: ficar na mão...

O próximo passo é instalar a interface web. Para isso, o administrador precisa simplesmente copiar (ou mover) o diretório *web* para dentro da área de documentos HTML (chamada normalmente de *document root*) do servidor web que

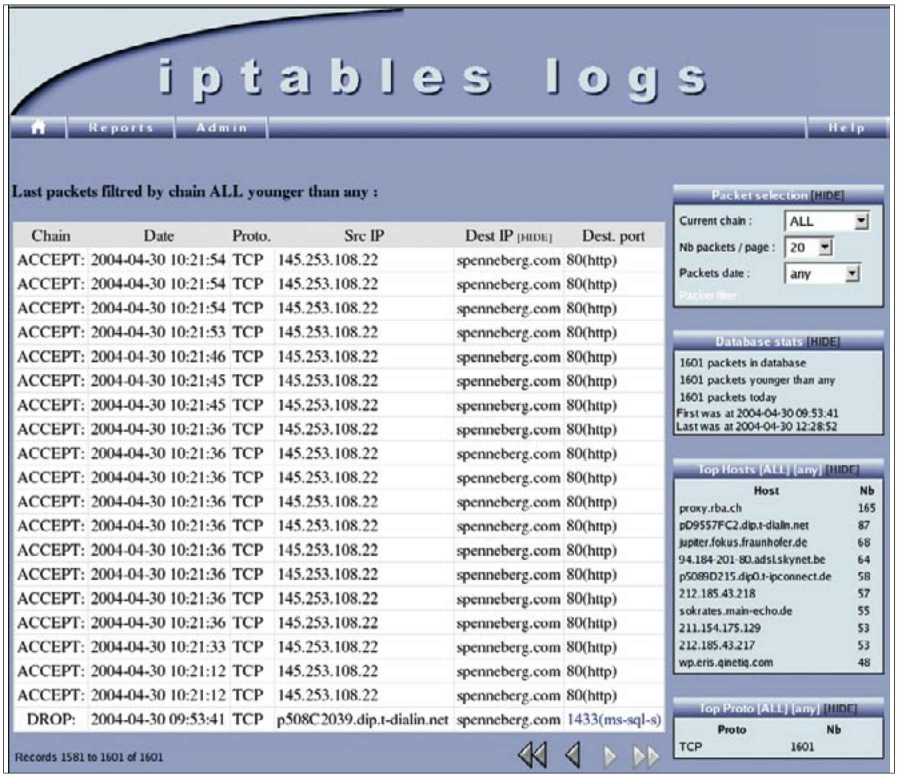


Figura 1: O IPtables Log Analyzer oferece uma visão geral bem clara sobre o estado dos firewalls.

estiver usando (provavelmente o Apache) e modificar o arquivo *configure.php* para refletir as configurações reais do banco de dados e do servidor web (usuário, senha, URL). A última etapa é instalar e ativar o gerenciador de saída (*database feeder*). Talvez seja preciso alterar novamente as credenciais de usuários do banco de dados.

O IPtables Log Analyzer possui três variantes: *feed_db.pl*, *feed_db-shorewall.pl* e *feed_db-suse.php*. Para rodar o feeder automaticamente durante o boot, é preciso copiar o script de inicialização *scripts/iptablelog* para o diretório */etc/init.d* e criar os links simbólicos apropriados nos diretórios *rc*.

wflogs

O wflogs é a ferramenta de análise do projeto Wallfire [2], embora possa ser usada independentemente dos outros módulos. O programa interpreta e processa arquivos de log dos firewalls baseados em Netfilter, IPchains, IPfilter, Cisco PIX, Cisco IOS e do sistema de detecção de intrusos (IDS - *Intrusion Detection System*) Snort. Os relatórios de evento podem ser produzidos em texto puro, HTML e XML, além de um interessante modo interativo, em que os eventos são mostrados em tempo real. O wflogs não guarda as informações processadas em um banco de dados, mas pode converter entre os formatos de arquivo do Netfilter, IPchains e Ipfilter.

Instalar o wflogs em um sistema Debian é tarefa simples; basta usar o APT. O Debian versão *Sid* inclui o wflogs nos repositórios oficiais. Para o Debian estável (*Woody*) o programa pode ser baixado de [8]. Usuários de outras distribuições têm a opção de compilar o wflogs a partir dos fontes – quem gosta de RPM não terá sorte desta vez. Será necessário, antes, instalar a biblioteca Wfnetobjs, outro componente do Wallfire [2]. Recomenda-se ainda a instalação da biblioteca alternativa de DNS, *adns* [9], para que seja possível fazer resolução de nomes por DNS assíncrono.

Para compilar o wflogs, siga as etapas de sempre, com os comandos: *./configure; make; make install*. Talvez seja preciso especificar, no script *configure*, o diretório onde a biblioteca *WFnetojbs* se encontra.

De Netfilter a HTML em poucos passos

O wflogs pode processar logs de firewall tanto online quanto offline. O comando a seguir cria um relatório simplificado formatado em HTML e gerado a partir de um arquivo de log do Netfilter (figura 2):

```
wflogs -i netfilter -o html ➘
netfilter.log > logs.html
```

No modo *real time*, o wflogs analisa cada novo evento registrado no log e, depois de processados, joga todos na tela. Os administradores podem usar um shell para mudar o comportamento do wflogs interativamente. Por exemplo, o comando a seguir diz ao wflogs para monitorar o arquivo */var/log/warn* em tempo real:

```
wflogs -RI -o human ➘
/var/log/warn
```

A opção *-P* obriga o wflogs a processar mensagens antigas no arquivo. O wflogs ignora mensagens que não sejam específicas de firewall.

Figura 2: A página *wflogs Summary* (resumo do wflogs) mostra quantos pacotes foram registrados para cada origem.

Listagem 1: Banco de dados MySQL

```
# mysql -u root -p
mysql> create database iptables;
mysql> grant create,select,insert on iptables.* to iptables_admin@localhost identified by 'g3h31m';
mysql> grant create,select on iptables.* to iptables_user@localhost identified by 'auchgeheim';
mysql> quit
# cat sql/db.sql | mysql -u iptables_admin -p iptables
```

Filtrando mensagens

Há opções de filtragem extremamente poderosas que podem restringir a exibição de mensagens a tipos bastante específicos. O filtro a seguir foi retirado da documentação do wflogs. Ele lista apenas as conexões bloqueadas de Telnet e SSH ocorridas nos últimos três dias e vindas da rede 10.0.0.0/8:

```
wflogs -f '$start_time >= [this 3 days ago] && $start_time < [this 2 days ago] && $chainlabel =~ /(DROP|REJECT)/ && $sipaddr == 10.0.0.0/8 && $protocol == tcp && ($dport == ssh || $dport == telnet) && ($tcpflags & SYN)' -i netfilter -o text --summary=no
```

fwlogwatch

Boris Wesslowski desenvolveu o fwlogwatch para o RUS-CERT na Universidade de Stuttgart, Alemanha. A versão 1.0 do programa [3] foi, finalmente, liberada sob a licença GPL.

O fwlogwatch possui três modos de operação: Log Summary Mode (modo de relatório resumido), Interactive Report Mode (modo interativo de relatórios) e Realtime Response Mode (modo interativo em tempo real). No modo Log Summary, o programa gera relatórios em texto puro ou HTML com os resumos da análise dos logs (figura 3). No Report Mode, o fwlogwatch automaticamente cria relatórios de incidentes que os administradores podem enviar às pessoas afetadas pelo incidente sempre que necessário.

No Realtime Mode, o fwlogwatch responde a ataques executando scripts, enviando mensagens de email ou automaticamente modificando as regras do firewall. Os administradores podem usar o servidor web embutido (não há necessidade de um Apache) para monitorar o estado do fwlogwatch.

O programa reconhece os arquivos de log de firewalls baseados em IPchains (opção *i*), Netfilter (*n*), IPfilter (*f*), IPFW (*b*), Cisco IOS (*c*), Cisco PIX (*p*), Netscreen (*e*), Windows XP (*w*), Elsa Lancom (*l*) e o IDS Snort (*s*). A instalação é feita com um simples *make && make install && make install-config*. Boris Wesslowski possui pacotes para Red Hat Linux e Debian no site oficial do fwlogwatch.

Os administradores podem configurar o comportamento do fwlogwatch usando como base o arquivo exemplo de configuração, bem documentado e com comentários informativos, ou pela linha de comando. A página de manual explica a sintaxe e funcionamento de todas as opções. Por exemplo, o comando mostrado a seguir executa o fwlogwatch em *Summary Mode*:

```
ureserver.info) port 80 (http) with TCP flags SYN
at Apr 30 12:28:48, 1 packet logged on host P15097491: chain IPTABLES ACCEPT: HI
TP-Zugriff, on input interface eth0, source mac address 00:00:5a:9d:10:ba, desti
nation mac address 00:20:ed:2f:ed:68, protocol tcp, from 80.58.0.172 (unknown ho
stname) (80.58.0.0/16 RIMA (Red IP Multi Acceso) AS3352 Internet Access Network
of TDE) port 50651 (unknown service name) to 217.160.128.61 (p15097491.pureserve
r.info) port 80 (http) with TCP flags SYN
at Apr 30 12:28:52, 1 packet logged on host P15097491: chain IPTABLES ACCEPT: SS
H-Zugriff, on input interface eth0, source mac address 00:00:5a:9d:10:ba, destin
ation mac address 00:20:ed:2f:ed:68, protocol tcp, from 212.204.17.133 (iD4CC118
5.versanet.de) port 64541 (unknown service name) to 217.160.128.61 (p15097491.pu
reserver.info) port 22 (ssh) with TCP flags SYN
/var/log/messages:3250: warning: line format matches none of the specified modul
e(s): netfilter
wflogs> help
help            Display this text.
?               Synonym for `help'.
quit            Quit.
exit            Synonym for `quit'.
beep            Set beep mode: [on|off|?]. Beep for every log entry displayed.
filter          Set filter expression: [expression|unset].
realtime        Set realtime mode: [on|off|?]. Monitor new log entries..
verbose         Set verbosity level: [level].
wflogs>
```

Figura 3: No *Summary Mode* o fwlogwatch dá aos administradores uma visão geral da atividade nos arquivos de log.

Figura 4: O servidor web embutido no fwlogwatch permite que o administrador monitore o estado atual do firewall.

```
fwlogwatch -b -Pn -U 'Spenneberg.Com'-p -n -N -o output.html -t -w /var/log/messages
```

A opção *-Pn* ativa o interpretador para os logs do Netfilter. Já *-U* permite que o usuário especifique um cabeçalho para o relatório. A opção *-o* especifica o arquivo de saída; *-w* estipula que o relatório será gerado no formato HTML. *-n* e *-N* ativam a resolução de nomes para máquinas e serviços. Como resultado, obtemos um belo relatório formatado em HTML, mostrando toda a atividade de nosso firewall.

Resposta imediata

A opção de rodar o fwlogwatch em modo real time permite que os administradores reajam com ações e comandos às mensagens do arquivo de log. Ao mesmo tempo, o estado do firewall é mostrado em uma janela de navegador. O fwlogwatch roda em segundo plano como um daemon e monitora o arquivo de log. Se o daemon receber um sinal *SIGHUP*, o arquivo de configuração é lido novamente. Já o sinal *SIGUSR1* ordena ao daemon que reabra o arquivo de log. Esse recurso é bastante útil com arquivos de log rotativos.

Listagem 2: IPtables Log Analyzer

```
iptables -N LOG_DROP
iptables -A LOG_DROP -j LOG --log-tcp-options --log-ip-options --log-prefix '[IPTABLES DROP] : '
iptables -A LOG_DROP -j DROP
iptables -N LOG_ACCEPT
iptables -A LOG_ACCEPT -j LOG --log-tcp-options --log-ip-options --log-prefix '[IPTABLES ACCEPT] : '
iptables -A LOG_ACCEPT -j ACCEPT
```

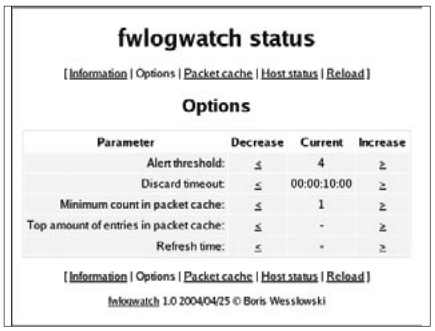


Figura 5: Os administradores podem usar o navegador para configurar o fwlogwatch. O *Alert Threshold* (Limiar de Alerta) especifica o número de mensagens necessárias para disparar uma contramedida.

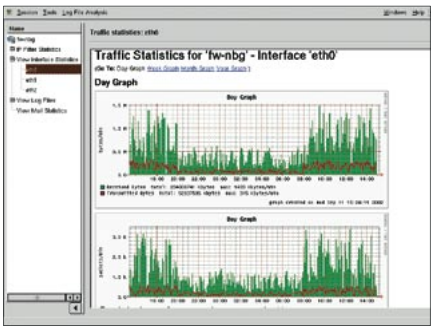


Figura 6: Apesar de ser uma ferramenta interessante, o *SuSE Firewall* não será mais mantido.

Os administradores podem especificar valores que definam qual a gravidade necessária nas mensagens para que o fwlogwatch reaja, emitindo alertas ou executando scripts de retaliação. Há duas opções importantes: *recent* (*-l*) define o período de tempo a monitorar, enquanto *alert_threshold* (*-a*) define, dentro desse período, o número de eventos que dispara uma resposta. A listagem 3 mostra um exemplo que configura o fwlogwatch em modo *Real Time* com o interpretador para arquivos de log do Netfilter. O processo roda como o usuário *fwloguser*.

Listagem 3: fwlogwatch Realtime Mode Analyzer

```
realtime_response = yes
parser = n
run_as = fwloguser
recent = 600
alert_threshold = 5
notify = yes
notification_script = /usr/sbin/fwlw_notify
server_status = yes
bind_to = 127.0.0.1
listen_port = 8888
status_user = ralf
status_password = iOQlAmOg4PrAA
refresh = 10
```

Se o limiar de cinco conexões em 600 segundos (dez minutos) é excedido, o fwlogwatch dispara uma ação configurável. O servidor web interno roda no endereço 127.0.0.1:8888, no qual o usuário *ralf* pode "logar-se" com a senha *password*. As senhas são criptografadas com o algoritmo DES, que podem ser geradas com o comando *htpasswd -nb usuário senha*. Quando o usuário registra-se na página, depara-se com algo parecido com a figura 4. Uma grande quantidade de opções pode ser alterada nessa interface web (figura 5).

Escolhas

O fwlogwatch possui um vasto cardápio de recursos, desde a exibição de resumos simplificados ao poderoso modo em tempo real com respostas configuráveis. Mas as outras ferramentas mostradas no artigo também merecem ser consideradas e testadas. Se você precisa de uma filtragem fenomenal, o wflogs pode ser sua melhor opção. O IPtables Log Analyzer é interessante porque usa um banco de dados para guardar as informações recolhidas. Para quem conhece, a opção de usar declarações na linguagem SQL para fazer pesquisas nas mensagens do firewall baseadas em critérios arbitrários é bem interessante e poderosa. Muito mais poderosa do que depender do *front-end* via web oferecido pelas outras ferramentas. ■

INFORMAÇÕES

[1] IPtables Log Analyzer: *http://www.gege.org/iptables/*

[2] Projeto Wallfire (wflogs e Wfnetobjs): *http://www.wallfire.org*

[3] FWlogwatch: *http://fwlogwatch.inside-security.de*

[4] Ulogd: *http://gnumonks.org/projects/ulogd*

[5] Ulogd PHP: *http://www.inl.fr/download/ulog-php.html*

[6] Shorewall firewall: *http://shorewall.sourceforge.net*

[7] SuSE firewall: *http://www.suse.de/en/business/products/suse_business/firewall/*

[8] Pacotes do wflogs para o Debian Woody. coloque, em seu sources.list, a linha *deb http://people.debian.org/~kelbert/ stable main*

[9] GNU adns: *http://www.chiark.greenend.org.uk/~ian/adns/*

SOBRE O AUTOR

Ralf Spenneberg é um instrutor freelance *de Unix e Linux. Em 2002 publicou o livro "Intrusion Detection for Linux Servers", rapidamente seguido de "VPNs on Linux". Em breve, mais um livro seu, "Intrusion, Detection and Prevention with Snort and Co.", estará nas livrarias.*

Implementando um firewall de camada 2 (bridge)

Ponte levadiça

Os firewalls são, tipicamente, implementados como roteadores. Entretanto, as coisas não precisam ser assim. Filtros de pacotes baseados no princípio de *bridging* possuem algumas vantagens, entre elas a de poderem ser inseridos em qualquer ponto de sua rede sem absolutamente alterar a topologia e a configuração de seus nós. POR RALF SPENNEBERG

O Linux alcançou uma respeitável reputação de excelente plataforma para firewalls. O kernel possui um eficientíssimo filtro de pacotes, o subsistema netfilter/iptables. Em um firewall tradicional, o netfilter é colocado para funcionar em uma máquina que serve de roteadora, onde divide a rede em duas ou mais subredes. Entretanto, adicionar um firewall em uma rede já estabelecida pode envolver uma drástica alteração na infra-estrutura do cabeamento e do próprio projeto lógico. A dor de cabeça pode ser tamanha que talvez seja necessário mudar todos os endereços IP das estações e até impor mais restrições no acesso aos serviços internos.

Em vez dessa complicação toda, que tal instalar uma *bridge*? São muito mais simples de operar e não causam nenhum impacto numa rede já existente. As bridges (em português, "pontes") são dispositivos que trabalham na camada 2 do modelo de referência OSI e normalmente inspecionam os endereços MAC (ou seja, os endereços físicos das placas de rede) em vez do endereço IP – consulte o quadro "Construindo Pontes". O Linux pode tirar partido dessa capacidade para criar firewalls "transparentes". Obviamente, a bridge não deixará de inspecionar pacotes dos protocolos de camadas mais altas (endereços IP, portas TCP e UDP) em sua tarefa de firewall. A grande vantagem é que os computadores da rede não vão sequer notar que há um firewall no caminho. Para eles, há uma ligação direta com a Internet, mas alguma "entidade espiritual" bloqueia o envio de pacotes ilegais.

Configuração do kernel

Lennert Buytenhek e Bart de Schuymer escreveram um patch que adiciona um modo de *bridging firewall* ao kernel 2.4. Se você usa o kernel 2.6 nem precisa aplicar nada: o recurso já existe, basta apenas configurar o núcleo do sistema de acordo (ver figuras 1 e 2).

Todas as opções de bridging no grupo *netfilter* são importantes. Por exemplo, o suporte a tabelas ARP (ARP Tables) nos módulos *IP_NF_ARPTBLES*, *IP_NF_ARPFILTER* e *IP_NF_ARP_MANGLE*. Essas funções podem ser implementadas como módulos ou compiladas de forma monolítica no kernel. A opção *Physdev match* também é importante; o módulo chama-se *IP_NF_MATCH_PHYSDEV*. Ela é necessária para que as séries 2.6 e superiores possam consultar a interface física (camada 2) e, assim, aplicar as regras de filtragem nos pacotes que passam por elas.

Depois de compilar sem erros (e colocar para rodar) o novo kernel, são necessárias mais algumas ferramentas no espaço do usuário para que possamos montar nosso bridgewall. Embora muitas distribuições já tenham, por padrão, o programa *iptables* disponível, é bem provável que seja necessário instalar os utilitários *arptables* e *ebtables* na maioria delas. Se você está rodando uma versão atual do kernel 2.6 em uma distribuição mais antiga (que vinha

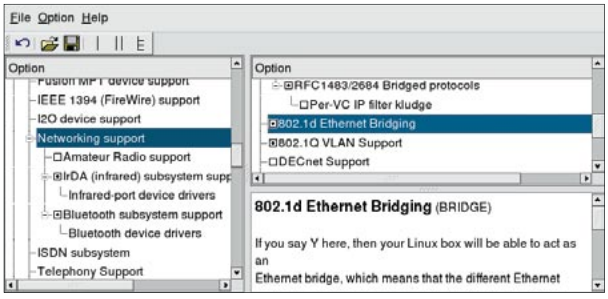


Figura 1: Para ativar o modo bridge no kernel 2.6, marque a opção *802.1d Ethernet Bridging* no grupo *Networking Support*.

com o kernel 2.4, por exemplo) também será necessário atualizar o pacote com o utilitário *iptables*.

O pacote bridge-utils [2] também deve ser instalado. Ele será usado para configurar a bridge. As distribuições modernas normalmente disponibilizam esse pacote por padrão. Nele encontramos o comando *brctl*, de uso exclusivo do usuário root. Se o emitirmos assim:

```
brctl addbr br0
```

criamos uma bridge chamada *br0*. Já o comando a seguir:

```
ip link show br0
```

confirma que a bridge existe. Como ela possui um nome, pode-se até rodar diversas bridges virtuais em uma única máquina com Linux!

A seguir, a bridge precisa saber qual placa de rede Ethernet deve administrar. O comando *brctl* permite adicionar interfaces à bridge:

```
brctl addif br0 eth0
brctl addif br0 eth1
```

As placas de rede não devem estar configuradas neste ponto; ou seja, não devem estar ativas (*UP*) ou possuir qualquer endereço IP. A idéia é ativá-las só depois de estarem associadas à nossa bridge.

```
ip link set eth0 up
ip link set eth1 up
```

Construindo Pontes

O termo *bridge* (ponte) refere-se a uma categoria de dispositivos que manipulam e redirecionam pacotes de rede na camada 2 do modelo OSI. Além das próprias bridges em si, muitos equipamentos populares, presentes em quase todas as redes do planeta, são um tipo de bridge. Um deles é o manjadíssimo *switch* (em português, *comutador*, embora infelizmente o termo em inglês seja mais usado). Para redirecionar pacotes na camada 2, a bridge precisa relacionar todos os endereços MAC da rede local e lembrar-se quais computadores estão ligados em cada uma de suas saídas, chamadas de *portas*. Quando a bridge recebe um pacote de um endereço MAC já conhecido, simplesmente redireciona o pacote para a saída correta, reduzindo bastante o tráfego nas outras portas. Se a bridge não souber para onde deve enviar o pacote, faz um *broadcast* (difusão) e o envia a todas as suas portas.

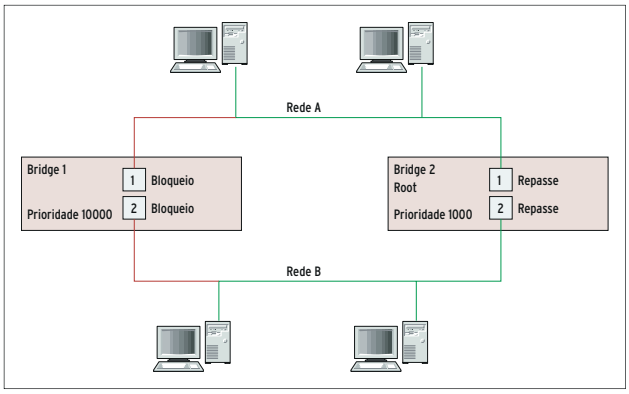


Figura 4: As bridges 1 e 2 conectam as redes A e B. A bridge de menor prioridade usa o protocolo STP para definir os caminhos válidos para pacotes enviados entre A e B.

O comando *brctl showmacs br0* lista todos os endereços MAC que a bridge já conhece. O resultado é mostrado em forma de tabela. A primeira coluna contém o número da porta para a qual o pacote deve ser enviado, pois o computador de destino está conectado a ela. A segunda coluna contém o endereço MAC desse computador. As demais colunas mostram outros dados sobre cada nó da rede.

Memória Fraca

Para ficar sempre atualizada, a bridge remove os endereços MAC mais antigos da tabela ARP. É possível especificar quanto tempo um endereço MAC que não esteja em uso pode ficar na tabela antes que a bridge o descarte. O comando apropriado é *brctl setageingtime br0 tempo_em_segundos*.

O esforço computacional interno seria desnecessariamente alto se a bridge descartasse imediatamente endereços MAC inativos. Em vez disso, a bridge primeiro marca os endereços ociosos e os remove posteriormente a intervalos regulares. O processo é conhecido como *garbage collection* (numa tradução aproximada, "coleta de lixo"). Esse intervalo pode ser ajustado com o comando *brctl setgcint br0 tempo_em_segundos*. O valor padrão é 0 (zero).

Spanning Tree Protocol

Switches modernos usam o *spanning tree protocol* (protocolo de árvore distribuída ou STP) para oferecer uma configuração de alta disponibilidade. Com esse protocolo, dois ou mais switches conectam subredes entre si e, dessa forma, descobrem todas as rotas possíveis entre elas. Depois de eleito, um switch mestre (ou "raiz") define os caminhos ativos e inativos para acesso a cada uma das subredes e propagam essa informação a todos os switches envolvidos.

Os switches restantes bloqueiam os caminhos inativos e suas interfaces, impedindo assim que pacotes duplicados (ou seja, um mesmo pacote que tomou dois caminhos diferentes) cheguem até a rede de destino (ver figura 4). Se um switch apresenta algum mau funcionamento, os switches remanescentes descobrem quais caminhos alternativos ainda estão disponíveis e contornam o equipamento defeituoso.

O Linux suporta o protocolo STP; entretanto, o administrador precisa emitir o comando *brctl stp br0 on* para ativá-lo. Um valor de prioridade entre 0 e 65535 pode ser associado à bridge: *brctl setbridgeprio br0 prioridade*. A bridge com o menor número de prioridade (portanto, com a prioridade mais alta) assume a função de "raiz".

1, 2, 3: testando...

As bridges verificam se suas colegas estão "vivas" mandando mensagens de teste ("hello") a intervalos regulares. Os administradores podem determinar o intervalo de envio dessas mensagens com o comando *brctl sethello br0 tempo_em_segundos*. o comando *brctl setmaxage br0 tempo_em_segundos* especifica quanto tempo as outras bridges devem esperar para receber a mensagem de "hello" das companheiras. Como um grande formigueiro, todo o sistema de bridges assume que a bridge que não responder depois desse intervalo está "morta".

Quando uma bridge está conectada a uma rede, precisa esperar um período de tempo arbitrário antes de começar a redirecionar pacotes. Essa espera é necessária para verificar se a rede está usando o protocolo STP. O comando *brctl setfd br0 tempo_em_segundos* ajusta o tempo de espera.

Apesar da explicação acima, o protocolo STP deve estar *completamente desativado* em uma bridge que vá atuar como firewall por filtragem de pacotes: *brctl stp br0 off*. O firewall tem que se basear em seu próprio conjunto de regras e não pode, sob hipótese alguma, ser desabilitado por *spoofing* de STP.

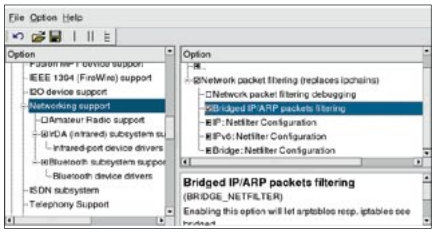


Figura 2: Não esqueça de marcar a opção *Bridged IP/ARP packets filtering* na configuração do netfilter dentro do kernel!

```
ip link set br0 up
```

Nossa bridge está, a partir de agora, pronta para entrar em ação! Veja as informações obtidas com o comando *ip link show* (figura 3). A bridge redireciona os pacotes e mantém sua própria tabela ARP, usando esse cache para rastrear quais endereços MAC estão conectados a (isto é, podem ser acessados por) cada interface (veja mais detalhes no quadro "Construindo Pontes", na página ao lado).

Bridgewalling

Como qualquer firewall, a bridge deve possuir um conjunto de regras que defina quais pacotes podem passar e quais devem ser bloqueados. Há três comandos para criação dessas regras: *iptables*, *ebtables* e *arptables*.

Todos os pacotes que a bridge redireciona (entram por uma porta e saem por outra) passam pela cadeia *FORWARD* do netfilter. Há algumas coisas a observar quando usamos o comando *iptables* em uma bridge. Se uma regra especifica que os pacotes devem atravessar a bridge em apenas uma direção, deve-se usar a opção *-m physdev* (tabela 1). Isso permite que a política de conexão identifique a porta da bridge pela qual o pacote entrou, ou mesmo se ele chegou a ser manipulado por ela.

O exemplo a seguir permite conexões SSH ao endereço IP 192.168.0.16 (porta TCP/22) em apenas uma direção. O servidor SSH está conectado à placa de rede *eth1*. As conexões só podem ser estabelecidas por clientes conectados à *eth0*. Precisamos de dois comandos *iptables* para definir a regra:

```
iptables -A FORWARD -m physdev
  --physdev-in eth0 ➥
  --physdev-out eth1 ➥
  --dport 22 -d 192.168.0.16➥
   -m state ➥
  --state NEW -j ACCEPT
iptables -A FORWARD -m physdev ➥
  --physdev-is-bridged -m state ➥
  --state ESTABLISHED,RELATED ➥
  -j ACCEPT
```

O primeiro comando cuida do estabelecimento da conexão, que deve ocorrer em apenas uma direção. O segundo comando permite que os pacotes que pertencem a essa conexão já estabelecida possam passar pelo firewall.

Coleta de endereços

Os firewalls baseados em bridges normalmente são usados para melhorar a segurança de redes existentes e já bem grandinhas. Com eles, não existe a necessidade de mudar a estrutura da rede e, muito menos, os endereços IP. Um dos pontos em que um firewall desse tipo pode ser bastante útil é na frente de um roteador que o administrador não possa alterar ou que não possua funcionalidade de firewall. Mas os

Figura 3: A bridge virtual já está no ar! O comando *ip link show* mostra os dados da bridge após o *4: br0:*.

bridgewalls são realmente magistrais quando usados para dividir grandes redes em áreas menores, com menos tráfego e gerenciamento melhor.

Nesse caso, os endereços IP de ambos os lados da bridge não podem ser alterados para pertencer a classes ou subredes diferentes, como seria o caso de um firewall comum baseado em roteador. Portanto, para que possamos filtrar o tráfego baseado nos números IP precisamos recorrer a um truque. O comando *ipset* dá aos administradores um meio de subdividir a rede, criando um banco de endereços no qual se podem coletar IPs arbitrários. Os comandos a seguir criam um banco chamado *left* e adicionam três endereços IP ao banco:

```
ipset -F; ipset -X; ipset -N ➥
left iphash
ipset -A left 192.168.0.5
ipset -A left 192.168.0.17
ipset -A left 192.168.0.18
```

O novo banco pode ser usado nominalmente em regras do iptables, com a opção *-m set*. O comando *--set nome_do_banco* indica o banco a ser considerado:

```
iptables -A FORWARD -m physdev ➥
--physdev-in eth0 ➥
--physdev-out eth1 ➥
--dport 22 -m set ➥
--set left -m state ➥
--state NEW -j ACCEPT
```

Tabela 1: Regras com *physdev*

Opção	Significado
--physdev-in *Nome*	Especifica a porta pela qual o pacote precisa entrar na bridge para que esta regra se aplique a ele.
--physdev-out *Nome*	Especifica a porta pela qual o pacote precisa sair da bridge para que esta regra se aplique a ele.
--physdev-is-in	O pacote veio de uma interface diretamente conectada à bridge.
--physdev-is-out	O pacote deixará o computador por uma interface diretamente conectada a uma bridge.
--physdev-is-bridged	O pacote trafega por dentro da bridge.

Listagem 1: MAC-NAT

```
ebtables -t nat -A PREROUTING -p ➥
ARP --arp-ip-dst  -j arpreply ➥
--arpreply-mac 0:ff:90:2b:a6:16
ebtables -t nat -A PREROUTING ➥
-p IPv4 -d 0:ff:90:2b:a6:16 ➥
--ip-dst 192.168.0.16 -j ➥
dnat --to-dst fe:fd:0:0:0:1 ➥
--dnat-target ACCEPT
ebtables -t nat -A POSTROUTING ➥
-p IPv4 -s fe:fd:0:0:0:1 -j ➥
snat --to-src 0:ff:90:2b:a6:16 ➥
--snat-target ACCEPT
```

Além dos pacotes IP, os pacotes ARP são úteis em regras de firewall. Muitos ataques originados dentro da própria rede são baseados em requisições e respostas ARP forjadas (*ARP spoofing*).

Tabelas ARP

O comando *arptables* pode filtrar pacotes ARP. Afora operações de *bridging*, o comando só deve ser usado nas cadeias *INPUT* e *OUTPUT*, pois os roteadores não encaminham pacotes ARP de uma interface a outra. Entretanto, em uma bridge, é possível filtrar pacotes ARP também na cadeia *FORWARD*. A sintaxe do comando é similar à do *iptables*. O *arptables* usa os alvos *ACCEPT* e *DROP*; *REJECT* não faz nenhum sentido em se tratando de pacotes ARP.

```
arptables -A FORWARD -s ! ➥
192.168.0.15 --destination-mac ➥
fe:fd:00:00:00:01 -j DROP
```

O comando acima descarta todos os pacotes ARP de resposta enviados ao computador cujo endereço MAC é *fe:fd:00:00:00:01* e que não foram originados pelo computador cujo IP é 192.168.0.15. As respostas ARP informam à entidade requisitante qual o endereço MAC do computador cujo IP foi especificado na pergunta. Aqui, o computador com o endereço MAC *fe:fd:00:00:00:01*, que vive em uma rede do outro lado da nossa bridge, só consegue enxergar o endereço MAC do computador cujo IP é 192.168.0.15.

Tabelas Ethernet

O comando *ebtables* é muito mais poderoso, permitindo que se faça coisas fabulosas como, por exemplo, NAT de endereços MAC! Com NAT, a bridge pode evitar que invasores descubram endereços MAC de computadores conectados a outras portas. A bridge manda seu próprio MAC como resposta a uma requisição ARP e, a partir daí, faz a tradução MAC-NAT para todos os pacotes IP que a atravessarem. O primeiro comando da listagem 1 diz à bridge para responder com seu próprio MAC (00:ff:90:2B:A6:16) a qualquer requisição ARP destinada a descobrir o endereço MAC do IP 192.168.0.16.

O endereço IP do computador que se quer esconder atrás da bridge precisa ser escrito atrás da opção *--arp-ip-dst*. A opção *--arpreply-mac* é o endereço MAC da bridge. Para MAC-NAT de pacotes IP, pode-se precisar também dos comandos nas linhas 2 e 3 da Listagem 1. Em nosso exemplo, 192.168.0.16 é o endereço IP do computador a ser escondido por detrás da bridge; e seu endereço MAC é *fe:fd:00:00:00:01*.

A documentação no site oficial do ebtables [3] fornece mais informações sobre as capacidades deste comando.

Nas profundezas da rede

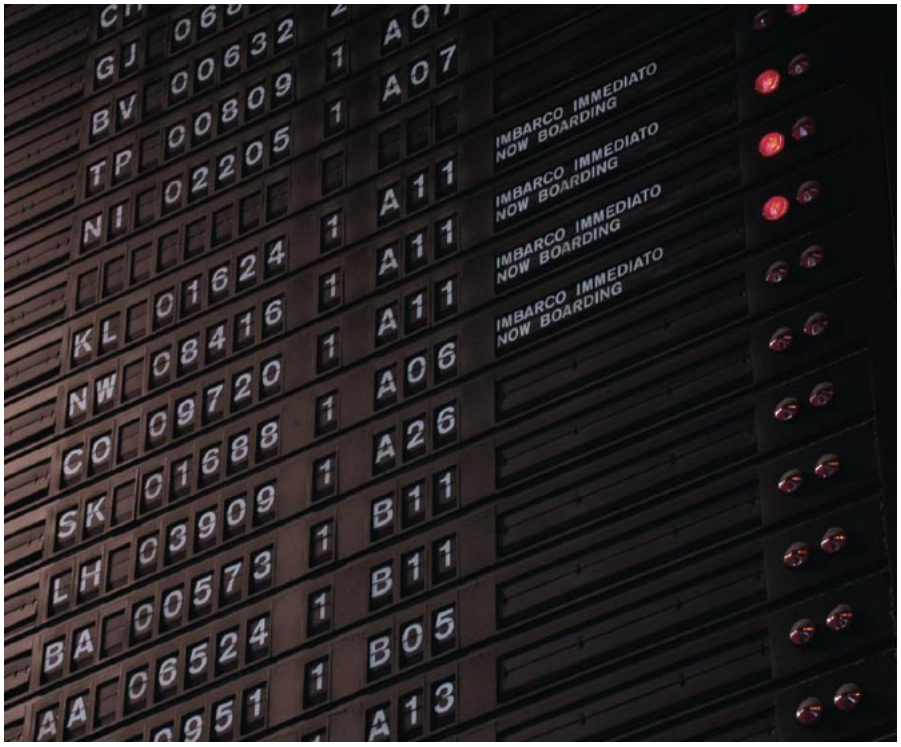
A técnica de *bridgewalling* dá aos administradores uma nova classe de filtros de pacotes que abre imensas possibilidades de controle sobre protocolos de camada 2. Mas a coisa mais espetacular em relação ao bridging é a possibilidade de inserir um firewall transparente entre dois pontos de uma rede pré-existente. A bridge simplesmente substitui o hub, o switch ou mesmo um cabo *cross-over*. E se for necessário bloquear alguns computadores suspeitos, é possível fazê-lo sem ter que reprojetar toda a numeração IP de sua rede. Basta colocar em operação um *bridgewall*. ■

INFORMAÇÕES

[1] iptables: *http://www.iptables.org*
[2] Linux bridge: *http://bridge.sf.net*
[3] ebtables: *http://ebtables.sf.net*

Configurando o Netfilter/IPtables com o Shorewall

Cumprindo tabela



Andrea Jaccarino: www.sxc.hu

Quando os usuários pensam em suas estações de trabalho ou seus computadores domésticos, geralmente se esquecem da segurança. Mas o perigo está à espreita, esperando para esfaquear os incautos usuários. POR JAMES MOHR

A única maneira de tornar seu computador completamente imune a ataques é desligá-lo da Internet. Sempre que você abre a porta de sua casa para sair à rua, algum gatuno pode esgueirar-se e entrar. Da mesma forma, os potenciais crackers estão à espera de portas abertas para invadir seu sistema.

Alguns usuários tem a errônea impressão de que os intrusos atacam apenas máquinas caras de empresas conhecidas e não estariam interessados em um computador doméstico. Ledo engano. A verdade é que cada computador conectado na Internet – mesmo os mais insignificantes – é uma vítima potencial de ataques e pode, inclusive, ser usado para originar ataques a uma terceira pessoa. Nem os usuários de conexões discadas estão a salvo. Na maioria dos casos, esses ataques são perpetrados por pessoas com nível técnico não tão bom assim, que fazem tentativas seguindo uma longa lista de falhas de segurança já publicamente conhecidas.

Para frustrar esses pivetes digitais, pode-se simplesmente fechar todas as portas de saída. Entretanto, se você precisa oferecer serviços à Internet – por exemplo, montar um servidor web – é preciso usar outra forma de proteção.

Mesmo as microempresas precisam da proteção oferecida por um firewall. Muitos usuários não têm poder aquisitivo para bancar um produto comercial. Como usuários de Software Livre, há soluções abertas que nos dão a proteção necessária. Uma delas é o Shorewall.

Nos bastidores

O Shorewall é o nome mais conhecido do produto chamado Shoreline Firewall. Do ponto de vista do usuário, o Shorewall é um conjunto de arquivos fáceis de configurar e, com eles, construir um firewall baseado no Netfilter [1]. Este, por sua vez, é um recurso presente no kernel das séries 2.4.x e 2.6.x que permite que diversos módulos do próprio kernel acessem o protocolo de rede em vários lugares. Por serem partes do núcleo do sistema, os módulos em questão possuem poderes quase esotéricos e podem fazer praticamente qualquer coisa com um pacote, desde simplesmente bloquear sua passagem até manipular seu conteúdo ou cabeçalho.

O Netfilter também trabalha com o velho ipchains, o filtro de pacotes do kernel 2.2, mas precisa ser explicitamente colocado em "ipchains compatibility mode" (modo de compatibilidade com o ipchains) para que funcione.

Você pode baixar a versão mais atual no site oficial do programa em [2]. Estão disponíveis versões em pacotes RPM ou arquivos tar com o código fonte. Os pacotes RPM não foram testados em todas as distribuições, apenas com as "majors": SuSE, Redhat e Mandrake.

Verifique o site para mais detalhes. Usuários do Debian podem usar o sistema APT, pois o Shorewall está disponível nos repositórios oficiais.

Para que o Shorewall funcione, é preciso ter instalados os pacotes *iptables* e *iproute/iproute2*. Esses pacotes são instalados por padrão na maioria das distribuições, portanto não devem ser empecilhos. A razão da necessidade do iptables é simples: o Shorewall não é, por si só, um firewall. Trocando em miúdos, o Shorewall não é responsável pela verificação, filtragem e manipulação de pacotes de rede. Pelo contrário: o Shorewall simplesmente lê seus arquivos de configuração e usa o comando *iptables* para carregá-las no kernel.

Como o *iptables* assume a tarefa de manipular as tabelas de filtragem do kernel, o Shorewall não é necessário mais do que uma vez a cada boot. É possível, inclusive, ver o que realmente o programa faz olhando dentro dele. Não se assuste, não é preciso um editor hexadecimal nem "cavocar" no código fonte para isso: o programa *shorewall* (normalmente em */sbin/shorewall*) não é nada mais do que um shell script.

Para que o iptables saiba o que fazer, é preciso dizer ao kernel quais são as restrições que se quer impor ao tráfego. Os chamados *rulesets* (conjuntos de regras) são definidas no interior do iptables e consistem de uma conexão e um certo número de "classificadores". Cada ruleset determina se uma conexão em particular deve ser permitida ou bloqueada, se e como deve ser manipulada, regras para seu redirecionamento e assim por diante.

Esse conceito é basicamente o mesmo em qualquer firewall, seja comercial ou gratuito, e com qualquer número de máquinas, desde algumas poucas até centenas delas – embora nesse caso é provável que o administrador prefira quebrar a rede em segmentos menores. Cada segmento poderia ser gerenciado por seu próprio firewall – e o Shorewall é perfeito para isso.

Um detalhe a ser observado é que não é necessário ter um computador dedicado para ser o firewall. Embora seja a prática comum (e, geralmente, uma boa idéia), usuários domésticos provavelmente não têm espaço ou recursos financeiros para montar computadores extras só para agir como firewalls. Se sua estação de trabalho está conectada diretamente à Internet, por que não colocar o firewall diretamente nela?

Listagem 1: Um exemplo de arquivo *zones*

```
#ZONE DISPLAY COMMENT
net Net          a zona da Internet
loc Local        a rede local
fw  FW           o firewall
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
#A linha acima não deve ser apagada nem alterada.
```

Em casa, minha rede consiste de um computador com uma conexão ADSL à Internet, uma segunda máquina com Windows XP e outra com Linux. Cada uma delas serve a um propósito diferente e eu permito apenas as conexões apropriadas a cada uma delas, tanto saindo como entrando.

Como minha rede é bastante simples, preciso editar uns poucos arquivos de configuração. Por ser uma situação bem simples e comum, minha rede doméstica é um ótimo local para começar.

Configuração básica

O arquivo de configuração principal fica em */etc/shorewall/shorewall.conf*. O arquivo permite que se configure qualquer coisa, desde a ativação até a desativação do sistema. Embora seja possível configurar um grande número de valores nesse arquivo, ainda não encontrei razões suficientes para fazê-lo em minha rede doméstica.

Você já deve estar familiarizado com a palavra *segmento* para referir-se a porções específicas em uma rede. O Shorewall usa, para isso, o termo *zona*. Em casa, possuo quatro zonas: *fw* (o próprio firewall, ou seja, o meu computador), *net* (a Internet) e *loc* (a rede local). A quarta zona chama-se *games*, mas falaremos dela mais adiante.

Os nomes das zonas devem ser curtos (cinco caracteres ou menos) e podem conter letras e números. Observe que não é possível usar a zona especial *all*. A variável *FW*, no arquivo *shorewall.conf*, define a zona específica do seu firewall. O valor padrão para ela é *fw*. Lembre-se que não é possível usar esse nome em outra zona!

Mesmo que não esteja oferecendo nenhum serviço a computadores fora de sua rede local (ou seja, na Internet) ainda assim a zona da Internet será necessária. Lembre-se que as regras do IPtables definem conexões ponto-a-ponto, ou seja, precisam de um ponto de origem e outro de destino. Um desses pontos seria, talvez, a sua estação de trabalho, mas o outro certamente é a Internet. Portanto, é necessário definila como uma zona específica.

Esses nomes são apenas convenções. Embora sejam os valores *default* do Shorewall, pode-se usar o nome que bem entender, desde que haja consistência em todos os arquivos de configuração.

Por padrão, as permissões no diretório */etc/shorewall* são definidas com o valor 700, o que significa que apenas o proprietário (ou seja, o root) tem acesso aos arquivos. Mesmo acesso apenas de leitura para outros usuários seria perigoso, pois alguém *poderia* descobrir um furo de segurança e explorá-lo.

As zonas são definidas no arquivo */etc/shorewall/zones*. Cada linha possui três valores: o nome da zona (usado para referenciar essa zona nos outros arquivos), nome legível (que aparece

Listagem 2: Um exemplo de arquivo *interfaces*

```
#ZONE INTERFACE BROADCAST OPTIONS
net     ppp0    -       routefilter,norfc1918
loc     eth0    detect  -
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
#A linha acima não deve ser apagada nem alterada.
```

Listagem 3: um exemplo de arquivo *policy*

```
#SOURCE    DEST       POLICY     LOG LEVEL
fw         net        ACCEPT     info
fw         loc        ACCEPT     info
loc        net        REJECT     info
loc        fw         ACCEPT     info
net        all        DROP       info
all        all        REJECT     info
#LAST LINE -- ADD YOUR ENTRIES BEFORE THIS ONE -- DO NOT REMOVE
#A linha acima não deve ser apagada nem alterada.
```

quando o Shorewall está carregando as regras) e um comentário. A listagem 1 mostra um arquivo *zones* básico.

Definindo as regras de comunicação

Embora tenhamos definido as zonas, o firewall ainda não sabe como conectá-las. Não há associação entre o nome da zona e a rede que representa. Para isso, precisamos mexer no arquivo */etc/shorewall/interfaces*, que contém uma tabela com quatro colunas: zona, interface, broadcast e opções.

Em meu sistema, o arquivo *interfaces* parece-se com o da listagem 2.

A zona é o nome definido no arquivo *zones*. Já a interface é o nome (no Linux) da interface de rede conectada a essa zona. Por exemplo, minha conexão ADSL usa o protocolo Point-to-Point Protocol (PPP), portanto o nome da interface é *ppp0*. O nome da interface para a minha placa de rede Ethernet é *eth0*. Para descobrir os nomes de suas interfaces de rede, use o comando */sbin/ifconfig*. A coluna *broadcast* é o endereço de difusão da rede conectada àquela placa. Como era de se esperar, a coluna *options* especifica quaisquer opções que se queira usar.

Em meu sistema há duas linhas:

```
net ppp0 routefilter,norfc1918
loc eth0 detect -
```

A zona da Internet está conectada à interface *ppp0*. Já a zona local está associada à interface *eth0*. Como há três zonas por padrão, você deve estar se perguntando onde foi parar a linha referente ao firewall. Bem, de forma simples, a zona *firewall* conecta-se às outras por uma das interfaces já especificadas. Portanto, pense no arquivo *interface* como sendo um modo de definir quais interfaces a zona *firewall* usa para se comunicar com as outras zonas.

No caso da interface *ppp0*, a coluna *broadcast* possui um hífen (-). Como uma conexão PPP não possui broadcast, eu poderia ter deixado a coluna em branco. Entretanto, eu desejava especificar opções adicionais, portanto precisava do "-" para "guardar o lugar" da coluna broadcast e, assim, chegar à coluna *options*. Se não houvessem opções, essa coluna poderia ser deixada em branco. Como "mania" pessoal, entretanto, sempre coloco os hífens para me lembrar de que alguma coisa existe naquela posição.

A opção *routefilter* diz ao kernel para rejeitar, naquela interface, qualquer pacote que deveria ter sido roteado por outra interface. Em nosso caso, se a *ppp0* recebesse, de dentro para fora, um pacote que deveria sair pela *eth0*, este seria bloqueado imediatamente. Esse recurso é chamado de *anti-spoofing*.

A segunda opção, *norfc1918*, ordena ao kernel que não roteie endereços especificados como "privados" pela RFC 1918. A RFC (Request For Comments) de número 1918 é um documento do IETF (Internet Engineering Task Force) que lista as faixas de endereço IP que podem ser usadas por qualquer um sem precisar de permissão especial e que, *sob hipótese alguma*, devem ser roteadas para a Internet. Essa opção assegura que, realmente, não serão. Mais detalhes sobre a RFC 1918 podem ser obtidos em [3].

Nesse ponto deparei com um dilema. Quero que meus computadores possam acessar a Internet; mas, como possuem endereços RFC 1918, eles estão bloqueados. Como fazer para acessar a Internet, então? Já voltaremos a esse problema.

É possível configurar o Shorewall para comportar-se de qualquer maneira que desejemos, desde que manipulemos suas zonas. Cada zona pode possuir uma *política de funcionamento* (*policy*), configurada – como já deve ter ficado óbvio neste ponto da matéria – pelo arquivo *policy*. Nele, os campos são: *source* (a zona de origem), *dest* (a zona de destino), *policy* (a ação a ser tomada por padrão) e o *log level* (quanta informação registrar nos *logs*).

A Listagem 3 mostra um exemplo do arquivo *policy*. Observe atentamente as duas últimas linhas. A primeira afirma que qualquer tráfego desconhecido vindo da Internet é sumariamente bloqueado (ou seja, ignorado).

Há quatro opções possíveis para a política de conexões:

ACCEPT – Aceita os pedidos de conexão.
DROP – Ignora ("derruba" ou "bloqueia") o pedido de conexão.
REJECT – Além de ignorar, retorna uma mensagem de erro ao computador que requisitou a conexão.
CONTINUE – Permite colocar estações em mais de uma zona e aplicar políticas a todas elas.

O nível de registro determina quanta informação é enviada ao subsistema de registro de eventos do kernel, o *syslog*. Lembre-se sempre de que tudo isso é feito pelo *iptables*, que é, em última análise, outro subsistema do kernel. Tudo o que o Shorewall faz é especificar a prioridade das mensagens a serem registradas. Para mais informações sobre o registro de eventos do sistema, consulte a página de manual do *syslogd* (*man syslogd*).

As duas últimas linhas do meu arquivo *policy* são:

```
net all DROP info
all all REJECT info
```

A primeira linha nos diz que qualquer tráfego não coberto por nenhuma outra regra (veremos essas regras mais adiante) e que venha da Internet será sumariamente bloqueado (ação DROP). Entretanto, qualquer outro tipo de tráfego será rejeitado (ação REJECT), o que quer dizer que o emissor das men-

sagens bloqueadas receberá um aviso relatando o motivo do bloqueio. Pode parecer meio estranho, mas quando raciocinamos sobre o comportamento de ambas as ações vemos que faz todo o sentido do mundo. Quando um pacote chega ao firewall vindo da Internet e destina-se a uma porta ou serviço que bloqueamos voluntariamente, queremos simplesmente impedir sua entrada. *De forma alguma* queremos informar o que aconteceu ao remetente. Se este for um indivíduo mal-intencionado (um cracker, por exemplo) esse tipo de aviso pode ajudá-lo a contornar nossas barreiras de segurança. Entretanto, pacotes originados em qualquer outra rede que não a Internet (em nosso caso, minha máquina e minha rede interna) são rejeitados, permitindo que os programas-cliente possam ter alguma idéia do que aconteceu.

Note bem: estamos configurando o *comportamento padrão* das zonas. Mas o que isso significa? Simples: que se você não definir nenhuma regra específica para cada um dos tipos de conexão, o comportamento padrão definido no arquivo *policy* será aplicado.

No arquivo *policy* da rede que tenho em casa possuo apenas essa linha padrão para a zona *net*. Teoricamente, qualquer pacote vindo da Internet deveria ser bloqueado. Entretanto, possuo um servidor web em minha máquina com um bocado de informações de referência que gostaria de acessar do trabalho. Uma vez que essa conexão se dá através da Internet, a linha correspondente no arquivo *policy* impede que eu acesse meus bookmarks e anotações quando estiver longe de casa.

A resposta está no arquivo *rules*, que é o coração do Shorewall. Aqui definimos como o firewall deve se comportar para cada tipo específico de conexão. Podemos criar regras usando qualquer abstração criada pelo Shorewall: zonas, serviços de rede (portas), segmentos de rede, máquinas individuais e praticamente qualquer combinação desses elementos. Quando uma requisição de conexão chega vinda da Internet, o sistema primeiro verifica se alguma regra definida no arquivo *rules* pode decidir o que fazer com ela. Se nenhuma regra apropriada existir, usa-se a regra padrão definida no arquivo *policy*.

As colunas no arquivo *rules* são: ação a ser tomada, origem da requisição, destino da conexão, protocolo usado, porta destino, porta de origem e o destino original. Destino original? Calma, já veremos do que se trata.

Além das ações já vistas no arquivo *zone*, o arquivo *rules* guarda algumas surpresas. A ação DNAT permite que se faça o chamado NAT – *Network Address Translation* (tradução de endereços de rede). O DNAT – *Destination Network Address Translation* – é uma especialização do NAT e permite que requisições de conexão possam ser redirecionadas para computadores dentro da rede, com endereços diferentes do original e mesmo em portas diferentes.

A ação REDIRECT redirecionará conexões para portas específicas na mesma máquina – útil para redirecionar pedidos de conexão HTTP para o proxy local (por exemplo o squid [4]). Para redes pequenas, que não precisam ter acesso irrestrito à Internet, descobri que a ação REDIRECT é extremamente útil. Pode-se, por exemplo, permitir apenas tráfego HTTP para os clientes internos, bloqueando qualquer outra coisa. No caso da minha rede, faço algo parecido com a seguinte linha no arquivos *rules*:

```
REDIRECT loc 3128 tcp www - ➥
!10.2.38.0/24
```

Resumidamente, a ação acima diz que todas as conexões que usem o protocolo de transporte TCP e que requisitem um serviço de web (protocolo de aplicação HTTP, normalmente disponível na porta 80) sejam redirecionadas para a porta 3128. Essa é a porta em que o servidor proxy Squid "escuta".

Aqui temos a figura do destino original. Especificamos uma classe C inteira usando a notação CIDR (*Classless Inter-Domain Routing*): 10.2.38.0/24. O sinal de exclamação na frente do endereço indica que estamos *negando* a regra, ou seja, pacotes vindos de qualquer endereço são redirecionados ao Squid *exceto* os endereços pertencentes à rede 10.2.38.0/24. A razão é óbvia: quero que todos na minha rede local acessem o servidor web sem passar pelo proxy, reservado apenas ao tráfego externo.

A ação LOG primeiro registrará o evento no syslog. Ao contrário das outras ações, entretanto, não interromperá o processamento das regras abaixo dela. Em vez disso, passará o comando para a próxima regra assim que registrar o evento. Dessa forma, é possível registrar um evento e, depois, aplicar outra ação a um mesmo pacote.

A ação QUEUE redireciona pacotes para programas que rodem no espaço do usuário, que por sua vez os manipulam e os devolvem à pilha de rede.

Para configurar seu arquivo *rules*, assuma como regra geral que se deve *bloquear* tudo e liberar apenas os serviços necessários – e olhe lá! muitas pessoas fazem o contrário, começando com uma rede completamente escancarada e desligando, um a um, os serviços desnecessários – ou pior: desligando só o que já foi invadido! Essa é uma prática perigosíssima: se esquecer de fechar alguma coisa, pode ser (ou melhor, será) alvo de um ataque em breve. No primeiro caso, esquecer de abrir alguma coisa bloqueará momentaneamente algum serviço, que pode ser aberto facilmente depois. O que é uma ligeira inconveniência comparada ao risco real de ataque?

Escovando bits

Já deve estar óbvio para o leitor como a nomenclatura dos arquivos do Shorewall funciona. Por exemplo, é mais que evidente que o arquivo */etc/shorewall/hosts* permite a definição de máquinas (hosts) específicas. Normalmente não é necessário mexer nesse arquivo em redes pequenas. Quanto menor for a rede, mais provável que a configuração em uma zona seja configurada da mesma maneira (ou, mais especificamente, que todas as máquinas conectadas a uma mesma *interface* do firewall possuam configuração semelhante).

Talvez esse não seja seu caso e você precise de regras de acesso diferentes para cada máquina de uma mesma rede ou segmento. Há alguns meses meu filho começou a jogar um RPG via Internet. Para isso, precisava de mais do que acesso HTTP via proxy.

Para liberar esse acesso, defini um conjunto específico de computadores como uma nova zona. Para tanto, criei uma nova linha para ela no meu arquivo *zone*, à qual chamei *game*. Depois, criei

uma linha no arquivo *hosts* contendo, apenas, o computador do meu filho.

```
game eth0:10.2.38.13
```

Mas não é só. Lembra-se de que minha rede local era 10.2.38.0/24? Este é um dos endereços privados definidos na RFC 1918. Mesmo que eu quisesse rotear esses endereços pelo meu ADSL, posso apostar que meu provedor de Internet os bloquearia mais adiante. O que podemos fazer, então?

A solução é o que chamamos de *IP masquerading*, ou mascaramento de IP. Como o nome indica, um endereço IP válido (ou seja, roteável) é usado para "mascarar" os endereços IP privados. No meu caso, o endereço IP da linha ADSL (que possui um endereço válido) mascara os IPs da minha rede local. Portanto, preciso criar uma linha no arquivo */etc/shorewall/masq* semelhante à seguinte:

```
ppp0 10.2.38.0/24
```

A interpretação é simples: todo o tráfego vindo da rede 10.2.38.0/24 e que saia para a Internet através da interface *ppp0* tem que ser mascarado com o IP dessa interface.

Nesse ponto, a configuração básica do mascaramento já está pronta. Agora, entretanto, é que vem a parte *penosa*. Não foi fácil conseguir informações sobre o jogo e sobre como jogá-lo através de um firewall. Para falar bem a verdade, toda a documentação que encontrei falava praticamente a mesma coisa: se quiser jogar, desabilite seus firewalls!

Entretanto, quase que por acidente consegui fazer o jogo funcionar. Para tanto, coloquei o nível de registro (log) no arquivo *policy* em modo *debug* e observei detalhadamente o que acontecia, anotando todas as tentativas de conexão do computador de meu filho com *qualquer* máquina. Depois, verifiquei via consultas DNS se aquela máquina pertencia aos desenvolvedores do jogo. Por último, adicionei a porta específica usada pelo jogo ao meu arquivo *rules*.

Se eu quisesse, poderia ter adicionado regras para acesso ao serviço WWW (porta 80). Entretanto, como já uso Squid como proxy e meu filho precisava apenas do acesso ao jogo, continuei usando a cláusula REDIRECT para conexão a websites.

Caçando bugs

É bem possível (provável, até) que você tenha problemas ao configurar o Shorewall. Portanto, a capacidade de mostrar todos os detalhes de cada conexão é uma ferramenta valiosa.

Uma técnica útil de depuração é colocar o nível de *logging* em modo *debug*, o que gera uma quantidade espantosa de informação. Entrar em detalhes sobre os pormenores de cada evento está fora do escopo deste artigo. São, entretanto, fáceis de inferir mesmo sem verificar todas as linhas. Por exemplo, observe o registro a seguir:

```
Nov 1 11:19:32 saturn kernel: ➥
Shorewall:net2all:DROP:IN=ppp0 ➥
OUT= MAC=
SRC=1.2.3.4 DST=10.2.38.11 ➥
LEN=48 TOS=0x00 PREC=0x00 ➥
TTL=116 ID=47048 DF
PROTO=TCP SPT=1 292 DPT=1080 ➥
WINDOW=64240 RES=0x00 SYN URGP=0
```

Este é um registro padrão de */var/log/messages*. No começo temos a data, o nome do computador e o subsistema do syslog (no caso, o kernel). Depois disso vem a mensagem real. Como se pode notar, o pacote entrou pela ppp0 (meu modem ADSL) e o pacote foi bloqueado (DROP). Podemos ver, também, que a conexão *net2all* estava em uso, ou seja, o pacote estava vindo da Internet em direção a alguma das outras zonas.

Se voltarmos à discussão sobre o arquivo *policy*, veremos que o procedimento padrão era bloquear (DROP) qualquer pacote que viesse da Internet em direção a qualquer outra interface (*all*). Algum dos pacotes tentou chegar à porta de destino (DPT) de número 1080, a porta do serviço *socks*. Não possuo nada rodando ali e certamente não me comuniquei com ninguém que precisasse dela. Como não é uma porta "padrão", nem há um serviço associado a ela por *default*, não há razão para ninguém acessá-la pela Internet. Para mim parece óbvio: alguém estava me testando para tentar explorar uma falha conhecida do Windows.

Se o computador remoto (ou mesmo outro computador na mesma rede) continuasse a tentar acessar várias portas em minha máquina, eu poderia colocá-lo numa "lista negra". Para isso, basta colocar os IPs (ou redes inteiras, no formato CIDR) no arquivo */etc/shorewall/blacklist*. Isso quer dizer que, em detrimento de todas as regras que porventura permitissem acesso, pacotes vindos desses IPs são sempre bloqueados.

Por padrão, o Shorewall usa o subsistema de registro (*syslog*) do Linux para enviar as mensagens automaticamente para */var/log/messages*. Mesmo quando se configura o *syslogd* para enviar mensagens a outro arquivo, acho maçante que as mensagens do meu firewall estejam misturadas a mensagens do kernel.

Para resolver a questão usamos ULOG, que deve estar habilitado no kernel. A boa notícia é que isso é padrão nas distribuições modernas. A má é que o pacote *ulogd*, necessário para que os usuários possam usar o serviço, quase nunca está disponível nas distros. Em todo caso, pode-se baixá-lo de [5].

Uma vez que o *ulogd* esteja configurado e rodando, não se usa mais os níveis de registro do *syslogd* no arquivo *policy*. Em vez disso, use ULOG (tudo em maiúsculas). A configuração do *ulogd* é independente do *syslogd*, portanto qualquer mudança em */etc/syslog.conf* não afeta o *ulogd*.

Uma última dica: o Webmin possui um módulo para configuração do Shorewall, disponível em [6]. Para mais informações sobre o Webmin, visite o site oficial do programa em [7]. ■

INFORMAÇÕES

[1] Site oficial do Netfilter: *http://www.netfilter.org*

[2] Site oficial do Shorewall: *http://www.shorewall.net*

[3] Endereços para Redes Privadas (RFC 1918): *http://rfc.net/rfc1918.html*

[4] Proxy Squid: *http://www.squid-cache.org/*

[5] Ulog: *http://gnumonks.org/projects/ulogd*

[6] Módulo do Webmin para administrar o Shorewall: *http://www.webmin.com/download/modules/shorewall.wbm.gz*

[7] Site oficial do Webmin: *http://www.webmin.com*

Navegador leve do projeto Mozilla ganha mercado

The *quick firefox* jumped over the lazy explorer

Desde o fim da guerra dos browsers, em 1998, um navegador de Internet não chama tanto a atenção do público. Saiba mais sobre o Firefox e descubra o porquê. POR RAFAEL RIGUES

Tom Kolby: www.sxc.hu

O Firefox nasceu dos esforços de dois desenvolvedores, Dave Hyatt (que hoje trabalha para a Apple no browser *Safari*) e Blake Ross, que consideravam o navegador Mozilla pesado e com recursos demais. Ambos tinham razão: composto por navegador, cliente de e-mail, cliente de IRC e editor HTML, as primeiras versões do conjunto de aplicativos do projeto Mozilla (codinome *SeaMonkey*) se arrastavam a passos de tartaruga, mesmo em máquinas poderosas. Seria necessário uma "dieta" rigorosa se o projeto quisesse chegar a algum lugar.

Dave e Blake começaram desmembrando o programa. Separaram o "engine" de renderização de páginas HTML, chamado *Gecko*, e ao redor dele construíram uma interface em XUL, mesma linguagem usada para criar o navegador Mozilla. A primeira versão do Firefox (0.1, codinome Pescadero) foi lançada em 23 de Setembro de 2002.

Mais de dois anos e 19 versões depois, o Firefox é um fenômeno de popularidade. 20 milhões de cópias da versão 1.0 foram baixadas desde o seu lançamento em 9 de novembro de 2004. A participação no mercado de browsers já chega a 5% e, pela primeira vez em muitos anos, a participação do principal concorrente, o Internet Explorer, caiu. O motivo é simples: os usuários estão fartos de janelas pop-up, banners, sites que "seqüestram" seu navegador ou instalam software malicioso em seus sistemas e falhas de segurança que permitem que um site se passe por outro. Esses problemas não afetam o Firefox.

Instalação

As vantagens do Firefox se tornam aparentes logo na instalação: enquanto um download do Internet Explorer 6 via Windows Update pode chegar a 79 MB, o Firefox tem apenas 8,4 MB. As versões Linux e Windows trazem um instalador para ajudar a colocar o programa em seu sistema; basta seguir as instruções na tela. Na versão para Mac OS X o instalador sequer é necessário: após o download, basta arrastar o ícone do aplicativo para a pasta *Applications (Aplicativos)*. Muitas distribuições Linux recentes já trazem o Firefox instalado por padrão ou fornecem pacotes pré-compilados com o programa.

Outro destaque é o suporte à internacionalização. São 32 idiomas suportados, incluindo o português do Brasil e

línguas exóticas como asturiano, catalão e galês. Basta escolher seu favorito na hora do download.

Principais recursos

O Firefox é um mágico com muitos coelhos na cartola. Um dos mais visíveis e adorados é o bloqueador de popups, que impede que os sites abram aquelas irritantes janelinhas com propaganda. O recurso vem habilitado por padrão e há uma lista de exceções, na qual você indica os sites que podem abrir popups (por exemplo, o site do seu banco). Quando uma janela popup é bloqueada, uma mensagem surge no topo da página, dizendo: *Firefox prevented this site from opening a popup window. Click here for options...* (O Firefox impediu este site de abrir uma janela popup. Clique aqui para mais opções...). Clique na mensagem para bloquear todos os popups vindos do site, liberá-los, exibir a janela que foi bloqueada ou acessar as opções do bloqueador. Extensões como a AdBlock trazem ainda mais recursos e podem bloquear banners animados e propagandas em Flash, entre outras.

Outro recurso muito útil é a navegação com abas, ou "tabbed browsing". Ela permite ver vários sites em uma única janela do navegador, cada um em uma aba, o que elimina a confusão de múltiplas janelas do navegador entulhando o desktop. Para quem realmente aprecia esse recurso, a extensão *Tabbrowser Preferences* é essencial. Ela adiciona à janela de preferências do Firefox opções para "ajuste fino" do comportamento das abas.

Se você quer procurar por um termo em uma página Web, não precisa ir até o menu *Editar* | *Localizar* ou teclar *Control+F*. Assim como no *vi*, no leitor de man pages do Linux ou no *less*, basta teclar */*, seguido do termo, para fazer uma busca na página (um painelzinho de busca surge na parte inferior da janela do navegador). O Firefox automaticamente pula para a primeira ocorrência e a destaca em fundo verde. Os botões *Próxima* e *Anterior* no painel de busca encontram respectivamente a próxima ocorrência do termo e a anterior. *Realçar* destaca todas as ocorrências simultaneamente em amarelo, como se tivessem sido marcadas com uma caneta destaca-texto. E *Diferenciar Maiúsc./Minúsc.* faz com que a busca respeite maiúsculas e minúsculas.

Para quem tem problemas de visão ou odeia aqueles sites que propositadamente utilizam fontes praticamente ilegíveis, o atalho de teclado *Control* + é uma bênção. Use-o múltiplas vezes para aumentar o tamanho das fontes e deixá-las ao seu gosto. O atalho *Control* - tem o efeito inverso. Para quiosques de acesso à Internet ou terminais de informação em que o conteúdo é tudo o que importa, há o modo full-screen, acessado através da tecla *F11*.

Há uma forma de criar palavras que funcionam como atalhos para sites e sistemas de busca como o Google (que já tem um campo de busca ao lado da barra de endereços) e Wikipedia. Veja um exemplo:

1. Faça uma busca qualquer na Wikipedia e descubra qual é a expressão usada para essa busca na barra de endereços. Fizemos isso e descobrimos que ela é a seguinte (a palavra procurada foi Linux):

```
http://en.wikipedia.org/wiki/
Special:Search?search=Linux&go=Go
```

2. Faça um bookmark com essa URL. No menu *Favoritos*, clique com o botão direito do mouse sobre o item Wikipedia que você acabou de adicionar. Vai aparecer um menu de contexto. Escolha a opção *Propriedades*. Na janela de diálogo que aparece, troque agora a palavra chave na URL (Linux) por %s. O resultado final será:

```
http://en.wikipedia.org/wiki/
Special:Search?search=%s&go=Go
```

No campo *keyword* escreva *wp*, que vai ser o nosso atalho para Wikipedia.

Agora é só experimentar: na barra de endereços digite "wp" seguido do termo a buscar, como: *wp gentoo*. O resultado deve ser uma página da Wikipedia com os resultados da busca. Divirta-se!

O Firefox também é mais compatível com os padrões em vigor na Internet, como o "Cascaded Style Sheets 2" (CSS2), definidos pelo W3C (World Wide Web Consortium). E por não ser tão intimamente ligado ao sistema operacional, ao contrário do Internet Explorer, apresenta menor risco de ser usado como vetor para um ataque ou invasão ao sistema. Por padrão, o navegador bloqueia o download de extensões vindas de sites de terceiros e alerta sobre os riscos decorrentes da instalação de extensões e plugins não assinados. Além disso, o navegador Firefox busca atualizações críticas durante a inicialização. Sempre que uma está disponível, a página inicial (homepage) é substituída por um alerta informando sobre o problema, com um botão para download imediato de uma correção.

Extensões

Extensões são pequenos programas que se "anexam" ao navegador e expandem seus recursos. Eles variam desde pequenos inutilitários (que informam quantas mensagens existem na sua caixa postal, por exemplo), passando por tecnologias experimentais (como menus circulares, ou *pie menus*) até ferramentas como gerenciadores de downloads em massa (úteis para lidar de uma só vez com páginas lotadas com links para vários arquivos) e interfaces para escrita em blogs como o Blogger, Drupal, LiveJournal, MoveableType, e outros.

O site Firefox Central [1] lista quase duas centenas de extensões para o Firefox. Instalá-las é fácil; basta clicar no link *Install* logo abaixo da descrição da extensão. Uma janela surge perguntando se você deseja mesmo instalar o arquivo. Clique em *Instalar Agora* pra continuar. Depois, basta reiniciar o navegador para que a nova extensão funcione. Você pode removê-la, buscar por atualizações ou configurar opções extras no gerenciador de extensões, encontrado no menu *Ferramentas* | *Extensões*. A seguir, algumas das extensões mais populares ou úteis para o Firefox:

ChatZilla – Um cliente IRC completo dentro do seu navegador, que dispensa programas como o XChat ou mIRC e é compatível com o Mozilla e o Firefox em qualquer plataforma suportada por esses programas. Infelizmente, não fun-

Trivia

Além de nomes de criaturas mitológicas, *Firebird* e *Thunderbird* também são nomes de automóveis esportivos, produzidos pelas montadoras Pontiac (em 1967) e Ford (em 1954), respectivamente.

Figura 1: Chatzilla, um cliente IRC para o Firefox.

ciona com alguns derivados do Mozilla como o Galeon, o Epiphany (ambos para Linux) e o K-Meleon (para Windows). Tem apenas 271 KB e a interface pode ser personalizada com o uso de *motifs* (motivos), que alteram a aparência do texto e adicionam imagens de fundo, emoticons e outros itens às conversas. Um FAQ [2] tira as principais dúvidas relativas ao uso do programa.

DictionarySearch – Extensão muito útil para quem está aprendendo inglês e quer consultar rapidamente o significado de uma palavra. Basta selecioná-la, clicar com o botão direito do mouse e escolher a opção *Dictionary Search* no menu. Uma nova aba se abre, mostrando o significado da palavra. Na verdade a extensão é uma interface para o serviço gratuito de consulta disponibilizado pelo site dictionary.com

Colorzilla – Muito útil para desenvolvedores Web, essa extensão permite que você selecione e veja rapidamente os valores RGB para as cores de qualquer elemento de uma página Web. Também permite que você dê "zoom" na página e meça a distância entre dois pontos, entre outros recursos.

AdBlock – Essencial para quem detesta janelinhas e banners de propaganda com gorilas roxos e falsas "mensagens de erro" do Windows (como a clássica: "Seu computador tem um endereço IP! Isso é uma falha grave de segurança, clique aqui para corrigir"). Além de eliminar a chateação, sem os banners as páginas ficam mais leves e carregam mais rápido.

FireFTP – Um cliente FTP dentro do seu navegador web. Firefox e Mozilla já têm recursos básicos para acesso a servidores FTP, mas são limitados apenas ao download de arquivos, um por vez. Com o FireFTP você pode fazer download e upload de arquivos, continuar downloads interrompidos, criar "filas" de arquivos a ser transferidos ou verificar o status (tempo e quantidade de dados restantes) da transferência, entre vários outros recursos. A interface é similar à de clientes FTP populares como o GFTP (para o Gnome, no Linux), WS-FTP (para o Windows®) e Transmit (para o Mac OS X).

BandwidthTester – Extensãozinha que testa a velocidade de sua conexão à Internet, a chamada "largura de banda". Muito útil para diagnosticar uma aparente lentidão em sua conexão ou para saber se sua operadora de telefonia está realmente entregando o que prometeu no contrato de seu acesso via DSL ou se vende gato por lebre.

User Agent Switcher – Esta extensão muda a string que o navegador usa para se identificar junto ao servidor web. Alguns websites usam essa informação para decidir que versão da página mostrar ao usuário; outros, mal programados, bloqueiam o acesso de internautas que não usam o navegador para o qual o site foi "otimizado" (geralmente o Microsoft Internet Explorer). Com o User Agent Switcher você pode, por exemplo, fazer com que o Firefox se identifique como o Internet Explorer 6, contornando essa limitação.

Sage – Um agregador de notícias e "feeds" RSS, que pode ser usado para ver, em um único local e de forma centralizada, as últimas novidades em seus sites favoritos. Ele lê feeds RSS nas versões 0.9x, 1.0, 2.0 e feeds Atom, pode descobrir novos feeds sozinho, se integra aos bookmarks do Firefox, tem visualização de notícias personalizável através de folhas de estilo e suporta os idiomas catalão, chinês, holandês, francês, alemão, húngaro, italiano, japonês, coreano, russo, espanhol e sueco. Alguém se habilita a criar uma versão em português do Brasil?

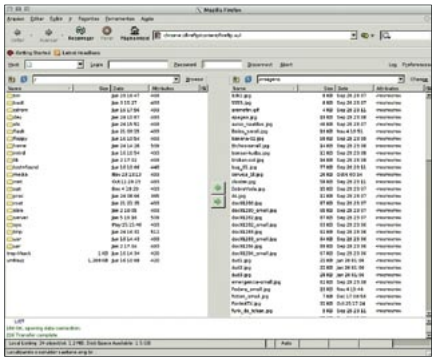


Figura 2: Faça transferências via FTP com o FireFTP.

Temas

Além da flexibilidade em termos de recursos, graças às extensões, o Firefox também é flexível no visual, com os temas. Eles são conjuntos de ícones, imagens e descrições de estilos de fontes e cores que são usados para mudar a cara do navegador, seja para integrá-lo melhor ao ambiente onde está sendo executado, seja por pura diversão.

A versão Mac do Firefox, por exemplo, vem com um tema padrão (Pinstripe) diferente das outras plataformas, que faz com que o navegador se integre muito melhor ao mundo "Aqua" do sistema operacional da Apple. Há temas que deixam o navegador com a cara do Internet Explorer (*Luna*, excelente para facilitar a migração de usuários inexperientes), ou que o integram perfeitamente ao Gnome (Gnome-Fx) ou KDE (Plastik). Uma lista com mais de 60 temas, divididos em várias categorias, está disponível no site Firefox Central.

Após escolher um tema, instalá-lo é fácil. Basta clicar no link *Install* logo

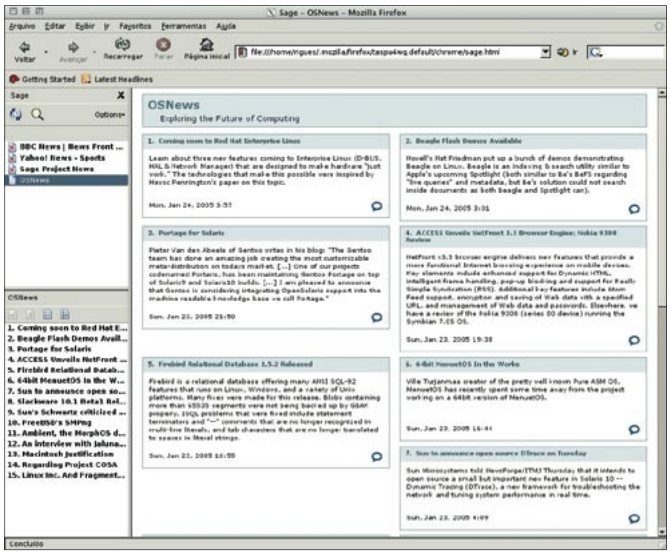


Figura 3: Instale o *Sage* e nunca mais fique por fora das novidades.

abaixo de sua descrição. Uma janela aparece, perguntando se você realmente deseja instalar o tema. Basta clicar em *OK* e esperar o download. A janelinha "Temas" irá surgir (se não, clique em *Ferramentas* | *Temas*), mostrando todos os temas já instalados e, em alguns minutos (dependendo da velocidade de sua conexão à Internet), o tema que você acabou de baixar.

Para usar um tema, dê dois cliques sobre o nome dele na lista (é necessário reiniciar o navegador para que o tema seja aplicado à interface). Para remover um tema, selecione-o na lista e clique no botão com um *X*, no rodapé da janela. Para atualizá-lo, clique no segundo botão, com as setinhas.

Crise de identidade

O navegador que hoje conhecemos como Mozilla Firefox passou por duas mudanças de nome em um passado não muito distante. Até a versão 0.5, o projeto era conhecido como Phoenix, nome do pássaro lendário que renascia das próprias cinzas, assim como o navegador renasceria das cinzas do Mozilla. Entretanto a Phoenix Technologies, fabricantes de BIOS para PCs e de um dispositivo de acesso à Internet, temeu uma confusão entre os produtos e solicitou a mudança no nome do navegador, no que foi atendida.

Em 14 de Abril de 2003 o navegador Phoenix e o cliente de e-mail Minotaur tiveram seus nomes mudados para Firebird e Thunderbird, respectivamente. Mas os problemas com nomes não acabaram por aí. Rapidamente a comunidade de usuários do banco de dados FirebirdSQL, descendente do Interbase, da Borland, reclamou que a coincidência entre os nomes estava causando confusão. A IBPhoenix, uma empresa que desenvolve produtos e serviços baseados no FirebirdSQL, incitou sua comunidade a inundar fóruns e listas de discussão do projeto Mozilla com reclamações contra o novo nome. Um dos responsáveis pelo projeto, Asa Dotzler, tentou uma solução conciliatória, sugerindo que o navegador fosse oficialmente chamado *Mozilla Firebird*, mas ela não foi aceita.

A saída foi mais uma mudança de nome: a partir de 09 de Fevereiro de 2004 o Firebird passou a ser conhecido como Firefox (Raposa de Fogo), um dos nomes para um animal também conhecido como "Panda Vermelho" [1] (a "raposinha" ao redor do globo no ícone do programa). Sem reclamações de qualquer espécie, o cliente de e-mail Thunderbird pôde manter seu nome.

Os usuários não perderam tempo para se divertir às custas da controvérsia. A extensão *Firesomething* combina duas listas, uma com nomes de animais e outra com elementos (água, terra, fogo, vento...) para criar um novo nome para o navegador a cada vez que ele é aberto. Hoje você usa o Firepanda, amanhã pode ser o Moontiger, na semana que vem o Earthbadger. Barras de título, janelas de preferências e a caixa de diálogo *Sobre...* (*About...*) são modificados. A identificação do navegador, usada por alguns sites para determinar que página exibir ao visitante, não é modificada, portanto a extensão não prejudica a compatibilidade do aplicativo.

Mostramos aqui apenas um pequeno conjunto dos recursos e vantagens do Firefox. Um programa em constante desenvolvimento, e com tantas possibilidades de expansão, com certeza esconde dezenas de outras agradáveis surpresas que só serão encontradas com a convivência diária. Experimente! ■

Figura 7: O *Panda Vermelho* não é uma raposa, mas é o mascote oficial do projeto Firefox.

INFORMAÇÕES

[1] Firefox Central: *http://www.mozilla.org/products/firefox/central.html*

[2] Chatzilla: *http://www.mozilla.org/projects/rt-messaging/chatzilla/*

[3] A verdadeira "Firefox": *http://en.wikipedia.org/wiki/Red_Panda*

[4] Thunderbird, a ave mitológica: *http://en.wikipedia.org/wiki/Thunderbird_%28mythology%29*

[5] Sage: *http://sage.mozdev.org*

[6] AdBlock: *http://adblock.mozdev.org*

[7] FireFTP: *http://fireftp.mozdev.org/*

SOBRE O AUTOR

Rafael Rigues, um maníaco por videogames e computadores antigos, já foi funcionário da Conectiva, onde participou do desenvolvimento do Conectiva Linux 5 e Conectiva Linux: E-Commerce, e membro da equipe da Revista do Linux, da qual foi editor durante um ano. Atualmente é um dos editores da Linux Magazine Brasil.

Figura 4: O Firefox é a cara do Gnome (*Gnome-FX*)...

Figura 5: ...do Windows XP (*Luna Blue*)...

Figura 6: ...ou do Mac OS X (*Pinstripe*).

É, sim, possível controlar o setor de atendimento técnico com Linux.

Liga pro suporte!

Ficar atento aos pedidos de suporte dos clientes pode fazer a diferença quando se trata de voltarem – ou não – a contratá-lo. Felizmente, diversos aplicativos baseados em Linux oferecem salvação para seu departamento de suporte técnico. POR JAMES MOHR

Quem quer que já tenha precisado ligar para uma empresa por causa de algum problema com um produto teve também de enfrentar o suporte técnico. Manter um registro dessas ligações não apenas ajuda a resolver problemas mais rápido como pode até mesmo evitar que alguns deles aconteçam.

A maior parte dos centros de suporte atualmente usa algum pacote de programas para gerenciar as ligações, tanto o suporte técnico ao cliente quanto aquele ao usuário interno.

Em ambos os casos é preciso manter um registro dos pedidos de suporte. Um pedido que chega é tipicamente registrado num "chamado técnico" (*ticket*). Portanto, para simplificar, vou me referir aos dois produtos como "sistemas de chamados técnicos". Neste artigo, conheceremos um pouco sobre esses sistemas e veremos alguns dos que estão disponíveis atualmente para Linux.

Curso rápido de sistemas de chamados técnicos

Sistemas de chamados técnicos podem ser simplesmente ferramentas que permitem registrar pedidos de suporte. Podem ser também ferramentas mais complexas que capacitam você a gerenciar problemas, por exemplo iniciando ordens de serviço diretamente a partir dos chamados. Saber diferenciar entre chamados técnicos e ordens de serviço é muito útil, seja ao lidar com clientes externos ou com funcionários de outros departamentos da empresa.

Nada impede que você use o mesmo sistema para gerenciar pedidos de suporte e chamados técnicos. Misturar os dois pode ser confuso, então é bom atribuí-los a diferentes grupos ou construir campos especiais que indiquem se o pedido é um chamado técnico normal ou uma ordem de serviço. Nem todos os produtos facilitam essa tarefa, porém.

Um aspecto fundamental do sistema de chamados técnicos é a capacidade de atribuir um único número (ou ID) a cada chamado. Embora esse recurso seja padrão nos sistemas, o modo como o ID é gerado não é sempre o mesmo – e nem sempre você ficará satisfeito.

Além disso, sistemas de chamados técnicos precisam de um modo de registrar o responsável pelo chamado. No mínimo, o sistema deve permitir a definição dos usuários a quem os chamados podem ser atribuídos. A maior parte dos sistemas vai além e permite criar grupos de usuários. A terminologia com freqüência é muito diferente, mas o objetivo é atribuir um chamado a um grupo de pessoas para que o próximo usuário disponível possa atendê-lo.

Em geral, sistemas de chamados técnicos compõem-se de três unidades principais: usuários, grupos e filas (embora a terminologia não seja consistente). A fila é o elemento administrativo central para o processamento de chamados técnicos. Por exemplo, você pode criar filas separadas para redes, impressora e problemas com aplicativos.

Será que funcionam?

Se você fizer uma busca no Google e no SourceForge, encontrará rapidamente uma dúzia de produtos diferentes que recaem na categoria "ticket systems". Se fôssemos falar de cada um deles, teríamos que nos restringir a um ou dois parágrafos para cada um. Vamos portanto limitar um pouco a busca.

O primeiro critério, obviamente, é que o programa precisa rodar em Linux. A seguir, as buscas foram limitadas a produtos que estão sendo desenvolvidos atualmente. "Atualmente", aqui, é um tanto arbitrário, mas não incluí de propósito produtos que não foram

recentemente atualizados. Alguns eram bonitinhos e tinham recursos interessantes, mas sabe-se lá para onde iam me levar depois de eu perder um tempão para instalá-los.

A escolha final dos aplicativos a incluir também foi algo arbitrária. Basicamente dei uma olhada na lista de recursos para ver o que eles ofereciam e escolhi as ferramentas cujos recursos me agradaram mais. *Não* me limitei a produtos não-comerciais ou open source.

As interfaces que vi eram exclusivamente baseadas no navegador web. Isso permite fácil acesso tanto aos clientes quanto ao pessoal do suporte. Porém, ao testar esses produtos, é preciso ver como ele reage a seu navegador padrão. Em alguns casos, descobri que a interface web tinha alguns defeitos em diferentes navegadores do Linux.

Em cada caso, o produto dava suporte à abertura de chamados técnicos via email, seja fazendo com que o servidor de email mandasse a mensagem para um filtro especial (a maioria dos servidores de email pode fazer isso) ou fazendo com que o sistema de chamados recolhesse ele próprio as mensagens usando POP3 (alguns fazem as duas coisas). Não importa se você está gerenciando seu próprio servidor ou usando um provedor de hospedagem, um desses métodos deve funcionar.

Outra característica que busquei foi a facilidade de ajustar o sistema às necessidades. Parte disso é a capacidade de restringir o acesso a diferentes partes do sistema. Por exemplo, dá para separar chamados técnicos em diferentes grupos, deixando aos usuários acesso apenas a chamados dentro de seu próprio grupo? Ou acesso de apenas leitura aos outros grupos? Dá para evitar que os usuários apaguem chamados? Que outros direitos ou privilégios podem ser definidos? É possível configurar a interface ou as respostas padrão para os clientes?

Ajustar o sistema a suas necessidades também pode significar incluir seus próprios tipos de dados, com a capacidade de acrescentar novos campos. Assim como para as permissões dos usuários, o modo como esse recurso é implementado varia de produto para produto. Pessoalmente, acho que essa característica é, com muita freqüência, subestimada.

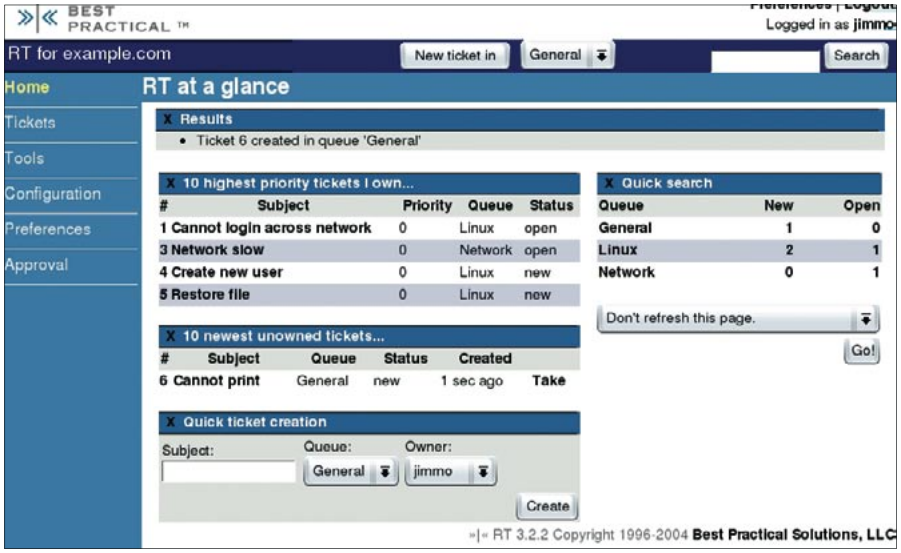


Figura 1: Request Tracker "num relance".

Como não conheço sua empresa e suas exigências específicas, não posso definir qual desses produtos você deveria usar. Todavia, vou deixar você dar uma olhada no que está disponível e em alguns dos recursos que cada produto tem a oferecer. Se você realmente quer tomar uma decisão informada, sugiro escolher uma ou duas dessas ferramentas e instalá-las em seu computador. Vale também visitar as demonstrações online disponíveis para alguns desses produtos.

Request Tracker

O Request Tracker [1] é um sistema de código aberto para gerenciamento de chamados técnicos da Best Practical Solutions. O processo de instalação do Request Tracker (RT) foi bastante incômodo. Além de exigir Perl 5.8.3, a versão que baixei necessitava de uma longa lista de módulos Perl, muitos dos quais eu nem sabia que existiam.

Forneceram um arquivo *make* que incluía recursos para testar/exibir as dependências, assim como para "consertá-las" automaticamente, aproveitando a capacidade do Perl de baixar e instalar sozinho os módulos do repositório online CPAN. Porém, ele não funcionou como esperado e tive de baixar alguns módulos manualmente.

Uma coisa chata é que, se o RT não encontrar a versão do Perl "exigida" ao rodar o *make fixdeps*, ele continua a baixar módulos do Perl para a versão que já está em seu sistema. De acordo com a Best Practical, isso será consertado na próxima versão.

Na primeira vez em que você se loga, aparece um tipo de "Introdução ao RT". Esse recurso mostra mal e porcamente o que está acontecendo. Nem todos os chamados abertos são vistos diretamente na primeira página, mas em geral estão a apenas um clique de distância.

Nessa página há também um bloco de nome "Criação rápida de chamados". Como o próprio nome diz, esse recurso permite criar chamados técnicos muito rápido. Aqui, só é possível escolher o assunto, o proprietário e a fila. Ao clicar em "create" (criar), o chamado é criado e você é levado de volta à página inicial, para criar mais chamados ou mexer nos que já existem.

Se você der uma olhada na figura 1, verá que cada função tem seu próprio cabeçalho com um *X* na extremidade esquerda. Ao clicar nesse *X* a caixa é "colapsada" e só o cabeçalho fica visível. Clique novamente no *X* pra voltar a abrir a caixa. Esse recurso aparece em praticamente todas as páginas relacionadas a chamados técnicos e pode ser muito útil se a página estiver muito atulhada de informações.

O RT tem uma sub-classe de usuários chamados "watchers" (vigias). Como o termo sugere, são as pessoas que vigiam o chamado. Podem ser as pessoas envolvidas ativamente, como o requerente e o técnico do suporte. Também podem ser as pessoas apenas interessadas no chamado. Por exemplo, se o proprietário da empresa abre um chamado técnico, o gerente do departamento de suporte pode ter interesse em seguir o processo.

Assim como com qualquer produto, os chamados técnicos são processados por pessoas com habilidades específicas. É de bom-tom enviar cada chamado apenas para a fila da pessoa designada para tratar dele. Assim, por padrão, uma fila também serve como a "categoria" do chamado. Porém, com o RT é possível criar campos personalizados, como "categoria", "classificação" ou qualquer coisa que os produtos comerciais possam oferecer.

Como unidade administrativa, grupos e filas podem tipicamente ser usados na atribuição de direitos. Pessoas específicas podem ter o direito de *modificar* chamados dentro de certas filas mas apenas *ler* chamados em outras. Alguns grupos podem cuidar de funções específicas. O RT faz um ótimo trabalho ao controlar os direitos de cada grupo e as funções de cada fila. Como os usuários são parte dos grupos, você está controlando indiretamente os privilégios dos usuários.

O RT leva o conceito de histórico de chamados dois passos além dos outros produtos. Primeiro, além de um histórico padrão, há um histórico de mudanças relacionadas ao chamado – por exemplo, quando é atribuído a uma pessoa diferente ou seu *status* muda. Em segundo lugar, é possível marcar itens não-administrativos do histórico como uma resposta ao requerente ou como "particular", de forma que o requerente não veja a resposta. Esse recurso é útil para repassar informações para outros representantes do suporte sobre o relativo conhecimento do requerente (ou falta dele).

As prioridades podem variar numa escala que vai de 0 a 99; o chamado pode receber maior ou menor prioridade conforme se aproxima o fim do prazo. Cem níveis de prioridade pode ser uma quantidade exagerada, mas é possível configurar o comportamento das prioridades para que se adequem à sua empresa.

Outro recurso muito útil é a capacidade de criar relacionamentos entre chamados técnicos. Pode ser apenas uma nota que diga "Ei, olhe aquele outro chamado ali". Ou você pode criar dependências reais. É possível até mesmo criar relações pai-filho.

Esse mecanismo permite usar o RT para criar um sistema de "ordens de serviço", no qual um problema é relatado e gera um segundo chamado técnico, que é atribuído à segunda fila, responsável por realizar o trabalho. Ou mesmo "projetos", em que uma tarefa grande é composta de diversas tarefas menores.

O RT acrescentou uma melhoria em sua comunicação com o consumidor, que é chamada *scrips* (sem o *t* mesmo). Scrips são notificações ao consumidor que reagem segundo critérios diversos. O exemplo clássico é o gerente do suporte receber uma mensagem no pager avisando que o dono da empresa abriu um novo chamado técnico.

A maior parte dos produtos tem a capacidade de buscar ligações específicas. Porém, a busca do RT merece menção especial por causa do detalhamento que se pode atribuir à busca.

OTRS – Open Ticket Request System

O Open Ticket Request System (OTRS) [2] é outro sistema Open Source de chamados técnicos. A instalação do Open Ticket Request System (OTRS) é excepcionalmente fácil se você tiver uma distribuição Linux que trabalhe com empacotamento RPM. Embora a instalação exigisse alguns pacotes que eu não tinha em meu sistema, estes foram instalados facilmente a partir dos CDs do SuSE. Após instalar o pacote RPM, usei um navegador web para o resto da configuração, que incluiu a criação e a adição dos usuários necessários.

O aplicativo suporta também distribuições Linux que não usam RPM e dialetos UNIX, mas você só tem um arquivo *.tar.gz* e precisa realizar todas as etapas da instalação "à mão". Isso inclui procurar todos os módulos de Perl necessários, criar e configurar sua base de dados e ajustar o servidor web. Essa instalação completa é útil se você quiser entender melhor como os componentes funcionam, mas é bem incômoda se esse não for o caso.

Embora os privilégios, usuários e grupos não possam ter ajustes tão finos como nos demais gerenciadores de chamados, há algumas opções de configuração para as filas que não estão disponíveis em outros produtos. A opção de sub-filas para organizar ainda melhor a atribuição de tarefas é uma delas. Há também como definir o tempo máximo de atendimento do chamado e um ajuste para limitar o tempo pelo qual os chamados podem ser trancados (o trancamento de chamados é um recurso que poucos produtos têm).

O OTRS pode ser configurado para permitir que os usuários criem suas próprias contas antes de solicitar a abertura de chamados (o que se chama "Customer Self Registration", Registro do Cliente). Funciona assim: o cliente dá um nome e endereço de email. Se o email não chegar, a empresa tem um registro de que o usuário entrou em contato com o suporte técnico. Os clientes também podem criar uma conta enviando um email.

Em sua maioria, os chamados técnicos ORTS são administrados como emails comuns. As mensagens são armazenadas em texto puro no disco rígido; os cabeçalhos ficam na base de dados. Esse formato permite procurar chamados mais rapidamente. Mesmo os anexos são tratados como no email comum.

A aparência dos chamados técnicos dentro de uma fila é semelhante à lista de emails em um cliente comum de correio eletrônico, com os tópicos sendo mostrados no padrão *From* (De), *To* (Para) e *Subject* (Assunto). Junto com esses tópicos há alguns valores específicos dos chamados, como estado/status e prioridade na fila atual. Você pode dar um "zoom" no chamado para ver os detalhes.

Um recuso elegante é uma caixa na página de chamados com o nome "modelos de resposta", que permite compor respostas padrão livremente configuráveis, como "Favor procurar no Google", "RTFM" e "Leia as FAQ". Coisas assim são extremamente úteis se você tiver uma alta porcentagem de chamados repetidos ou chamados que possam ser resolvidos com as respostas padrão. Modelos diferentes podem ser atribuídos a filas diversas se você quiser dar uma resposta variada de acordo com a fila. Também na página de chamados há o *CustomerID*, que é um link que leva você a todos os chamados abertos de um cliente específico.

O OTRS faz um bom trabalho ao registrar as mudanças no histórico. Muitas das mudanças administrativas (ou seja, a atribuição de um novo proprietário) permitem que uma nota seja acrescentada; a mudança é então acessível através do histórico. Por exemplo,

quando você re-atribui um chamado, pode adicionar uma nota sobre a razão para ele estar sendo re-atribuído.

O OTRS tem algumas coisas bem desagradáveis. Uma delas é que, embora haja alguns campos personalizáveis, eles são em número de quatro – muito poucos, eu diria. Também não é permitido encerrar um chamado, a menos que você seja o proprietário. Em minha experiência com outros produtos, todos os atendentes do suporte deveriam também poder fechar um chamado.

e-Support

O e-Support [3] é um sistema comercial de chamados técnicos da Kayako Web Solutions. Achei que a instalação do Request Tracker houvesse sido incômoda, mas a instalação do e-Support foi um verdadeiro espinho no pé. Embora houvesse poucos passos a realizar – não precisei baixar dezenas de módulos e grande parte da configuração foi feita através do navegador web – foi preciso *um monte* de clarividência para adivinhar onde enfiar as coisas e como iniciar a instalação.

Para um produto que exige dinheiro de verdade, a documentação de instalação é bastante rala, o que tornou a instalação muito mais difícil. A resposta às perguntas que fiz à empresa foi basicamente "Está na documentação" ou "está no site". Quando perguntei em que lugar do documento estava aquilo, não puderam me dar uma resposta, e nada achei sobre o tema no website.

Assim como nos outros produtos, há usuários e grupos (que são chamados 'departamentos') e os usuários só vêem chamados atribuídos aos grupos de que são membros. É possível mover chamados de um departamento para outro e esse movimento é registrado no histórico do ticket, junto com o nome do usuário. Porém, os detalhes não são automaticamente incluídos (por exemplo, "moveu-se do grupo A para o grupo B").

Um voto de indulgência para o e-Support é a interface muito fácil de usar. Isso se aplica não somente à interface com o usuário (figura 2), como também à administração. Embora não seja possível configurar tanto quanto em alguns produtos não-comerciais, muito da configuração é mais auto-explicativa. Porém, assim como na instalação, tive de exercer meus dons de adivinhação, assim como usar o velho método da "tentativa e erro".

Outro aspecto muito útil do e-Support é a "base de conhecimento". Assim como com outros produtos, ela fica mais perto das FAQ clássicas que de uma base de conhecimento, já que é uma resposta específica para uma questão dada. Os usuários podem estender a base de conhecimento adicionando comentários ou notas às entradas. Por padrão, todos os comentários são moderados, mas isso pode ser desabilitado, caso necessário.

Figura 2: A página inicial do e-Support

A interface da base de conhecimento facilita a navegação, a criação de novas categorias, a criação de perguntas e respostas e assim por diante. Além disso, você pode relacionar código HTML à resposta criando (por exemplo) links para respostas mais detalhadas. Você também pode ter links para "tópicos relacionados" (ou seja, outros artigos).

Há também um recurso de resolução de problemas. Assim como seu homônimo no Windows, a resolução de problemas guia você através de uma série de questões para ajudar a delimitar o problema. O processo leva tempo, mas se você tiver uma porção de probleminhas fáceis de resolver, esse recurso pode ser bastante útil.

O produto tem também alguns recursos de relatório. Você pode criar um relatório sobre a atividade de cada usuário. Pode também gerar "estatísticas" para um dado "cliente". O relatório de estatísticas lista os chamados abertos com informações sobre cada um deles, como o assunto, a data de abertura e assim por diante.

A utilidade dos relatórios é bastante reduzida pelo fato de que o "cliente" é a primeira pessoa a abrir qualquer chamado para qualquer endereço de email determinado. Abri com sucesso inúmeros chamados para um único endereço, mas com nomes diferentes. O programa não reclamou de nada disso (o que dirá me avisar...). Ao gerar as estatísticas, apenas o primeiro nome usado foi exibido, de forma que reunir as estatísticas de uma só pessoa não é lá muito fácil.

Um recurso interessante é o chamado *InstaSearch*. Ele busca na base de conhecimento por respostas *possíveis* a um problema. Essas respostas podem ser enviadas junto com emails de resposta automática, bem como quando o usuário abre um chamado. Infelizmente, segundo minha própria experiência somada ao que pude descobrir nos fóruns do e-Support, pode ser que haja tantos itens que o botão "submit" (enviar) do chamado técnico acabe desaparecendo da tela. Como resultado, a pessoa pensa que o chamado já foi aberto, quando na verdade nunca foi.

perlDesk

O sistema comercial perlDesk [4] é muito simples e fácil de usar. Isso também quer dizer que ele não possui muitos dos recursos oferecidos pelas outras ferramentas, inclusive as de código aberto. Porém, o perlDesk oferece uma cara muito profissional. Enquanto certos produtos deixam óbvio que você está trabalhando com um navegador web, o perlDesk dá a impressão de um aplicativo "real".

Assim como com o e-Support, a falta de documentação num produto comercial é desapontadora. As instruções de instalação do perlDesk eram ligeiramente melhores, mas não mais que isso. Porém, o suporte oferecido pelo desenvolvedor do perlDesk (a logicNow Limited) foi exemplar; eles definitivamente se esforçaram para assegurar que as coisas funcionassem (em chocante contraste com o outro produto comercial).

Os usuários podem abrir chamados via email ou pela interface web, que também fornece um mecanismo de auto-registro chamado "Quick User Signup," semelhante ao OTRS.

Um recurso realmente útil é a capacidade de acrescentar um chamado diretamente à base de conhecimento (FAQ). Ao clicar no botão enviar, você é levado para um novo formulário, onde pode escolher a categoria da FAQ bem como editar o texto, o que não afeta o texto do chamado técnico original.

As FAQ funcionam basicamente do mesmo modo que nos outros produtos, com um conjunto de respostas fixas para problemas conhecidos. Isso é apoiado pelo componente "resolução de problemas", que também funciona de maneira igual à dos outros produtos. É muito fácil adicionar novos itens à resolução de problemas.

Um belo recurso é a capacidade de fazer um backup da base de dados de dentro do próprio produto. Se o servidor estiver rodando numa máquina próxima a você, pode não haver a necessidade de usar esse recurso. Porém, ele é muito útil se você estiver usando um provedor de serviços de hospedagem e esse for o único modo de salvar seus dados.

Outro recurso elegante é o *Staff Rating*. Quando ele está habilitado, os usuários podem avaliar as respostas da equipe de suporte (numa escala de 1 a 5). O usuário também pode acrescentar comentários à avaliação. Para mim, isso é mais útil para determinar a efetividade da equipe do que simplesmente olhar para o número de chamados fechados por cada atendente.

Apenas por preferência pessoal, não gostei do fato de a administração ser completamente separada da equipe. Para administrar o sistema, você precisa de um login completamente diferente. Embora isso ajude a separar as funções, pode ser um aborrecimento se você precisar fazer uma mudança no meio do processamento de um chamado.

Também desapontou a (pouca) capacidade do perlDesk na busca por chamados específicos. Você pode procurar por todos os chamados abertos (não-resolvidos) ou fechados (resolvidos), assim como exibir os chamados abertos por prioridade, departamento e membro da equipe. Porém, isso é praticamente tudo o que ele pode fazer.

Um recurso do perlDesk que não estava imediatamente disponível em muitos outros produtos é a capacidade de mudar o desenho das páginas com um editor de templates. Isso é feito simplesmente com um *textarea* num formulário HTML, mas você pode configurar com facilidade praticamente todas as páginas do sistema.

OSTicket

O sistema de código aberto OSTicket [5] foi sem dúvida o mais fácil de instalar. Não precisei baixar quaisquer outros pacotes ou arquivos de outros sites. Não precisei fazer uma configuração detalhada em meu servidor web. O aplicativo estava funcionando em menos de cinco minutos. O único ponto no processo de instalação a que eu chamaria de, no mínimo, incômodo foi o fato de que tive de criar a base de dados à mão. Porém, uma vez que isso também era requerido para os produtos comerciais, posso facilmente fazer vista grossa a esse respeito.

O produto é um sistema de chamados técnicos enxuto, mas efetivo, com poucos recursos. Descompactado, o pacote ocupa apenas 300 KB (!) no disco rígido; por conseguinte, é óbvio que não se pode esperar o mesmo número de recursos de um produto 10 vezes maior. Porém, dependendo de suas necessidades, esse pequeno tamanho pode ser uma vantagem.

Note que os 300 KB não incluem a documentação, já que quase nenhuma é fornecida. Em vez disso, você é enviado ao fórum do OSTicket quando precisa de ajuda. Os desenvolvedores dizem que isso será melhorado na próxima versão. Mesmo assim, a interface não tem muitos meandros.

Afora gerenciar chamados técnicos, esse produto não faz grande coisa. É possível criar grupos e usuários (chamados representantes) e limitar a categoria de chamados que o usuário pode acessar, assim como os aspectos do sistema que podem ser configurados pelos usuários (embora não haja muitos). Mas aqui os recursos definham lentamente.

Os chamados técnicos são agrupados em categorias, que em seguida são atribuídas a grupos específicos. Os membros do grupo têm acesso, então, ao chamado nas categorias. Os usuários são limitados a um único grupo.

Um aspecto realmente desagradável do OSTicket é que, não importa por onde comece, você é invariavelmente levado de volta à página principal. Por exemplo, ao criar um novo usuário, o clique no botão *Create* não leva à página do novo usuário, mas à página principal. Isso é especialmente irritante quando, por exemplo, você esqueceu de completar um campo requerido. Claro que você pode simplesmente pressionar o botão *Voltar* do navegador, mas é altamente provável que prefira corrigir o erro voltando à página.

Uma coisa que realmente me impressionou ao fazer as pesquisas para o artigo foi o número de sites que oferecem hospedagem web para sistemas baseados no OSTicket. Isso demonstra a facilidade com que o OSTicket pode ser instalado e administrado.

Ocasionalmente, irritei-me com o número limitado de recursos oferecido pelo OSTicket. Porém, se tudo o que você quer é um modo simples e fácil de manter um registro de qualquer tipo de pedido, o OSTicket realmente merece uma chance. ■

INFORMAÇÕES

[1] Request Tracker: *http://www.bestpractical.com/rt/*

[2] Open Ticket Request System: *http://otrs.org/*

[3] e-Support: *http://www.kayako.com/?_a=products&_m=esupport.features#*

[4] perlDesk: *http://www.perldesk.com/index.html*

[5] OSTicket: *http://www.osticket.com/*

Editoração mais profissional com o Scribus

Tipografia e texto

Na segunda desta série em três capítulos [1], Jason Walsh trata de alguns detalhes da confecção de um jornal no Scribus. Veremos também como administrar aqueles arquivos gráficos chatinhos com esquema de cores CMYK. POR JASON WALSH

www.photocase.de

No último mês, mostrei como criar um projeto de editoração eletrônica no Scribus e ensinei os primeiros passos para esboçar nosso jornalzinho de exemplo com a importação e inserção do nome do jornal no alto da primeira página. Também demos uma olhada na forma de colocar o texto numa caixa de texto. Este mês, veremos mais detalhadamente como inserir, ajustar, encaixar e destacar o texto. Mas, em primeiro lugar, vamos falar da tarefa de colocar a fotografia principal na primeira página.

Importando uma imagem

A imagem principal é uma foto ou outra imagem gráfica que acompanha um artigo e é colocada próximo ao alto da página. O arquivo de imagem é criado separadamente e em seguida importado para o Scribus. Começaremos colocando uma imagem principal na primeira página de nosso jornal de Linux.

A primeira complicação com que deparamos ao importar uma imagem gráfica para o Scribus é a falta geral de suporte para espaços de cor CMYK no Linux. Como você aprendeu mês passado, a composição em CMYK é fundamental na editoração eletrônica profissional. Através desse processo, imagens coloridas podem ser quebradas em seus tons compostos de ciano, magenta, amarelo e preto. Embora a gama de cores que pode ser obtida na impressão em CMYK seja menor que a gama de vermelho, verde e azul de um monitor, mais tons podem ser reproduzidos com sucesso.

O principal editor de bitmap para Linux é o Gimp, que é efetivamente um clone do Adobe Photoshop 3. Porém, diferente do Photoshop, até ha bem pouco tempo o Gimp não dava suporte a espaços de cor CMYK. Felizmente o Gimp 2.0 já pode trabalhar em CMYK com o plugin *gimp-cmyk* (veja o quadro "Instalação do plugin *gimp-cmyk*"). Ele permite ao usuário converter uma imagem RGB para camadas CMYK individuais (usando Perfis de Cor de origem e destino especificados) e em seguida salvar essa coleção de camadas como um TIFF CMYK.

Figura 1: Editando o estilo das linhas.

Para importar uma imagem para o Scribus, selecione primeiro *New Image Frame* na barra de ferramentas, depois clique e arraste a moldura retangular até que esteja mais ou menos do tamanho desejado.

Em seguida importe a imagem escolhendo *File > Import > Get Picture*.

Finalizando a Imagem

Após importar a foto, posicione-a exatamente no local desejado e altere o tamanho dela caso necessário.

Cada publicação tem seu próprio estilo, o que é normalmente refletido nos guias de estilo. O East Belfast Observer exige uma linha preta, com dois pontos de largura, em torno de cada imagem. Isso não apenas cria um visual classudo como também serve para fins técnicos. Muitas impressoras de jornal "saem do registro" – isso significa que as pranchas de quatro cores usadas na impressão se desalinham e a imagem acaba saindo borrada, quase sempre fugindo dos seus limites. Uma linha de dois pontos tem duas vezes a largura de uma linha comum – seria muito incomum que uma das pranchas de cor se desalinhasse em mais de dois pontos. Portanto, mesmo se a impressora sair do registro, a linha preta ajudará a deixar a página um pouco mais limpa.

Como todas as fotografias neste nosso projeto apresentarão essa linha, é uma boa idéia criar um estilo que possa ser aplicado a qualquer imagem sem ter de recriá-la manualmente todas as vezes.

Para criar um estilo, selecione *Edit>Line Styles*. Na caixa de diálogo que aparece, selecione *New*. Surge uma nova janela, na qual definiremos o estilo de nossa linha (figura 1).

No alto da janela, digite um título para o estilo, como *linha2pt*. Na janela há também opções para ajustar as

Resolução

Imagens de tela, como vemos na Internet, estão tradicionalmente em resolução de 72 pontos por polegada – isso significa que cada polegada da imagem é composta por 72 pontos. Se você tentar imprimir imagens da web, elas ficarão granulosas e falhadas, já que essa resolução é demasiado baixa para impressão.

Ao imprimir em casa, a maioria das pessoas cria imagens com resolução de 300 dpi. Essa é uma resolução bastante segura, alta o suficiente para ser usada na maioria dos casos – na verdade, é desnecessariamente alta para um jornal.

Para perceber qual resolução usar para nossas imagens, é importante compreender o que está acontecendo com elas. Imagens impressas consistem de pontos em meios-tons (se você escanear uma página de jornal ou revista, verá esse padrão de meios-tons - ver figura 5).

Figura 5: Uma imagem publicitária do comediante Bill Bailey escaneada de um jornal irlandês. Note os pontos de meio-tom visíveis, formando um padrão na imagem.

Meios-tons não são medidos em pontos por polegada (dpi – dots per inch), mas em linhas por polegada (lpi – lines per inch). Esse processo é chamado screening. Quanto mais alto o valor de lpi, mais pontos de meio-tom são colocados por polegada; o resultado final é uma imagem de resolução maior. Jornais são impressos com 65-85 lpi, revistas com 100-150 lpi e livros entre 150-300 lpi.

Cada ponto de meio-tom é composto de pequenos pontinhos medidos em dpi. O dpi efetivo é o número máximo desses pontos que a impressora pode imprimir em uma polegada.

A razão de dpi para lpi fica normalmente entre 150 e 200%; por conseguinte, se sua tela meio-tom tem 85 lpi, suas imagens precisam estar numa resolução mínima entre 127,5 e 170 dpi.

A resolução relativamente baixa da tela de meio-tom, ou seja, 85 lpi, é adequada para jornais porque a tinta vai se esparramar pelo papel de baixa qualidade usado em sua impressão.

Claro que não dói deixar uma resolução levemente maior do que o necessário, de forma que todas as nossas imagens estejam em segurança com 200 dpi. Embora essa resolução seja alta o bastante para um jornal e permita que as imagens sejam ligeiramente ampliadas, se as imagens em 200 dpi forem impressas numa revista com um papel de maior qualidade o resultado não será muito satisfatório.

extremidades das linhas e como elas se unem. Para nosso projeto, deixe essas configurações no padrão: *Flat Cap* e *Miter Join*.

A seguir, ajuste a largura da linha para 2 pt e a cor para preto. Finalmente, clique OK e o estilo estará definido.

Para aplicá-lo, clique com o botão direito na imagem e chame a janela de propriedades com a opção *Show Properties*. Escolha *Line* (Linha) e selecione o estilo recém-criado da lista na parte de baixo da janela.

Esse processo de definir um estilo e aplicá-lo a uma imagem ou caixa de texto é a chave para a editoração profissional, uma vez que permite que o usuário crie layouts consistentes rapidamente e com a segurança de que serão uniformes em todo o documento.

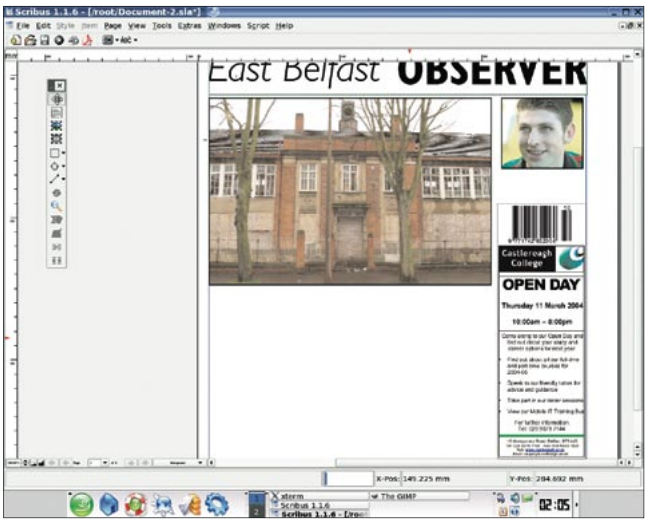


Figura 2: A primeira página até agora, sem o texto.

Limpeza da Página

O mesmo processo é repetido para a pequena coluna da direita: desenhe uma moldura, importe uma ilustração, ajuste o tamanho e aplique o estilo.

Uma pequena legenda relativa à fotografia virá sob essa imagem. Vamos ignorar essa legenda neste momento, mas não deixemos de reservar um espaço para ela. Abaixo da legenda vai o código de barras ISSN e uma propaganda.

Os anúncios são imagens criadas num programa de editoração e exportadas como arquivos EPS ("Encapsulated PostScript" – PostScript encapsulado). São importados da mesma maneira que as fotos: desenhe uma moldura e selecione *Import > Get Picture*. Diferente das fotografias editoriais, os anúncios não têm uma linha de dois pontos em torno deles. Em vez disso, trazem uma moldura de um ponto, que já é parte integrante da imagem mesmo antes de ela ser importada.

O código de barras é exatamente a mesma coisa: outro arquivo EPS, embora dessa vez seja criado em um programa especializado. Nenhuma moldura é necessária.

Instalação do plugin gimp-cmyk

Antes de instalar o plugin, verifique se o seu Gimp já não o possui. Muitas distribuições do Linux incluem o gimp-cmyk como parte de sua instalação padrão do Gimp. Por exemplo, esta série foi produzida usando o SuSE Linux 9.1, que traz o "gimp-cmyk" instalado como padrão. O mesmo se pode dizer do Debian Sarge, que possui o Gimp 2.2 já com o plugin para CMYK.

Uma versão binária do plugin gimp-cmyk está disponível, compilada para SuSE Linux 8.0, no site de desenvolvimento: http://www.blackfiveservices.co.uk/separate.shtml

Se você usar uma distribuição Linux diferente, ou quiser compilar no SuSE sua própria versão do plugin, precisará ter instaladas as bibliotecas Gtk, Gimp, LibTIFF e LCMS. Se já as tiver instaladas, basta:

```
make clean
make
```

Se o gimp-cmyk estiver instalado em seu sistema, será possível converter uma imagem RGB para o formato CMYK clicando com o botão direito na foto e selecionando "Imagem" no menu.

Um plugin do Gimp para CMYK é útil para nossos fins, mas CMYK numa rotina de trabalho não-colorida não seria o ideal. Num futuro não tão distante, os desenvolvedores do Gimp planejam passar a uma biblioteca gráfica mais genérica chamada GEGL. Essa manobra permitiria o suporte nativo a CMYK.

Rumo ao Texto

Agora que nossa imagem principal está em seu lugar, vamos voltar a atenção para o texto. O primeiro passo é selecionar um estilo de tipos. Muitas fontes foram projetadas especificamente para jornais. Para editoração profissional de jornais, vale a pena fazer um pouco de pesquisa sobre tipos.

No caso do East Belfast Observer, o texto do corpo (ou seja, o texto das notícias) é feito numa fonte chamada News 701. A News é uma fonte serifada, legível até mesmo em tamanhos pequenos – perfeita para jornais. Como não tenho uma licença para usar essa fonte neste projeto, usarei outra fonte serifada, a Bauer Bodoni. Qualquer fonte serifada bem projetada, como a Times, seria adequada para o texto principal, mas fontes como a News têm vantagens espaciais por serem desenvolvidas especificamente para uso num jornal.

Na verdade, o segredo para um bom uso de fontes é simples: escolha tão poucas fontes quanto possível e atenha-se a elas. Isso oferecerá ao leitor clareza e consistência.

Em nosso caso, todos os cabeçalhos são variações em negrito da fonte Gill Sans. No caso das notícias em si, é melhor importá-las de arquivos de texto; quanto aos cabeçalhos, basta desenhar uma caixa de texto, digitar o necessário e ajustar o estilo.

Assim como com os outros elementos no projeto da página, vamos criar um estilo para o texto principal – o das notícias – e aplicá-lo a elas. Escolha *Edit > Paragraph Styles* e clique *New* na janela *Edit Style*. Chame esse novo estilo de *Body Text News* (figura 3).

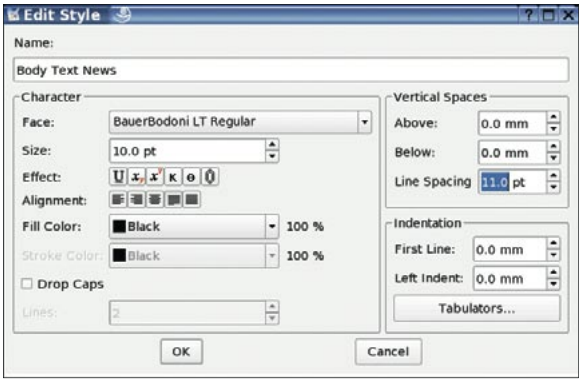


Figure 3: A caixa de diálogo usada para configurar o estilo do texto.

As opções que estamos usando são as seguintes: A fonte é Bauer Bodoni Regular com o tamanho de 10 pt. O tamanho que você dá a seu tipo depende da medida individual da fonte. Por exemplo, uma fonte relativamente grande como a News 701 seria ajustada para 8 pt, enquanto com a Times o tamanho pode chegar a 12 pt.

O espaçamento entre linhas (*Line spacing/leading*) é configurado para 110% do tamanho da fonte – ou seja, se a fonte tem 10 pt, use 11 pt, com uma configuração de fonte de 8 pt use um espaçamento de 8,8 pt e assim por diante. Isso é muito menos que a configuração padrão e dará ao texto uma cara densa e séria.

Grande parte do resto das configurações é questão de opinião. Neste caso, tomamos a decisão de "justificar" o texto das notícias (o texto forma uma linha reta nos dois lados da coluna). Repetimos, isso contribui para o ar de seriedade do texto. Evite centralizar, alinhar à direita e posições forçadas a qualquer custo: essas configurações

não ajudam em nada no texto principal.

O resto das configurações pode ser mantido nos valores padrão. Clique OK para voltar à janela *Edit Style* (Editar Estilos). Marque *Body Text News* e selecione *Duplicate*. Mude o nome para *Body Text Comment* e o alinhamento para *Align Text Left*. Clique OK. Assim como notícias factuais se beneficiam do texto justificado que as faz parecer mais densas, artigos de comentário, opinião e entretenimento ganham com o alinhamento à esquerda, pois a margem direita desigual os faz parecer mais leves. Feche a janela *Edit Style*.

Importando e ajustando o texto

Olhe um jornal com atenção; você vai notar que, diferente de muitas revistas, as notícias cabem direitinho no lugar destinado a elas, sem espaço em branco sob a última linha. Isso ajuda a dar um ar sóbrio e profissional às páginas de notícias e, obviamente, queremos reproduzir isso em nosso teste. Ajustar sutilmente o kerning e o tracking ajuda um pouco, mas o único modo de fazer isso direito é editar ou "remendar" o texto e fazê-lo caber. Essa é uma das principais razões por que as páginas são projetadas por sub-editores e não pelos diagramadores: a compreensão das regras da língua e a forma como fazer um texto fluir exige a travessia de um longo caminho.

Vamos remendar nossa notícia da primeira página. Desenhe uma caixa de texto sob o cabeçalho com exatamente a mesma largura, mas alta o bastante para chegar quase embaixo – digamos, cerca de 1,5 cm acima da margem.

Embaixo, desenhe outra moldura similar que encoste na margem inferior. Essa moldura conterá informação sobre o que mais há no jornal e determinará até onde nossa notícia principal pode chegar.

Importe o texto normalmente (Selecione a moldura de texto e vá em *File > Import > Get Text*). A seguir, aplique o estilo do Texto Principal em *Tools > Properties*. Na caixa de diálogo *Properties*, selecione *text* e escolha o estilo apropriado (*Body Text News*). A seguir, vá em *Shape* e selecione 3 colunas. Lembre-se de deixar um espaço entre as colunas, digamos, de 2 mm. Isso é feito editando-se o valor das *Columns* na seção *Shape* (Forma) da caixa *Properties*.

Feche a caixa Properties. Qualquer arquivo de texto razoavelmente grande serve – neste caso, estamos usando uma cópia das primeiras centenas de palavras deste artigo, já que não temos acesso à notícia verdadeira.

Agora a melhor parte (ou a pior – dependendo de sua opinião). Edite manualmente o texto para que se encaixe no espaço dado, como se você fosse um editor.

Retoques Finais

Tudo o que falta agora é finalizar a página. Na figura 4, você pode ver os elementos restantes que precisam ser adicionados.

Em primeiro lugar, acrescente o cabeçalho da notícia secundária no lado direito da página. A fonte do cabeçalho é Gill Sans Bold; ajuste o tamanho.

A seguir, abaixo da notícia principal, acrescente, ainda em Gill Sans Bold, os detalhes do conteúdo do resto do jornal. Lembre-se que uma moldura já foi criada para isso.

Depois, abaixo do nome mas acima da imagem principal está um destaque da seção de variedades. Esse destaque consiste de uma caixa de imagem e uma caixa de texto.

A última tarefa é finalizar os três destaques principais no alto da página.

Destaques

No mês passado escrevemos "The new newspaper for East Belfast" no alto da caixa à extrema esquerda. Vamos repetir isso agora e iniciar o processo de adicionar texto e imagem às duas outras caixas.

Trace uma caixa de texto sobre o retângulo da extrema esquerda e digite as palavras: "O novo jornal do East Belfast". Em seguida, ajuste a fonte, o tamanho e o espaçamento. Decidimos por valores de texto de 42 pt, na fonte Gill Sans Italic, com um espaçamento

Definições de Tipo

O que esses termos significam?

Serifa e Sem Serifa

Embora haja muitas subcategorias, a divisão básica das fontes é entre serifadas e não-serifadas. *Serifas* são os "tracinhos" nas extremidades das letras, que na verdade representam as marcas de cinzel dos antigos artesãos romanos.

Fontes serifadas são consideradas mais elegantes, suaves, sérias e legíveis, sendo por isso comumente usadas para grandes blocos de texto – confira num romance ou num jornal.

Uma fonte sem serifa (ou, menos comumente, Grotesca) não apresenta esses detalhes. Sua aparência modernista dinâmica as torna adequadas para textos de tamanho maior, como títulos, textos em pôsteres e livros de arte inexoravelmente "moderninhos".

Entrelinhas e Espaçamento

O espaçamento entre linhas, como é conhecido no Scribus, é chamado, em inglês, de *leading* – "lead" significa "chumbo" e traz referência às tiras de chumbo usadas para criar espaço entre as linhas no tempo da tipografia mecânica. Em português, o nome é "Entrelinhas".

As entrelinhas permitem ao diagramador alterar a densidade dos blocos de texto. Jornais são bastante densos e usam um valor de espaçamento menor do que o usado em revistas, panfletos ou pôsteres.

Figura 6: Fontes serifadas e não serifadas.

Kerning e Tracking (Entreletras)

Quando uma fonte é projetada, o projetista atribui a cada caractere uma largura, permitindo que caracteres consecutivos sejam colocados numa linha sem se tocar. Porém, as intenções do projetista das fontes podem não ser iguais às suas. O Kerning permite que você ajuste manualmente o espaço entre dois caracteres quaisquer. O Tracking (Entreletras) permite que o usuário aplique uma forma de espaçamento universal entre todos os caracteres. Dessa forma, é uma opção importante e, logo que você encontre uma configuração que julgue adequada, eu o aconselho a deixá-la em paz e ajustar o kerning manualmente para mudanças menores.

de 36pt. Posicione a caixa de texto de forma que fique dentro da caixa azul.

A seguir, desenhe duas caixas de imagem, uma sobre cada uma das caixas restantes. Dentro delas, importaremos fotografias, mas antes precisamos recortar as imagens (com o recurso *path*). Isso significa traçar uma linha em torno do aspecto da imagem que se quer usar, permitindo que o Scribus descarte o resto da imagem.

Abra as figuras em seu editor de imagens. Seja ele PixelFX!, PhotoPaint ou GIMP no Linux, ou Photoshop no Mac/Windows, o processo é similar. Explicaremos como fazer isso no próximo artigo .

Em nosso caso usaremos uma foto publicitária da atriz francesa Audrey Tatou e uma imagem PR em alta resolução de um iPod, baixada do site da Apple.

Pare, repita, repita novamente.

Agora que já quase acabamos de criar a primeira página, o processo deve ser repetido para as outras páginas. A contracapa fica ao lado da primeira página e, na tradição dos jornais anglófonos, é a página de esportes. As outras duas páginas mostradas são a capa da seção de variedades e uma página de notícias. Se você vem seguindo este tutorial, será capaz de criar páginas similares com bastante facilidade.

No próximo mês

No capítulo final da série, daremos os retoques finais na página, explicaremos como usar modelos para criar um documentos de muitas páginas e daremos uma olhada mais de perto na forma de levar as páginas ao prelo. Também pediremos que um profissional de pré-impressão avalie o jornal pronto. ■


The new newspaper for East Belfast Launch Issue

Football legend Derek Dougan Interview >>42

Win an iPod digital music player with Memsis.com >>27

East Belfast OBSERVER

Thursday 4 March 2004 No. 1 50p

INSIDE: theWEEK Full thirteen page Entertainment Guide with Film, Music & TV Listings >> Page 17

Leeman Calls on fans to lend a hand >> Page 48

INSTITUTE TO REOPEN

By Alex Crumlin and Josephine Long

The Belfast Institute of Further and Higher Education (BIFHE) is set for a shock return to East Belfast, after what many see as 'years of neglect'.

Institute bosses are set to announce a major new initiative for the East that will include significant new investment in state-of-the-art facilities.

The move comes after the college was presented with a report by education consultant Roger Mullin. His findings make sensational reading as it identifies a huge education problem in East Belfast.

With 60, 000 people living in this part of the City the report claims that almost 9000 people in the East have literacy and numeracy problems.

Controversially the report also found that whilst 35% of Catholics have no qualifications, 42% of Protestants find themselves in the same situation.

The report's author, Roger Mullin told the East Belfast Observer that whilst he had been told by many people in East Belfast that educational under-achievement was a major problem, he had assumed this was only a perception. However, he now admits that his report's findings shows this to be the case.

The hard-hitting report does not pull any punches.

>>3

SPORT 40 CROSSWORD 23 BUSINESS 14 ENTERTAINMENT 17

Castlereagh College

OPEN DAY

Thursday 11 March 2004

10:00am – 8:00pm

Come along to our Open Day and find out about your study and career options for next year.

Find out about all our full-time and part-time courses for 2004-05

Speak to our friendly tutors for advice and guidance

Take part in our taster sessions

View our Mobile IT Training Bus

For further information, Tel: 028 9079 7144

Figura 4: Primeira página pronta.

Tipografia avançada

A tipografia é uma arte, algo que os candidatos a doutor passaram seus mestrados esforçando-se para compreender. Nossa simples introdução não pode ir muito além de uma visão geral da tipografia; assim, se você quiser aprender mais sobre tipos mas nem sonha ingressar numa faculdade de artes, alguns bons livros serão uma mão na roda:

Stop Stealing Sheep and Find Out How Type Works. Por Erik Spiekermann.

O guia clássico da Adobe Press para entender a tipografia. Esse livro disseca o assunto do início ao fim, oferecendo uma rica base para o uso de fontes e tipografia.

The Mac Is Not a Typewriter. Por Robin Williams.

Obviamente destinado a usuários do Macintosh, esse livro é ainda uma prática referência para os interessados em editoração de páginas em qualquer plataforma. Tópicos tratados: aspas inglesas versus aspas genéricas, travessões *en* ou *em*, tabulações e recuos, kerning, entrelinhas, espaço em branco, viúvas, órfãs e pontuação sobrando.

Designer's Handbook. Por Alastair Campbell.

Este simples manual já é meio antigo, mas ajuda a compreender todo o jargão que os impressores cospem em você, desde PANTONE até sangramento, trapping e muito mais.

INFORMAÇÕES

[1] Para a primeira parte da série "Layout no Linux" consultar: Jason Walsh: "Um Jornal Via Linux", Linux Magazine Brasil, edição 5, pág. 32.

[2] Scribus: *http://www.scribus.org.uk/*

[3] News Page Designer: *http://www.newspagedesigner.com/*

[4] Society for News Design: *http://www.snd.org/*

[5] News Today: *http://newstoday.com/*

SOBRE O AUTOR

Jason Walsh foi diretor de arte do East Belfast Observer *desde o lançamento, em janeiro de 2004, até julho de 2004. Antes disso, era diretor de arte das elegantes revistas britânicas* Gorgeous *e* CityCraic. *Atualmente trabalha como jornalista e contribuiu para a* Linux Magazine, *a* Variant, *a* Mute, *o jornal* The Guardian *e muitas outras publicações sobre arte, design e tecnologia.*

Quem disse que o chaveiro é só para as chaves do carro?

Linux é a chave!

Todo mundo já está careca de saber que dá pra iniciar o Linux em um computador a partir de um *Live CD*. Mas quantos de vocês já pensaram em "enfiar" o pingüim em um chaveiro USB? POR FABRIZIO CIACCHI

Um "Live CD" como o Knoppix [1] ou o Kanotix [2] oferece ao usuário um sistema operacional extremamente portátil. Você pode carregar seu sistema pessoal para onde bem entender e iniciá-lo em qualquer computador que queira. Suas ferramentas, seus arquivos e sua área de trabalho o seguirão para onde você for – mesmo que o PC do seu pai ou de sua namorada possua outro sistema operacional. Os Live CDs também são usados por administradores de sistema para consertar aquele computador que teima em não iniciar normalmente.

O que poderia ser, então, mais portátil que um Live CD? Hmmm... Que tal um chaveiro de memória USB (os chamados Memory Sticks) com uma distribuição Linux? Um dispositivo desses é muito mais portátil, na maioria das vezes mais durável e certamente mais bacana que um CD. Além disso, trabalhar com memória USB é muito mais fácil do que com um CD: os dados são gravados e lidos mais rapidamente – instantaneamente é a palavra – e podemos dispensar o irritante e incômodo trabalho necessário de gerar uma imagem de CD e depois "queimá-la" na mídia virgem. Com alguns truques e um pouquinho de suor, podemos configurar uma distribuição Linux para ser iniciada em um dispositivo USB. Tudo o que precisamos é do chaveiro em questão e uma distribuição pequena o bastante para caber no espaço disponível e completa o bastante para reconhecer dispositivos de armazenamento USB. Obviamente, isso só funciona se a BIOS do computador permitir o boot via USB, o que não é incomum em hardware moderno.

Primeiro passo: obtendo um Linux "magro"

Os chaveiros USB são vendidos com diversas capacidades de armazenamento. Alguns chegam a comportar até 1 GB de dados, o que impõe poucas limitações para o tamanho do sistema a ser instalado. Entretanto, dispositivos menores são mais baratos e bem mais comuns. Parte de nosso objetivo é criar um sistema de baixo custo, portanto nossa discussão será baseada no modelo mais comum de 128 MB – quase um padrão. As instruções são válidas para qualquer tipo de Memory Stick, o que quer dizer que é possível colocar um Knoppix inteirinho em um chaveiro de tamanho suficiente.

Outro objetivo é ter espaço para trabalhar e guardar documentos no sistema, portanto é uma boa idéia usar apenas metade do espaço disponível (64 MB) para o sistema operacional e programas afins. Obviamente, será muito difícil incluir ferramentas úteis mas enormes – como o OpenOffice e o Gimp – portanto vamos usar de bom senso.

Estamos com sorte. Muitos na comunidade Linux já estão trabalhando no problema de como acomodar o Linux em um espaço ínfimo. A idéia de colocar o Linux em um dispositivo de armazenamento USB de recursos modestos é relativamente nova, portanto não há muitas opções disponíveis. As opções a seguir, entretanto, são fortes candidatas a ganhar um lar em nosso chaveiro:

- **Damn Small Linux** [3]: A distribuição Damn Small Linux (DSL – numa tradução livre, "Linux Pequeno pra Chuchu") é baseada no Knoppix – e, por conseguinte, no Debian [4] – mas o autor reduziu o tamanho para meros 50 MBytes pela eliminação de parte da documentação e fazendo uma faxina radical em praticamente todos os diretórios. O Damn Small Linux pode inclusive carregar uma interface gráfica gráfica das mais leves, como por exemplo o Fluxbox.
- **RUNT** [5]: O RUNT (ResNet USB Network Tester) é uma distribuição baseada no Slackware [6] que trabalha em um chaveiro USB de pelo menos 128 MBytes de memória. A distribuição pode ser iniciada no computador por um disquete de boot disponível no site – é o mesmo disquete do ZipSlack [7], diga-se de passagem. O RUNT é usado como canivete suíço para manutenção e teste de redes e contém um número considerável de ferramentas para a tarefa.
- **Feather Linux** [8]: O Feather Linux (algo como "Linux Peso-Pena") é baseado no Damn Small Linux (DSL) [3] e também cabe em um chaveiro de 128 MBytes. Não é muito diferente do DSL mas inclui cerca de 14 MBytes a mais de software, além de já ter sido testado e "afinado" para uso em dispositivos USB.
- **Flonix** [9]: O Flonix também é baseado no DSL mas usa um conjunto diferente de programas – por exemplo, IceWM em vez de Fluxbox. O projeto Flonix também introduziu um sistema de instalação via web muito poderoso. O sucesso foi tão grande que virou um produto comercial e não está mais disponível para download gratuito. Entretanto, se você pro-

curar direitinho em todos os cantos da Internet vai encontrar, dentro de algum baú poeirento, alguma versão antiga (e ainda gratuita) para usar.

- **Puppy Linux** [10]: O Puppy Linux (hmmm... o "Linux do Cachorrinho") é um projeto bem recente, mas que já possui recursos de fazer inveja a seus primos mais velhos. Em primeiro lugar, carrega-se inteiramente na memória RAM do computador e roda a partir de lá. Isso o torna muito rápido, mas exige pelo menos 128 MB para funcionar. Além da RAM, seu cãozinho querido pode ser iniciado a partir de diversos tipos de dispositivos (disco rígido, CD, USB) e contém programas bastante "magros" e rápidos. A distribuição completa cabe em menos de 50 MBytes. O ambiente gráfico escolhido é o FVWM-95, com a "cara" do Windows.
- **SPB-Linux** [11]: O SPB-Linux é uma distribuição modular bastante pequena e muito poderosa. As versões mais antigas não usavam X; em vez disso escreviam direto no FrameBuffer (usando o *directFB*) com o ambiente gráfico Sawfish. Hoje, é possível usá-lo sem X, usar o XFCE 4 com X ou mesmo apenas o X com um gerenciador de janelas próprio, bem espartano. Além de conter programas como o Firefox, Java e o reprodutor de mídia Mplayer, o SPB-Linux é um projeto muito bem documentado.

Para este exemplo, vou instalar o Damn Small Linux (DSL) [3], o mais flexível. Tanto o Flonix quanto o Feather Linux são baseados nele e têm tamanho suficiente para caber em memory Sticks até menores que 128 MBytes, como os outrora comuns modelos de 64 MBytes.

Pingüim em lata de sardinha

O primeiro passo, e o mais óbvio, é fazer um backup de todos os dados que porventura estejam guardados no chaveiro. Em uma distribuição Linux já instalada, insira-o numa porta USB, abra um terminal e digite:

```
$ su
Password: (digite a senha↵
do root e pressione ENTER)
# mkdir usb_pen
# mkdir usb_backup
# mount -t vfat /dev/sda1 usb_pen
# cp -a usb_pen/* usb_backup
# umount usb_pen
```

Seu dispositivo no */dev* pode ser diferente. Se não funcionar com *sda1*, tente outro. Lembre-se de que pode ser preciso carregar o módulo usb-storage (*modprobe usb-storage*) e montar o sistema de arquivos usbfs (*mount -t usbdevfs none /proc/bus/usb*) para que o dispositivo USB seja reconhecido. Para mais informações (ou se nada funcionar como esperado) consulte as referências [12], [13], [14] e [15].

Agora que temos nossos dados a salvo, precisamos formatar nosso Memory Stick para reduzir as chances de encontrar um problema durante o boot. Para tanto, use o comando *fdisk* como root – o chaveiro deve estar conectado mas não montado.

```
# fdisk /dev/sda
```

Quando o fdisk entrar, pressione *p* para ver a lista de todas as partições presentes em seu chaveiro (normalmente apenas uma). Pressione *d* para excluir a primeira partição. Se houver mais partições, faça backup dos dados

e apague-as todas. Quando o chaveiro estiver "limpo", crie uma nova partição primária pressionando *n*, depois *p*, depois *1*. Torne-a uma partição de boot selecionando a opção *a*. Especifique o tipo de partição (*t* > *6* cria uma partição FAT16, que reconhece volumes de até 4 GB) e por último pressione *w* para gravar os dados na tabela de partição e sair do fdisk.

O próximo passo é baixar a imagem iso do Damn Small Linux, que tem aproximadamente 50 MB. Quando terminar, abra um console, logue-se como root e vá até o diretório com o arquivo "dsl-0.9.2.iso". Execute os comandos:

```
# mkdir dsl_temp
# mkdir dsl_usb
# mount -o loop dsl-0.8.3.iso ➥
dsl_temp
# cp -a dsl_temp/* dsl_usb
```

Nesse ponto, o conteúdo da imagem ISO foi copiado em um novo diretório no qual podemos trabalhar. Se tentássemos copiar esses dados diretamente para o chaveiro USB não seria possível iniciar o computador com ele, pois é preciso, antes, ajustar a posição e os nomes de alguns arquivos. Com o console aberto, emita os comandos:

```
# cd dsl_usb/
# mv boot/isolinux/* ./
# rm -Rf boot
# mv isolinux.bin syslinux.bin
# mv isolinux.cfg syslinux.cfg
# cd ..
```

A última parte do processo é bem simples: precisamos montar o chaveiro USB, copiar todos os dados dentro dele e, depois de desmontar o disco, torná-lo iniciável com o programa syslinux.

```
# mkdir usb_pen
# mount -t vfat /dev/sda1 ➥
usb_pen
# cp -a dsl_usb/* usb_pen
# umount usb_pen
# syslinux /dev/sda
```

Reinicie o computador com o chaveiro ainda conectado e veja se tudo deu certo. Não esqueça de verificar se a BIOS do computador permite o boot por dispositivos USB. Alguns dispositivos USB possuem registros-mestre de inicialização (*Master Boot Record* – MBR) defeituosos que podem apresentar problemas para inicializar pelo Linux. Para saber mais veja o quadro "Problemas mais comuns". ■

Problemas mais comuns

Alguns problemas são comuns, e você pode se considerar um sortudo se não tropeçar em algum deles. Por exemplo:

- sua BIOS não reconhece (ou não está configurada para iniciar) um sistema operacional pela interface USB
- seu dispositivo USB possui um sistema de boot não convencional. Assim, mesmo que tudo esteja nos conformes, o sistema não iniciará.

Em primeiro lugar, verifique se a BIOS possui opções de USB e se as opções de teclado (USB-keyboard) e dispositivos legados (USB-legacy) estão ativadas. A opção *Boot from Zip USB* (boot por disco Zip na porta USB) é, normalmente, um bom "chute". Depois selecione o dispositivo USB como primeiro na ordem de boot. Insira então o chaveiro na porta USB e verifique se, ao ligar a máquina, algum menu permite a inicialização por USB. Para entrar na configuração da BIOS – o chamado "setup" – pressione *del*, *F1* ou *F10* assim que solicitado, logo que ligar a máquina.

Em alguns computadores não existe possibilidade de configurar o boot por USB, já que o suporte a isso só foi introduzido muito recentemente. Para contornar esse problema, é possível usar um disquete de boot que carregue os módulos USB e passe o controle para a distribuição contida no chaveiro. É claro que carregar um disquete por aí para o boot não é tão *sexy* quanto iniciar diretamente do Memory Stick, mas é uma maneira de usar um Linux portátil sem ter que levar consigo um desajeitado e enorme CD, que além disso nem deixa você guardar seus documentos. Há uma imagem de disquete no site do DSL [16] já preparada para carregar os módulos apropriados. Depois de baixar o arquivo, crie o disquete de boot com o comando:

```
# dd if=bootusb-0.8.img of=/dev/fd0 bs=1024 conv=sync ; sync
```

Obviamente, deve ser emitido como root. Agora, desligue seu computador, insira o chaveiro USB e o disquete e ligue o computador. Se ainda assim não funcionar, é provável que sua BIOS esteja com o boot por disquete desligado. Verifique.

Outro possível problema, independente do processo de boot usado, é que alguns dispositivos USB realmente não oferecem a possibilidade de se iniciar um computador por eles. O problema pode ser causado, por exemplo, por um setor de boot (MBR) defeituoso ou simplesmente fora do padrão. Para este artigo, usei um Iomega Mini Drive de 128 MBytes (*www.iomega.com*), que funcionou de primeira e é bem rápido na transferência de dados. Outros dispositivos que testei não encaravam bem o processo de boot. Para tentar resolver o problema, a distribuição SPB-Linux possui um programa que implementa um setor de boot alternativo no dispositivo USB. Esse setor faz o que o nome deixa transparecer (ao contrário do original): permitir o boot por USB [17].

INFORMAÇÕES

[1] Knoppix: *http://www.knoppix.net*

[2] Kanotix: *http://kanotix.com*

[3] Damn Small Linux: *http://www.damnsmalllinux.org*

[4] Debian: *http://www.debian.org*

[5] RUNT: *http://www.ncsu.edu/project/runt*

[6] Slackware: *http://www.slackware.com*

[7] ZipSlack: *http://www.slackware.com/zipslack*

[8] Feather Linux: *http://featherlinux.berlios.de*

[9] Flonix: *http://www.flonix.com*

[10] Puppy Linux: *http://www.goosee.com/puppy*

[11] SPB-Linux: *http://spblinux.sourceforge.net*

[12] Tutorial de Ricardo Igarashi: *http://br-linux.org/noticias/000156.html*

[13] Tutorial de Indigo Meridian: *http://indiboi.com/history/2003/11/18/2887223*

[14] Linux USB Devices: *http://www.qbik.ch/usb/devices/*

[15] Linux USB: *http://www.linux-usb.org/*

[16] Imagem do disquete de boot USB do DSL: *http://ftp.belnet.be/packages/damnsmalllinux/current/bootusb-0.8.img*

[17] Setor de boot alternativo: *http://home.tele2.ch/spblinux/spbsetup/Linux/spb2_mbr.sec*

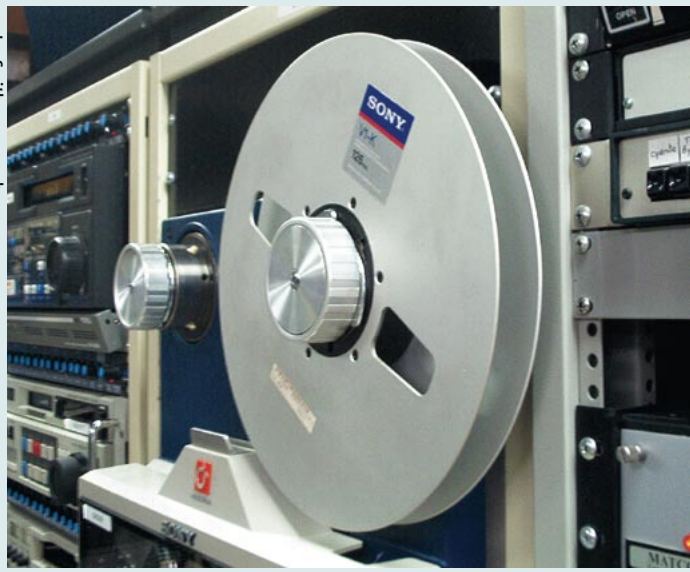
SOBRE O AUTOR

*Fabrizio Ciacchi (*http://fabrizio.ciacchi.it*) é estudante de Ciências da Computação na Universidade de Pisa. Trabalha como consultor para diferentes companhias e escreve artigos para a edição italiana da* **Linux Magazine** *(*www.linux-magazine.it*).*

Seguro morreu de velho

Garantindo o plano B

Segurança é essencial. Essa é uma afirmação raramente questionada e a razão de existirem os "planos B", como são popularmente conhecidas as alternativas que mantemos de prontidão, mas que preferimos nunca ter de usar. POR AUGUSTO CAMPOS



por Joe Smith: www.sxc.hu

O seguro da casa, mesmo quando há sistema de alarme; o cofre reforçado, mesmo onde não há alta incidência de roubos; o conjunto de remos guardado debaixo dos bancos da lancha a motor... são infindáveis os exemplos, no cotidiano, de situações em que julgamos ser melhor prevenir do que remediar.

Na informática, um dos exemplos mais claros de plano B é o backup. Quem tem um bom plano de backup o segue na esperança de que nunca precise usá-lo. Quem não tem, muitas vezes aprende a importância dele apenas quando já é tarde demais. Por mais seguro e inatingível que seja o seu servidor ou o seu desktop, sempre há alguma possibilidade de dano aos dados.

Ao definir sua estratégia de backup, é necessário levar em conta diversos fatores: custo, disponibilidade de mídias e gravadores, disponibilidade de operador, restrições técnicas de todos os sistemas envolvidos e outros mais. Ao planejar, leve em conta quais dados não podem ser perdidos em nenhuma hipótese (que tal guardar uma cópia atualizada deles em segurança e fora do prédio?) e em quanto tempo você precisaria ter tudo funcionando novamente após um evento de perda total.

Profissionais de administração de sistemas não deveriam precisar das dicas acima, pois são ensinadas em todo manual e curso da área. Mas insisto em mencioná-las mesmo assim porque cada vez mais os backups devem ser preocupação até mesmo dos usuários domésticos pois, com a proliferação do conteúdo digital, as perdas de discos domésticos começam a se tornar verdadeiros dramas familiares, com o desaparecimento súbito de todas as fotos, trabalhos escolares, relatórios, emails da família, catálogos de endereços, coleções de músicas, registros do imposto de renda e tudo o mais que hoje se guarda no PC doméstico comum.

A novidade é que hoje os backups estão ao alcance do usuário doméstico de Linux, sem necessidade de fazer um curso de introdução à administração de sistemas. Embora os mais experientes tendam muitas vezes a preferir usar, em suas estações de trabalho, as mesmas técnicas tradicionais adotadas nos servidores e redes que administram, já há alternativas amigáveis voltadas justamente ao usuário doméstico.

Um exemplo de aplicativo amigável para backup é o Kdar. Esse programa, integrado ao ambiente KDE, é uma interface agradável e simplificada (mas sem ser restritiva) para as tarefas mais corriqueiras de cópia e recuperação de dados. As características mais comuns e essenciais estão ao alcance do mouse: compressão (com gzip ou bzip2), separação do backup em arquivos de tamanho fixo (para melhor caber em CDs, fitas, Zip Disks ou outra mídia externa que você use), atributos estendidos, backup incremental e muito mais. O visual é agradável e a interação é bastante intuitiva. Aparentemente ainda não há suporte ao nosso idioma, mas isto é algo que a própria comunidade pode resolver (quem sabe algum leitor da Linux Magazine não se habilita?).

Quase posso imaginar os usuários experientes se perguntando como recuperar um backup desses em situações extremas, quando não se consegue fazer um logon completo e ativar o ambiente gráfico. Mas a solução é simples: há um utilitário chamado dar (independente do KDE ou do ambiente gráfico e similar ao tradicional tar) exatamente para essa tarefa, que pode até mesmo ser incluído em seu disco de recuperação para emergências.

Mas o mais importante não é a escolha da ferramenta: é a política de backups. Estabeleça uma e siga! Não descubra tarde demais que o seu backup mais recente é de 2 meses atrás porque ninguém lembrou de gravar os CDs, inserir a fita ou deixar o PC ligado no horário em que é gravada a cópia. ■

INFORMAÇÕES

dar: *http://dar.sourceforge.net/*

kdar: *http://kdar.sourceforge.net/*

SOBRE O AUTOR

Augusto César Campos é administrador de TI e desde 1996 mantém o site BR-Linux.org, onde cobre a cena do Software Livre no Brasil e no mundo. Foi colunista e autor de diversos artigos na Revista do Linux.

Checkpoint FW1 e Firewall Builder

Regras para todos

A tecnologia de firewall presente no kernel do Linux já há algum tempo atende às exigências profissionais. Agora, os configuradores gráficos chegaram para diminuir a distância entre o Linux e as soluções comerciais. POR CHRISTIAN NEY

Empresas que usam linhas privadas para se conectar à Internet precisam de muita proteção contra ataques à sua rede interna. Em muitos casos, os firewalls são a primeira linha defensiva, permitindo que os administradores definam o tipo de tráfego autorizado a entrar e sair de suas redes corporativas. De quebra, um firewall também pode restringir o acesso a serviços da Internet para os funcionários e usuários do sistema.

O mercado continua a crescer, principalmente devido à penetração da própria Internet. O leque de produtos inclui desde opções livres e de código aberto (normalmente gratuitas) a soluções comerciais esplendidamente caras e com "satisfação garantida ou seu dinheiro de volta".

Praticamente todas as soluções comerciais oferecem uma forma fácil de se configurar e administrar o firewall – normalmente um programa gráfico. Em oposição a isso, a grande maioria dos firewalls de código aberto impingem ao operador uma linha de comando que não é exatamente fácil de usar. Essa é a razão pela qual muitos administradores evitam usar qualquer sistema envolvendo Software Livre.

Obviamente, uma boa interface não pode substituir o conhecimento e a experiência. Nenhuma interface gráfica vai formar especialistas de firewall do dia para a noite, mas a maioria das pessoas prefere uma abordagem visual do problema em vez de uma interminável lista de longos comandos. É mais fácil implementar um projeto de segurança se houver ferramentas gráficas disponíveis para auxiliar na tarefa.

Nossa rede de exemplo

Basearemos nossos exemplos na rede mostrada na figura 1. Mostraremos como implementar políticas de segurança usando o CheckPoint Firewall 1 NG, comercial, e o Firewall Builder, um programa de código aberto. Nossa rede usa um roteador para se conectar à Internet. A empresa fictícia que possui essa rede decidiu que hospedará seu site num servidor próprio. O servidor ficará localizado numa *DMZ* (*Zona Desmilitarizada*, uma rede à parte, com menos restrições de acesso) em vez de diretamente na rede interna. Os empregados da companhia devem poder acessar serviços de FTP, HTTP e HTTPS na Internet protegidos por um proxy interno. Um servidor de email na LAN usará o protocolo POP3 para baixar as mensagens de email de um provedor externo. Isso permite que as mensagens sejam verificadas quanto a presença de vírus em um único local.

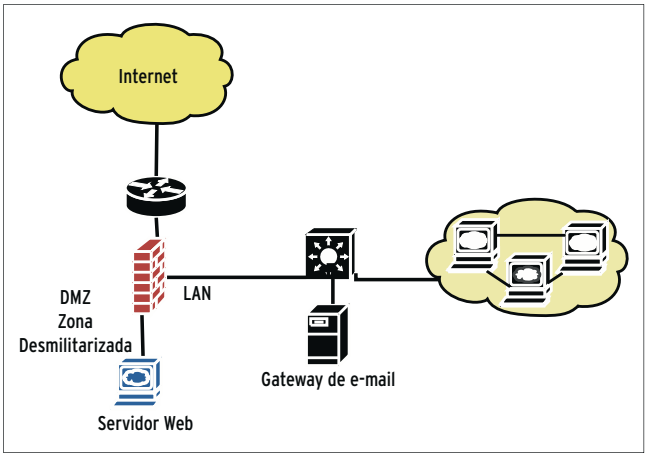


Figura 1: Um exemplo típico de configuração de uma rede pequena, mostrando a DMZ separada da rede principal.

O servidor de email é o único computador com permissão para enviar mensagens para fora usando o protocolo SMTP. O firewall terá proteção adicional, pois só poderá ser acessado por um único computador dentro da rede interna, e apenas por meio de SSH.

Um genuíno faz-tudo

O Firewall 1 NG (Next Generation), da empresa CheckPoint [1] é provavelmente o mais famoso produto comercial dessa categoria, sendo considerado um genuíno faz-tudo. O CheckPoint não só oferece um firewall com uma reputação ilibada de ser extremamente seguro como pode, a preços não tão módicos assim, ser estendido para oferecer uma solução de VPN que permite até mesmo a conexão com soluções de código aberto – como o FreeS/WAN, por exemplo. Infelizmente, até o presente momento o mundo do Software Livre foi incapaz de apresentar uma solução tão completa quando essa. O CheckPoint possui uma interessante interface de configuração chamada de *SmartDashboard* (algo como "Painel Inteligente").

O CheckPoint é composto por dois tipos de módulos:

- O *Management Module* (módulo de administração) é usado para "compilar" as regras criadas com o uso da interface. Trocando em miúdos: converter as regras em linguagem humana para um formato que o firewall possa entender. O módulo transfere essas regras para um ou mais firewalls e administra os módulos de registro (log), os objetos usados na configuração e os bancos de dados que autenti-

cam os usuários com direito de uso do firewall. O Certificate Authority (uma espécie de "cartório digital") é outro módulo importante. O CA gerencia os certificados emitidos para qualquer sistema autorizado a usar um dado recurso e roda no computador que administra o sistema. Essa arquitetura tem a vantagem de possibilitar a centralização do gerenciamento de múltiplos firewalls. Quaisquer computadores envolvidos em uma transação de rede devem confiar no servidor central.

- Módulos individuais de controle de tráfego, chamados "Enforcement Modules", são usados para montar dispositivos que agem como filtros de pacotes baseados em regras. Em outras palavras, eles são o que comumente se chama de firewall. Os módulos ajustam automaticamente o sistema operacional nos quais rodam para que se tornem menos vulneráveis a ataques – em inglês, chamamos isso de "*hardening*". Isso deixa apenas uns poucos itens para os administradores ajustarem manualmente, como por exemplo desabilitar os muitos serviços desnecessários sempre presentes no /etc/inetd.conf.

O conjunto de regras de filtragem é criado por um aplicativo especial, normalmente instalado em uma máquina cliente. O administrador pode usar sua interface gráfica para criar as regras, mas a informação é gerenciada de forma centralizada pelo Management Module.

Separando a interface gráfica dos módulos de administração

Obviamente, todos os módulos e até mesmo a interface gráfica podem rodar em uma única máquina. Essa configuração, entretanto, não é recomendada, já que interferiria no desempenho global do sistema. O registro de eventos (logging) impõe uma carga por demais pesada na máquina. Por razões de segurança, o Management Module deve rodar em um servidor dedicado. Esse arranjo também proporciona uma segurança adicional, uma vez que todas as máquinas envolvidas podem ficar em uma rede separada.

Em ambiente de produção, muitas empresas podem optar por rodar ambos os módulos em uma única máquina, embora a interface de administração tipicamente esteja instalada na estação de trabalho do administrador. Infelizmente, essa máquina deve, obrigatoriamente, rodar Windows ou Solaris, pois a interface de administração não possui uma versão para Linux.

A única distribuição de Linux que o módulo de firewall suporta, oficialmente, é o Red Hat. Com **muito** trabalho é possível fazer o mesmo com um Debian. A instalação, entretanto, é um desafio e tanto, pois os procedimentos são um tanto centrados no Red Hat. Em listas de discussão que tratam do *CheckPoint* [11] descobrimos que é possível instalar o FW-1 no SuSE, no Mandrake e no Slackware. Kernels especiais são necessários em alguns casos. Como de praxe, essas variações são levadas a termo por conta e risco dos especialistas envolvidos, pois o fabricante não prestará suporte sob nenhum pretexto.

A interface gráfica do CheckPoint está dividida em quatro painéis por padrão (ver figura 2). Você pode ajustar, é claro, o modo de visão de forma a satisfazer suas necessidades e preferências. O painel à esquerda mostra

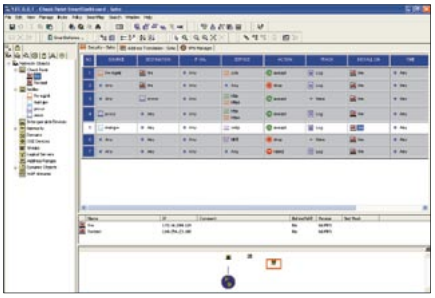


Figura 2: Exemplo de um conjunto de regras de filtragem criada com o CheckPoint SmartDashboard.

uma lista de objetos – firewalls, computadores, serviços e o que quer que um administrador atarefado precise para criar regras de filtragem. O *CheckPoint* oferece, felizmente, uma vasta seleção de definições. Até serviços praticamente extintos, como o Gopher, estão presentes, o que retira das costas do administrador um peso bastante incômodo. Com o tempo livre, o responsável pela segurança da rede pode criar novos objetos, pensar melhor nas regras ou ter tempo para um café.

Policy Editor

O *Policy Editor* (editor de regras, canto superior direito) ocupa uma grande parte da tela. O editor possui abas para as regras de filtragem, para as definições de tradução de endereços (NAT) e, dependendo da sua licença (e do quanto você pagou), outros objetos como o VPN Manager. Na primeira vez que você abre o programa, nenhuma regra de filtragem estará definida. A primeira tarefa é criar um gateway (concentrador de conexões para a Internet) que, futuramente, será também o firewall. Para isso, pode-se tanto usar um *wizard* (assistente) quanto fazer tudo manualmente. É importante adicionar o módulo ao SIC (Secure Internal Communication).

O fato de que objeto *firewall* possua internamente recursos de *anti-spoofing* entre suas regras nos dá uma camada adicional de proteção interna. Essas regras asseguram que invasores não possam conseguir acesso usando técnicas de impostura – em outras palavras, fingindo ser alguém de dentro. O exemplo clássico é o *IP Spoofing*: o atacante forja um endereço IP que pertença à nossa rede interna. Quando você ativa as regras de filtragem do firewall, o editor emite um aviso para cada interface

Inspeção de estado versus tabelas de estado

Em listas de discussão e newsgroups, usuários novatos freqüentemente fazem perguntas a respeito da diferença entre um filtro de pacotes por estado de conexão (o chamado *Stateful Firewall*) e a técnica usada no CheckPoint, batizada de *Stateful Inspection* (inspeção de estado) – e isso não é surpresa, pois o pedante nome dado pela empresa leva a uma conclusão errônea. Todos os filtros de pacotes discutidos neste arquivo oferecem a opção de simplificar o conjunto de regras usando as chamadas tabelas de estado (*state tables*). O uso dessas tabelas melhora o desempenho do filtro de pacotes e, ao mesmo tempo, a segurança.

Inspeção de estado de conexão

O Checkpoint usa a mesma tecnologia [10], mas também analisa os dados úteis do pacote – ou seja, os dados da camada de aplicação, da mesma forma que um proxy. Para conseguir isso, faz uso de scripts especialmente criados para a tarefa. Como resultado, o monitoramento das comunicações é muito mais eficiente.

que ainda não esteja com o *anti-spoofing* ligado, como um lembrete para que você pare tudo e faça isso. As mensagens também são gravadas no registro de eventos do sistema (syslog).

Com essas etapas cumpridas, podemos começar a definir os objetos necessários. Uma regra, basicamente, é composta por:

- Objeto de origem (Source);
- Objeto de destino (Target);
- O serviço usado para a comunicação;
- Uma restrição para a regra se a conexão usa uma VPN específica;
- Definição de registro no *log* se a regra for usada;
- Em quais firewalls essa regra deve ser aplicada – isso permite definir a mesma regras para vários firewalls;
- O horário em que essa regra deve estar ativa;
- Um comentário sucinto e informativo, que esclareça a qualquer um o objetivo dessa regra.

A regra precede uma lista de objetos, que é, no fundo, uma visão mais detalhada sobre os objetos criados pelo usuário. A lista revela várias das características intrínsecas de cada um, como, por exemplo, seu número IP. Isso pode ser extremamente útil quando se tem que lidar com um grande número de objetos e precisamos vislumbrar rapidamente os IPs usados em uma subrede específica.

Visão geral da rede

O último painel contém o *Visual Policy Editor*, um mapa gráfico de todos os nós, suas interligações e a infra-estrutura da rede. Um mapa desse tipo é sempre bárbaro caso existam conexões entre um punhado de computadores usando uma VPN – além de permitir que o administrador de rede se exiba para os colegas e superiores quando o chefe pede "para ontem" um relatório com o diagrama atualizado da rede. É possível exportar o diagrama como arquivo de imagem no formato do Microsoft Visio. Isso permite manter a documentação da rede sempre em dia – coisa extremamente necessária se os firewalls e a rede são administrados por uma equipe. Infelizmente, tudo isso só está disponível se você pagar – e não é barato!

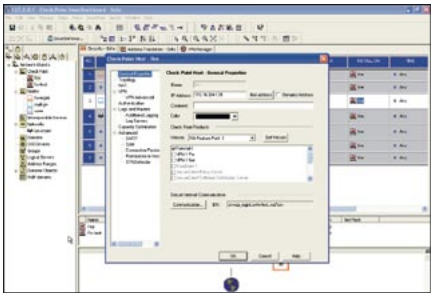


Figura 3: Os objetos do CheckPoint Firewall-1 permitem que quase qualquer tipo de operação de rede possa ser feita.

É muito simples criar regras individuais. Simplesmente arraste e solte os objetos desejados nos campos apropriados. É possível usar o mouse para deslocar regras – recurso que não estava disponível nas versões anteriores. A estrutura bem definida da interface torna a vida do administrador bastante simples, mesmo para os que não possuem muita experiência com esse tipo de tecnologia. Entretanto, a quantidade fabulosa de opções, nem todas óbvias ou aparentes, sempre deixa algum espaço para otimizações.

O recurso de permitir que o firewall faça tradução de endereços (NAT) para qualquer objeto é utilíssimo e pode poupar bastante trabalho. Obviamente, você pode configurar o NAT manualmente. Dependendo de como for feito, isso pode significar alguma configuração de roteamento adicional no firewall para refletir a estrutura real de NAT da sua rede – em versões anteriores, isso era *sempre* necessário. Por exemplo, pode ser preciso definir uma rota a partir do IP externo "válido", atrás do qual todos vão se esconder, para o IP interno.

Versões posteriores ao HotFix 3 da CheckPoint podem monitorar tráfego de rede em busca de atividade malévola, como o "Ping da Morte" ou comandos SMTP e SMNP ilegais. Esse recurso é chamado de *SmartDefense*, para combinar com os outros produtos da linha "Smart". Fique atento para as novidades nessa área.

Depois de conseguir uma configuração que funcione, podemos passar à árdua tarefa de deixar o sistema mais seguro, já que o firewall não trabalha apenas no nível dos pacotes (camada 3 do modelo OSI) mas também analisa o tráfego dos serviços como um IDS – *Intrusion Detection System* ou *Sistema de Detecção de Intrusos*.

Serviço de tradução

Depois de completar a lista de regras e o trabalho de pré-instalação, é preciso transferir as regras para a estação de administração num formato que os firewalls possam entender. No jargão da CheckPoint, isso é chamado de "compilação". O arquivo de texto com as regras de filtragem, legível para os humanos, é convertido num formato interno denominado *Inspect script*, que por sua vez é reconvertido no chamado *Inspect code*. É esse código que será instalado nos firewalls. *Inspect* é uma linguagem de script criada pela CheckPoint especificamente para seus produtos. Isso permite que os especialistas (e os inimigos das interfaces gráficas) possam fazer as coisas manualmente.

Como as modificações manuais não alteram os objetos mostrados na interface gráfica – o que só acontece durante a compilação – é melhor deixarmos as alterações manuais para os especialistas. O perigo de seu conjunto de regras de filtragem ficar inconsistente é gigantesco,

Firewall por estado de conexão

Essa tecnologia é baseada num princípio assaz simples. Como uma conexão TCP será sempre iniciada por um pacote SYN, o filtro usa as regras para verificar se esse tipo de comunicação é permitida. Se for, a conexão com o cliente é, primeiro, estabelecida, e depois incluída em uma tabela de estados como tal. As tabelas são armazenadas na memória do kernel, permitindo acesso extremamente rápido. O filtro de pacotes pode então usar esses dados para verificar os pacotes seguintes (que não possuem o flag SYN ligado) e decidir se eles pertencem – ou não – a uma conexão autorizada. Em outras palavras, não há necessidade de escrever um punhado de regras para lidar com todos os aspectos da conexão. Por estarem relacionados àquela primeira conexão já listada na tabela, os pacotes são autorizados automaticamente.

A redução do número de regras proporciona um desempenho bem maior para o firewall, permitindo que o sistema reconheça pacotes contíguos, como os pertencentes a uma conexão FTP, que não tenham relação direta com a conexão estabelecida previamente e marcada na tabela. O Linux usa um "ajudante" (*helper*) para esse fim. Esse comportamento também aumenta a segurança porque não há a possibilidade de se abrir múltiplas portas – veja a discussão sobre o assunto em [9].

especialmente se levarmos em conta que a documentação é bastante inadequada – mesmo que tenhamos pago por ela.

O CheckPoint tem seu recurso de propagação proprietário, chamado de *Secure Internal Communication* (SIC) para transferir o conjunto de regras de filtragem aos *Enforcement Modules*. O protocolo é baseado na conhecida técnica SSL / TLS, e usa Criptografia de Chave Pública para verificar a identidade dos nós envolvidos na transação, assegurar a integridade dos dados e criptografar todo o tráfego da rede. Com isso, a confidencialidade e a segurança dos dados é garantida.

O CheckPoint Firewall 1 NG é particularmente útil para aqueles que precisam de uma solução completa, do tipo "satisfação garantida ou seu dinheiro de volta". O produto é particularmente recomendado quando se pretende usar algum tipo de VPN, pois oferece o recurso de gerenciamento centralizado. Se você possui sistemas que precisam de alta disponibilidade, não há muita escolha. Os filtros de pacotes do mundo do Software Livre têm severas restrições no tocante a recursos desse tipo. Um exemplo básico: sincronização de tabelas que funcione direito e seja estável.

Irmão caçula

Se você já trabalhou com o CheckPoint antes, o Firewall Builder [3] o fará sentir-se em casa. Em aparência, é bem semelhante ao *SmartDashboard* do FW-1. A metodologia de separar a administração em uma interface gráfica e manter os firewalls de longe também foi "adotada".

O Firewall Builder permite gerenciar múltiplos firewalls ao mesmo tempo. Há quatro tecnologias de filtragem reconhecidas:

- iptables [4] (Linux com kernel da série 2.4 ou 2.6)
- ipfilter [5] (FreeBSD e NetBSD, com versões para vários outros "sabores" de Unix)
- pf [6] (OpenBSD)
- Cisco Pix

O Cisco Pix possui sua própria interface HTML, mas ela é usada apenas para a configuração inicial. Para que o Firewall Builder possa administrar essa plataforma, é necessário adquirir um módulo adicional, não-livre. Seu preço, entretanto (500 dólares americanos [7]) é baixo o bastante para ser uma pechincha em relação a outros produtos.

A abordagem modular torna o Firewall Builder interessante quando é preciso administrar de forma centralizada firewalls de várias procedências. Também é bastante indicado se um administrador já acostumado com o CheckPoint precisa trocar de plataforma.

O Firewall Builder não só pode trabalhar com múltiplos filtros de pacotes como também rodar em muitos sabores de Unix. Até o momento da edição, havia pacotes para inúmeras distribuições Linux, FreeBSD, OpenBSD e até para o Mac OS X.

Regras para mais de um firewall

Assim como o Checkpoint, o Firewall Builder é dividido em duas partes: o programa de administração, que roda na estação de trabalho do administrador e possui uma interface gráfica, e o firewall propriamente dito, rodando normalmente em máquinas que servem como *gateways*. Em contraste com o Checkpoint, usa um sistema de configuração para criar os conjuntos de regras e então transfere-as para as máquinas onde rodam os firewalls. Não há um mecanismo de transferência próprio, como no Checkpoint, mas o administrador pode usar o bom e velho SSH. Um daemon independente que cuide da transferência das regras, rodando em cada firewall, está no forno para as próximas versões – e isso já se reflete na interface gráfica – mas ainda não está disponível. Como seu "irmão maior", o Firewall Builder usa conexões protegidas por Chaves Públicas para a comunicação entre as máquinas.

A versão atual possui um script chamado *fwv_install*, muito útil ao administrador do firewall. O script faz uso de autenticação SSH via Chave Pública para transferir as regras aos firewalls. Depois de definir o conjunto de regras adequado, basta selecionar a opção *Rules* > *Install* para transferi-lo.

A interface gráfica chama o script *fwb_install* em segundo plano e mostra a saída do comando em uma nova janela. Para evitar redigitação, o sistema ativa o programa *ssh-agent*. O agente armazena todas as suas informações de conexão e senha até que você o desative novamente.

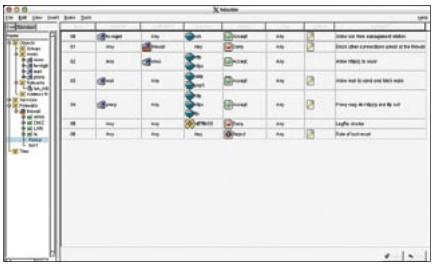


Figura 4: Um conjunto de regras de filtragem criado pelo Firewall Builder, como exemplo de uma possível solução para a rede mostrada na Figura 1.

Portátil, graças ao XML

O administrador pode definir um certo número de opções de transferência para o script. Por exemplo, o local para gravar o arquivo com o conjunto de regras no sistema de arquivos do firewall. Também é possível definir se o arquivo XML usado para "compilar" o conjunto de regras deve ser gravado junto com o arquivo "compilado" ou qual usuário se registrará ("logará") no firewall para a transferência e outras operações.

O procedimento tem a vantagem de evitar que o usuário *root* opere o firewall, propiciando um considerável aumento de segurança. Observe que o conjunto de regras não só é transferido mas também automaticamente ativado pelo script, coisa que requer privilégios de root. Se você definir um usuário com menos privilégios, certifique-se de que sua conta esteja cadastrada no sudo (*/etc/sudoers*).

Da mesma forma que com o *Checkpoint*, o Firewall Builder não trará nenhuma regra quando for executado pela primeira vez. Entretanto, já possui definições para os serviços mais comuns nos protocolos TCP e UDP e mensagens ICMP, além de já conhecer os endereços das LANs privadas. É possível usar um assistente para adicionar os computadores conectados à sua rede. Para detectar os nós de sua rede, o assistente pode – entre outras "espertezas" – ler seu arquivo */etc/hosts*, fazer uma transferência de zona DNS (*zone transfer*), usar SNMP ou simplesmente rodar um *scanner*.

Quando testamos essa função, o programa produziu um erro a cada vez que fazia uma consulta ao DNS. Por outro lado, usando SNMP é possível obter os dados de contato, a localização e até a descrição do objeto em questão. Isso

Figura 5: O Firewall Builder permite uma configuração bastante granular das várias opções de filtragem.

aumenta a base de conhecimento do administrador sobre a rede e pode (deve!) ser usado para documentá-la melhor.

Interfaces semelhantes – plágio?

O assistente incluído com o programa é uma ferramenta bastante útil para administradores inexperientes que precisam construir rapidamente um conjunto de regras inicial. Metódico, cuida das primeiras coisas primeiro: define uma DMZ e usa os dados iniciais para criar sozinho um simples mas eficaz conjunto básico de regras de filtragem. Isso cria uma fundação sólida sobre a qual é fácil construir um firewall, seja ele simples ou complexo. Obviamente, esse assistente não impede que você crie – manualmente, como bom *hacker* – os objetos de que precisar. Também não impede que você use os outros "druidas" disponíveis.

A interface gráfica lembra muito a do Checkpoint, incluindo campos muito semelhantes:

- Objeto de origem (Source);
- Objeto de destino (Target);
- O serviço usado para a comunicação;
- Uma ação a ser tomada caso a regra se aplique a um determinado pacote (liberar, rejeitar, bloquear e contabilizar);
- Definição de registro no *log* se a regra for usada;
- O horário em que essa regra deve estar ativa;
- Um comentário sucinto e informativo, que esclareça a qualquer um o objetivo dessa regra.

Além dessas regras "globais", cada interface de rede do firewall pode ter suas próprias regras, especificando a direção do fluxo de tráfego da rede (entrando, saindo ou ambos). O CheckPoint costumava ter essa opção, mas por algum motivo ela foi retirada da versão 5, a mais recente.

No caso do Firewall Builder, a opção pode ser uma armadilha, pois se aplica a uma única interface de rede no firewall, não à rede que se conecta a essa interface. Por outro lado, a opção permite emular as regras de anti-spoofing do CheckPoint de forma bastante próxima, evitando que potenciais invasores forjem pacotes IP que pareçam vir de sua LAN interna, permitindo que ganhem acesso não autorizado pela interface externa.

O Firewall Builder também permite configurar a tradução de endereços (NAT). Ao contrário do CheckPoint, casos especiais em que um nó interno precisa ser acessado de fora precisam ser roteados manualmente. O programa não oferece esse recurso automaticamente.

Arrastando e soltando regras e objetos

Ambos os produtos permitem que se crie e mova objetos e regras usando o mouse. As mudanças aplicadas em um objeto são imediatamente propagadas para cada instância da regra no conjunto. As diferenças estão nos bastidores. Embora o Firewall Builder não ofereça a mesma gama de funções que o CheckPoint, a grande vantagem da solução livre é sua independência de um filtro de pacotes específico.

Uma solução possível para a rede mencionada no início do artigo (ver figura 1) é mostrada na figura 4. As regras #0 #1 asseguram que a máquina chamada *fw-mgmt* seja a única com permissão de usar SSH para acessar os firewalls, bem como de registrar nos *logs* esses acessos. A regra #2 permite acesso por meio dos protocolos HTTP e HTTPS ao servidor Web que está na DMZ. Como o servidor web possui seus próprios arquivos de log, não ativamos o registro de eventos para ele.

As regras 3 e 4 permitem que o proxy e o servidor de email na rede interna acessem a Internet. Todo e qualquer acesso é registrado. A regra 5 serve apenas para limpar os arquivos de log,, apagando os eventos gerados pelos *broadcasts* NetBIOS enviados à LAN pelas máquinas com Windows. Os arquivos ficariam extremamente atravancados se não fossem limpos regularmente.

Os últimos blocos de regras bloqueiam qualquer acesso não explicitamente permitido por qualquer outra regra. Em contraste com as duas outras regras que bloqueiam acesso, essa regra usa a ação REJECT em vez de DROP. Isso permite que a conexão seja finalizada como se deve, usando a flag RST do protocolo TCP. Se fosse usado DROP, não haveria resposta para pacotes entrantes, o que levaria a um incômodo *timeout* depois de um período pré-determinado de tempo.

Sem complicação!

Obviamente, essas regras podem ser refinadas para incluir recursos, como por exemplo permitir apenas pacotes ICMP Redirect do roteador conectado aos firewalls, possibilitando a detecção de ataques nesse nível. É uma questão de compromisso entre risco e eficiência.

Há uma boa razão para manter as coisas simples quando estiver projetando seu firewall e montando seus conjuntos de regras. Qualquer regra que obrigue o kernel a verificar cada pacote que chegue irá diminuir o desempenho do sistema. Além disso, as regras devem ser inteligíveis para o administrador. Um bom conjunto de regras pode ser inútil se ninguém souber como funciona.

O Firewall Builder armazena o conjunto de regras e os objetos associados em um arquivo XML bem simples. Os compiladores interpretam, então, esse arquivo para gerar as instruções para o firewall apropriado. Usar o XML como formato intermediário permite que os administradores possam olhar o "motor" e descobrir o que o sistema está fazendo enquanto o chefe toma café.

O produto final não tenta esconder o fato de que o programa usa scripts altamente automatizados para cada uma das plataformas de firewall suportadas, poupando trabalho para o administrador, que teria que fazer tudo "na unha". Com a exceção do PIX, todos são shell scripts extremamente espertos que incluem instruções para os filtros de pacotes. Os scripts também detectam endereços dinâmicos (DHCP) e alteram a configuração para que reflita as mudanças.

A configuração abrange alguns detalhes normalmente esquecidos quando escrevemos as mesmas regras manualmente, como por exemplo prevenção de spoofing para todas as interfaces de rede.

Listagem 1: Regras em uso

```
01 [...]
02
03 echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
04 echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
05 echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
06 echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
07 echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
08 echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
09 echo 1800 > /proc/sys/net/ipv4/tcp_keepalive_intvl
10 echo 1 > /proc/sys/net/ipv4/tcp_syncookies
11 [...]
12
13 #  Regra 0(NAT)
14 $IPTABLES -t nat -A POSTROUTING -o eth2 -s 192.168.0.0/24 -j MASQUERADE
15 [...]
16
17 $IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
18 $IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
19 $IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
20 [...]
21
22 # Regra 0(eth2)
23 # Regra para ativar o anti-spoofing
24 $IPTABLES -N eth2_In_RULE_0
25 $IPTABLES -A INPUT -i eth2 -s $interface_eth2 -j eth2_In_RULE_0
26 $IPTABLES -A INPUT -i eth2 -s 192.168.1.1 -j eth2_In_RULE_0
27 $IPTABLES -A INPUT -i eth2 -s 192.168.0.1 -j eth2_In_RULE_0
28 $IPTABLES -A INPUT -i eth2 -s 192.168.0.0/24 -j eth2_In_RULE_0
29 $IPTABLES -A INPUT -i eth2 -s 192.168.1.2 -j eth2_In_RULE_0
30 $IPTABLES -A FORWARD -i eth2 -s $interface_eth2 -j eth2_In_RULE_0
31 $IPTABLES -A FORWARD -i eth2 -s 192.168.1.1 -j eth2_In_RULE_0
32 $IPTABLES -A FORWARD -i eth2 -s 192.168.0.1 -j eth2_In_RULE_0
33 $IPTABLES -A FORWARD -i eth2 -s 192.168.0.0/24 -j eth2_In_RULE_0
34 $IPTABLES -A FORWARD -i eth2 -s 192.168.1.2 -j eth2_In_RULE_0
35 $IPTABLES -A eth2_In_RULE_0 -m limit --limit 10/second -j LOG --log-level info --log-prefix "RULE 0 -- DENY "
36 $IPTABLES -A eth2_In_RULE_0 -j DROP
37 [...]
38
39 # Regra 0(global)
40 # Permite conexão SSH para o firewall vinda apenas da máquina do administrador.
41 $IPTABLES -N RULE_0
42 $IPTABLES -A INPUT -p tcp -s 192.168.0.2 -d $interface_eth2 --destination-port 22 -m state --state NEW  -j RULE_0
43 $IPTABLES -A INPUT -p tcp -s 192.168.0.2 -d 192.168.1.1 --destination-port 22 -m state --state NEW  -j RULE_0
44 $IPTABLES -A INPUT -p tcp -s 192.168.0.2 -d 192.168.0.1 --destination-port 22 -m state --state NEW  -j RULE_0
45 $IPTABLES -A RULE_0 -m limit --limit 10/second -j LOG --log-level info --log-prefix "RULE 0 -- ACCEPT "
46 $IPTABLES -A RULE_0  -j ACCEPT
```

Menos regras, melhor entendimento

O processo cria um conjunto de regras bastante abrangente mas, ao mesmo tempo, pequeno o suficiente para ser facilmente digerido por homens e máquinas. Os recursos do iptables são muito bem usados, incluindo aí as tabelas de estado de conexão e os "atalhos" disponíveis no Netfilter. Regras abrangendo múltiplas portas são apenas um exemplo. Com elas, uma mesma regra pode ser aplicada a vários serviços. Isso evita criar um grande número de regras idênticas, uma para cada porta.

A aba Firewall no objeto de mesmo nome permite que se especifique quantos desses truques podem ser aplicados. Também é possível definir o nível de registro no log para o filtro de pacotes.

A listagem 1 dá alguns exemplos. Todos esses itens estão incluídos no conjunto de regras. Mas preste muita atenção: você precisa, **obrigatoriamente,** saber o que está fazendo! Se agir sem conhecimento de causa, pode acabar escrevendo regras para funções do iptables que sequer estão implementadas.

Ainda na listagem 1, vemos muitos dos conceitos usados pelo Firewall Builder na criação das regras, baseado no que fizemos na interface gráfica. O script não só leva em conta endereços dinâmicos como também usa o sistema de arquivos */proc* para modificar opções e comportamento do kernel e, por exemplo, habilitar recursos como o redirecionamento de IP (*IP forwarding*). Observe as verificações de consistência que permitem ao kernel certificar-se de que o endereço IP foi originado na inter-

face correta (*rp_filter*) e definir temporizações seletivas para algumas conexões.

As regras para NAT ficam no início

O conjunto de regras começa, efetivamente, definindo a tradução de endereços (NAT). Isso evita problemas potenciais na interação entre o NAT e as demais regras. Essa seção é seguida pelas regras que asseguram a permissão das conexões já estabelecidas. Isso simplifica em muito o conjunto de regras. O firewall pode automaticamente reconhecer fluxos de dados relativos a uma conexão já existente, como o FTP, por exemplo, e não precisa de regras adicionais para tratar todas as situações decorrentes.

As regras exclusivas das interfaces de rede (regras de anti-spoofing) são definidas antes das regras "globais". Essa etapa pode ser deixada de fora, se desejado, e o módulo do kernel pode ser usado no lugar. Depois disso temos, enfim, as regras globais. Como o exemplo mostra, uma cadeia é definida para cada regra. Isso simplifica a contabilidade em um estágio posterior, mesmo que o método pareça mais complicado do que realmente é.

A regra #0 (global) no conjunto mostra que o software segue sem desvios o projeto do administrador para o firewall. Mesmo se o administrador quiser usar SSH no lado interno do firewall, aplicar essa regra ao objeto firewall abriria a porta 22 em todas as interfaces – incluindo aí a Internet.

Se você estiver interessado em como essas regras funcionam na prática, recomendamos estudar com afinco a configuração gerada por nossa rede de exemplo. Isso irá melhorar sua curva de aprendizado.

Especialidades do netfilter

Permitir a criação de serviços personalizados é mais uma dentre as muitas funções interessantes do Firewall Builder. Com esse truque, torna-se possível usar opções altamente específicas do filtro de pacotes que está sendo configurado. Em particular, uma combinação das extensões do Netfilter/iptables pode oferecer um número bastante grande de recursos avançados em comparação com o filtro de pacotes do kernel padrão.

Um exemplo é o *string patch* [8], que permite analisar o fluxo de dados à procura de uma cadeia de caracteres específica e aplica regras específicas a eles. Você pode, por exemplo, procurar pela chamada ao *cmd.exe* que vários worms na selva da Internet emitem – e que infestam os logs do Apache. O patch evita que a chamada sequer chegue a seu servidor Web, sendo bloqueada no firewall.

A opção de definir regras especiais para cada uma das interfaces de rede é outro recurso inteligente que permite aprimorar o efeito das regras no desempenho global do firewall. Muitas opções são aplicadas a um objeto em particular por padrão, sem que o administrador tenha de definir explicitamente – um exemplo clássico é o bloqueio de pacotes com roteamento na origem (source routing). Você pode comparar essa funcionalidade com as regras implícitas do CheckPoint. Dependendo da plataforma de firewall usada, o produto livre pode até mesmo oferecer mais opções de configuração que o CheckPoint.

Conclusão

O Firewall Builder é uma escolha particularmente interessante para administradores que não precisam das opções avançadas dos produtos comerciais, ou não podem usá-los devido ao seu parque heterogêneo de plataformas de firewall.

O Firewall Builder permite que os administradores com pouca experiência em configuração de filtros de pacotes criem conjuntos de regras com um mínimo de esforço. O fato de sua configuração ser aplicada de forma transparente permite que se aprenda à medida que o firewall fica mais complexo. ■

Mais front-ends: GuardDog e KNetFilter

O GuardDog é um configurador de iptables construído com a biblioteca Qt. É muito interessante para usuários com pouca ou nenhuma experiência em redes e protocolos. Permite que o usuário defina zonas em sua rede e escolha os serviços permitidos para cada zona. O GuardDog implementa a maioria dos protocolos mais comuns – particularmente em ambiente doméstico.

Embora esse tipo de firewall pessoal possa ser uma boa idéia para computadores de mesa, mostra suas limitações quando precisa ser aplicado a cenários mais complexos. Felizmente, o padrão desse programa é bloquear tudo o que não for explicitamente permitido – na medida certa para usuários inexperientes, evitando que permitam acesso ilimitado de crackers à sua máquina. Para mais informações sobre o GuardDog, veja a matéria sobre ele nesta edição, à página 20.

O Knetfilter é outro front-end para o iptables que, além disso, usa o tcpdump e o Nmap para monitoramento. Também oferece recursos de QoS – Quality of Service (qualidade de serviço). O programa espera que o administrador defina todas as regras sem qualquer ajuda. Não há serviços pré-definidos. Aliás, os serviços são distinguidos apenas por suas portas de origem e destino. O administrador obrigatoriamente tem que saber os números de cabeça ou providenciar um papel com uma "colinha".

Se isso não for problema, o KNetfilter é capaz de criar regras extremamente granulares, levando em conta, inclusive, informações vindas de mensagens ICMP ou do estado das conexões. O KNetfilter não é, definitivamente, um brinquedo para iniciantes, pois requer conhecimentos mais ou menos profundos a respeito do iptables. Administradores experientes, entretanto, vão amar o poder de configuração e o nível extremo de detalhe que a ferramenta oferece, embora seja necessário um chope ou dois para se acostumar com a interface gráfica.

INFORMAÇÕES

[1] CheckPoint: *http://www.checkpoint.com/products/protect/firewall-1.html*

[2] Lista de discussão dos "Gurus" do FireWall-1: *http://www.phoneboy.com/staticpages/index.php?page=20030517034933897*

[3] Firewall Builder: *http://www.fwbuilder.org*

[4] iptables, filtro de pacotes do Linux: *http://www.netfilter.org*

[5] ipfilter, filtro de pacotes do Free/NetBSD: *http://coombs.anu.edu.au/~avalon/ip-filter.html*

[6] pf, filtro de pacotes do OpenBSD: *http://www.benzedrine.cx/pf.html*

[7] Compilador PIX para o Firewall Builder: *http://www.netcitadel.com/pix.htm*

[8] Patch para o iptables: *http://www.netfilter.org/documentation/pomlist/pom-extra.html#string*

[9] Anatomia de um firewall por estado de conexão: *http://www.giac.org/practical/gsec/Lisa_Senner_GSEC.pdf*

[10] Tabela de estados do Firewall-1: *http://www.spitzner.net/fwtable.html*

[11] Listas de discussão da Checkpoint: *http://msgs.securepoint.com/fw1/*

Proteção de Memória com o PaX e o Stack Smashing Protector

Pax Romana

Aqui está um patch para o kernel que vai trazer um pouco de paz de espírito aos usuários paranóicos no que diz respeito à segurança. Peter Busser, desenvolvedor do Adamantix, explica os princípios básicos do PaX (em latim, "paz") e justifica sua inclusão como um módulo de sua distribuição ultra-segura. POR PETER BUSSER

Muito tem se falado e escrito sobre segurança de sistemas. O assunto do momento são as maneiras de se limitar o que os processos podem fazer com outros objetos no sistema, como por exemplo arquivos, dispositivos e memória compartilhada. Com tal limitação, os especialistas esperam aperfeiçoar a integridade do sistema, a confidencialidade dos dados e a disponibilidade dos serviços. Prova cabal dessa preocupação é a enorme quantidade de *patches* para o kernel do Linux e mesmo de programas *user space* que existem para esse fim. Um belo exemplo é o RSBAC [1], usado pela distribuição Adamantix [2] para atingir esses objetivos.

Pouco se fala, entretanto, sobre a proteção dos únicos objetos realmente ativos em um sistema Linux: os próprios processos! Mais importantes que os arquivos, dispositivos e memória já citados, os processos desempenham um papel crucial no Linux: são o único lugar no sistema em que algum código pode ser realmente executado. A despeito dessa importância, o mundo Linux tem uma triste tradição de ignorar o problema da proteção do quinhão de memória usado pelos processos. Com isso, um número avassaladoramente alto de *exploits* utiliza-se da corrupção da memória (os famosos *buffer overflow*) dos processos para atingir seus objetivos malévolos.

Por que é tão importante proteger a memória?

Em um mundo cor-de-rosa, nenhum software seria infestado por "bugs" e, portanto, nenhum maldito invasor seria capaz de conseguir acesso de escrita e leitura na memória usada pelos processos. Há pessoas clamando a altos brados que, para ingressar nesse mundo perfeito, bastaria parar de desenvolver em C e começar a usar Java, C-LISP, Ada ou qualquer outra linguagem que considerem ser a melhor do universo. Como se a vida fosse simples assim...

A afirmação acima nunca foi comprovada, já que linguagens como Java, Perl e Python dependem de interpretadores escritos... na linguagem C! Mesmo que

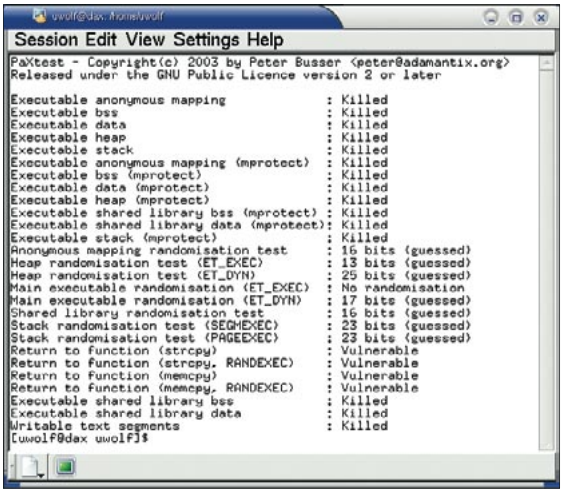


Figura 1: O *paxtest* verifica o funcionamento do PaX.

fosse verdadeira e começássemos a converter décadas de código em C para a novíssima, mais segura e sexy linguagem *XYZ*, anos e anos seriam necessários. Nesse meio tempo, os sistemas ainda dependeriam de programas escritos na insegura, vulnerável e "arcaica" linguagem C.

Nosso mundo não é perfeito e temos que aceitar o fato. Mesmo os melhores programadores cometem erros, e muitos deles podem possibilitar a corrupção da porção de memória interna usada pelos processos. Como os invasores externos sabem disso, o melhor a fazer é proteger nossa memória.

No início era o caos...

Há muitas razões para a atenção quase nula que os desenvolvedores dão à memória dos processos e sua proteção. A mais gritante é, infelizmente, a falta de compreensão da gravidade do problema. Tradicionalmente, as pesquisas sobre segurança voltaram-se quase que exclusivamente para o desenvolvimento de técnicas de controle de acesso e mecanismos de criptografia. A integridade dos processos não era um problema até que os estouros de buffer foram descobertos. Como tais falhas são mais populares a cada ano, o problema agora é assunto de Estado.

A recusa de Linus Torvalds de incluir no kernel o patch do projeto OpenWall, que impede a execução de código na pilha [3], foi um evento importante na história do Linux. Como o patch implementava apenas o controle da pilha, Linus argumentou que ele não era completo o suficiente, pois haviam outras porções de memória a proteger. O argumento era extremamente coerente, mas o que fizeram as outras pessoas em resposta a ele? Melhoraram o patch do OpenWall? Escreveram outros patches para proteger outras porções de memória? Não. Em vez disso, muitas pessoas começaram a considerar que **qualquer tipo** de proteção de memória era inútil. Isso é uma tolice tão grande quanto acreditar que a Terra é chata. Tristemente, é também uma tolice altamente disseminada e extremamente persistente.

Nos velhos e bons tempos em que todas as máquinas pequenas eram computadores pessoais, o assunto "segurança" não estava na ordem do dia. Desde que o computador não se conectasse a uma rede ou que as outras pessoas se mantivessem a uma distância segura de seu teclado ou mouse, o usuário podia se considerar a salvo. Muitos dos programadores de hoje cresceram nessa realidade. Entretanto, com o crescimento da Internet a situação mudou dramaticamente. Até então era fácil ignorar o problema e dizer às pessoas que não havia perigo. Hoje, fingir que o monstro não está embaixo da cama não é mais uma opção.

...e um dia veio a PaX!

Houve inúmeros esforços para criar patches que propiciassem melhor proteção de memória desde o fatídico dia em que Linus rejeitou a contribuição do projeto OpenWall. O site oficial do PaX [4] lista alguns deles. Muitos patches foram abandonados e não são mais mantidos, deixando o PaX como único espécime ainda vivo. Graças à persistência do autor do PaX, os usuários do Linux podem desfrutar da melhor proteção de memória do mundo livre.

O trabalho no PaX começou há mais ou menos trinta meses. Depois de examinar muitos "exploits", o autor do PaX chegou à conclusão de que a única maneira de neutralizá-los seria com uma eficaz proteção da memória usada pelos processos. Infelizmente não foi usado em larga escala até que o patch do *grsecurity* começou a incluir também o PaX. O *grsecurity* é bastante conhecido e, como muita gente o usa, repentinamente as atenções começaram a se voltar para o PaX.

Muitas distribuições, como por exemplo a versão segura do Gentoo ("hardened", ou "reforçada"), possuem o patch *gr-security* e, por conseguinte, o PaX. Isso criou uma demanda benéfica para a equipe de desenvolvimento: muitas pessoas começaram a pedir a inclusão de recursos, relatar falhas e mesmo contribuir para portar o PaX para outras plataformas. Onde havia apenas um desenvolvedor agora há uma pequena – e muito ativa! – comunidade que usa e desenvolve o software.

A diferença entre o PaX e os outros patches para proteção de memória é o fato de que ele não tenta prevenir exploits específicos. Em vez disso, tenta prevenir algumas classes de exploits. Como é que é? Classes? Do que diabos esse cara está falando?

Vou tentar explicar. Animais e plantas são categorizados tendo como referência suas similaridades. Há diversas classes de animais: mamíferos, aves, anfíbios, peixes... Suponha agora que alguém invente uma cerca que proteja suas terras contra javalis selvagens. Um expediente utilíssimo se temos uma lavoura freqüentemente vítima desses animais. Entretanto, a mesma cerca será inútil contra um estouro de elefantes ou mesmo pequenos roedores. Em vez disso, uma cerca que protegesse contra TODOS os animas terrestres seria, essa sim, muito superior.

O mesmo se aplica aos exploits e ataques. No mundo do software livre, o PaX é como essa supercerca, protegendo o sistema contra toda uma classe de exploits. Em outras palavras, proteção contra TODOS os animais terrestres. Infelizmente, a cerca ainda deixa passar pássaros e insetos. Precisaríamos, então, de outros softwares para trabalhar ao lado do PaX e proteger seu sistema contra os outros tipos de animais. É triste constatar, entretanto, que os poucos softwares similares que existem atualmente não dão conta do recado. Podemos dizer, por exemplo, que o patch do OpenWall protege apenas contra os roedores. Os projetos *W^X* [5] e *exec-shield* [6] do OpenBSD prote-

gem contra todos os animais terrestres, exceto elefantes. Confuso? Falaremos mais sobre isso logo adiante.

Classes de ataque

De modo geral, há três classes de ataques que os patches de proteção de memória tentam evitar:

(1) Injeção e execução arbitrária de código.
(2) Execução de código já existente mas fora da ordem original em que estava no programa.
(3) Execução de código já existente no programa e na ordem original, mas com dados arbitrários.

Cada tipo possível de corrupção de memória pertence a uma dessas três classes. Por exemplo, muitas falhas populares usando estouros de pilha pertencem a (1). O exemplo citado por Linus Torvalds, usando a técnica return-to-libc, pertence a (2). Exploits pertencentes à classe (3) são raros, mas existem. Normalmente é mais fácil usar as classes (1) e (2). Observe que essa é uma classificação de técnicas de exploração de falhas, não das próprias falhas. Ou seja, uma mesma técnica pode ser usada para explorar diferentes bugs, e uma mesma falha pode ser explorada de mais de uma forma.

A idéia por trás do PaX é fazer com que classes inteiras de técnicas de exploração parem de funcionar. Até o momento, o PaX conseguiu parar ataques da classe (1) – ou seja, é uma cerca contra todos os animais terrestres. Os exploits da classe (2) estão com os dias contados, pois suas técnicas estão sendo discutidas. A classe (3) será tratada em algum ponto do futuro, mas as pesquisas ainda não estão concluídas. O que destaca o PaX dos outros patches de proteção de memória é o fato de lidar com a classe (1) inteira, além de não ignorar (2) e (3).

Classe (1)

Injeção e execução de código arbitrário significa que é possível:

- Sobrescrever código que já está na memória da máquina;
- Sobrescrever dados que já estejam na memória e executá-los como se fossem código;
- Carregar código do disco para a memória e executá-lo.

Se é possível sobrescrever código, um invasor pode injetar o seu próprio código, invariavelmente malicioso, num processo vulnerável e fazer com que o próprio processo execute esse código. Em vez de fazer o que o programa foi projetado para fazer, o processo faz o que o invasor deseja que ele faça.

O mesmo vale para a segunda técnica. Dados do programa, guardados numa porção da memória, são sobrescritos por dados injetados pelo invasor e executados como se fossem código. Você pode pensar que o sistema deveria separar o que é código e o que são dados, mas os programadores preferem a facilidade de tratar tudo do mesmo jeito. Como resultado, essa técnica é usada pela maioria dos ataques de estouro de buffer. O PaX assegura que código seja código e dados sejam dados, permitindo que possamos ler e gravar a memória onde os dados estão, mas nunca executar código nela. De forma semelhante, o patch permite ler e executar código na área de memória apropriada, mas *nunca* escrever nada por lá.

As duas primeiras técnicas requerem apenas acesso de escrita e execução na memória. O PaX lida com essas técnicas à sua maneira. A terceira técnica é diferente porque requer, além de operações com a memória, acesso a arquivos no disco. A eletricidade é muitas vezes usada para acentuar a eficiência de cercas. O PaX faz algo semelhante ao influenciar ACLs e outros mecanismos de acesso como o RSBAC [1]. Isso garante uma proteção perfeita contra qualquer ataque de classe (1). É o único patch de proteção de memória que vem com uma garantia desse tipo.

Classe (2)

Lembram do exemplo que Linus Torvalds deu para rejeitar o patch do projeto OpenWall? Pois é, aquele era um ataque de classe (2). Explicando de forma simples, num ataque de classe (2) o invasor sobrescreve com novos dados um endereço de memória usado para controlar a maneira como o processo trabalha. Por exemplo, a alteraração do valor do ponteiro de retorno da pilha faz com que, quando a função chamada devolve o controle ao programa que a chamou, o processamento seja desviado não para o lugar certo no programa que está sob ataque, mas para outra posição de memória qualquer à escolha do cracker. Geralmente, nessa posição de memória já se encontra algum código malicioso, cuidadosamente colocado lá pelo invasor antes dele provocar o "estouro" da pilha.

Esse é o apenas exemplo mais comum, mas há inúmeros outros lugares em que algum tipo de endereço é armazenado. Cada um deles, obviamente, é usado para uma finalidade específica dentro do programa que se está atacando e,

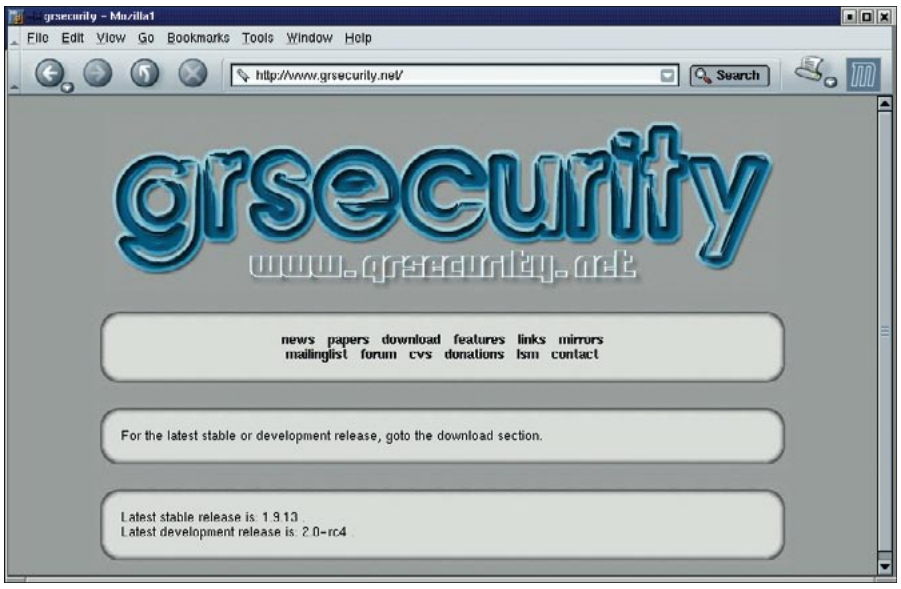


Figura 2: O *grsecurity* é uma outra alternativa em termos de proteção de sistemas Linux.

pelo menos em teoria, todos eles podem ser usados por invasores para influenciar o comportamento dos processos.

Classe (3)

Nessa classe, dados importantes são alterados durante o ataque. Nos desenhos animados, uma piada recorrente mostra o mocinho pondo um disfarce qualquer e fazendo o bandido pular num precipício ou correr na direção errada. Por incrível que pareça, os programas em geral são, via de regra, tão burros que é possível atacá-los usando o mesmo velho truque sujo. Caso um invasor possa gravar novos dados por cima de informações importantes, o programa pode ser levado a acreditar que o caminho certo a seguir é o caminho da perdição, trilhando uma lógica completamente diferente e fazendo coisas inesperadas.

Peguemos como exemplo o comando *mount*. Podemos configurá-lo para permitir que certos usuários possam montar discos. O programa faz algumas verificações para discernir entre usuários autorizados e não autorizados, montando os volumes de acordo com as permissões encontradas.

Se um invasor puder de alguma forma influenciar essas verificações, o comando mount pode acreditar que o atacante é um usuário autorizado e montar um disco para ele – ou seja, mais uma base para disparar novos ataques. Esse exemplo foi puramente hipotético. Exemplos reais são dificílimos de encontrar, pois falhas desse tipo são muito difíceis de se descobrir, que dirá de explorar.

Vista todas as suas armaduras

Proteger-se contra uma única classe de ataques não é lá uma atitude muito sábia. Linus usou um ataque de Classe (2) para driblar a proteção contra ataques de Classe (1) proporcionada pelo patch do projeto OpenWall. Isso é válido para qualquer classe: se o sistema protege contra um tipo, basta usar o outro!

As pessoas em geral (e mesmo desenvolvedores experientes) concentram-se em apenas uma classe de ataques, esquecendo-se das demais – foi o que o pessoal do projeto OpenWall fez. Pior do que isso, as pessoas costumam ignorar a utilidade de se proteger contra uma classe de ataques simplesmente porque precisam instalar outros mecanismos para se proteger de ataques em outras classes. Esse foi o erro cometido por Linus Torvalds. Em vez de esperar por uma ferramenta abrangente, o correto seria combinar diferentes mecanismos de proteção para cobrir todos os casos (e classes) possíveis. Não há isso de *tamanho único* em Unix. A abordagem correta é, e sempre será, a de ter várias ferramentas trabalhando juntas.

Stack Smashing Protector

A combinação de diferentes mecanismos de defesa é a razão pela qual o Stack Smashing Protector (SSP) foi agregado ao Adamantix. O SSP [7] (ou *ProPolice*) é um patch para o GCC que toma providências para evitar que alguns tipos de falhas das Classes (1), (2), e (3) possam ser exploradas. Mais precisamente, ele cria defesas contra os chamados *estouros de pilha* (*stack overflows*).

O SSP usa dois mecanismos para chegar a esse objetivo:

- Posiciona uma espécie de "mina terrestre" na pilha quando detecta uma função potencialmente perigosa. O mecanismo de detecção não é à prova de imbecis – tanto pode detonar a "mina" sem perigo como pode falhar e deixar a pilha ser estourada;
- Muda a ordem das variáveis locais, deixando as mais perigosas o mais perto possível da "mina". Isso aumenta muito a possibilidade de detecção.

O termo "mina terrestre", embora seja uma boa alegoria para o mecanismo, não virou jargão. A palavra mais conhecida no meio técnico é "canário" (*canary*), em referência aos passarinhos usados por mineiros de carvão para detectar monóxido de carbono (CO) em níveis letais. O gás mata os passarinhos muito mais rápido do que mataria os trabalhadores, dando a eles tempo de fugir. Na pilha, o "canário" é um número aleatório colocado em um ponto estratégico. Um ataque por estouro de pilha certamente alteraria esse número, evento que seria detectado pelo SSP antes que o código malicioso do invasor pudesse ser executado. O SSP então envia uma mensagem ao registro de eventos do sistema (syslog) e interrompe a execução do programa atacado.

Esse tipo de verificação é bastante dispendioso em termos de memória e tempo de CPU, especialmente quando funções muito pequenas são usadas. O SSP tenta detectar funções que possam ser vulneráveis a estouros de pilha e adiciona as rotinas de verificação apenas a elas. Como o mecanismo de detecção não é perfeito, é bem provável que o SSP deixe de injetar as rotinas em funções que delas precisem ou as inclua em lugares desnecessários. Às vezes, ambas as coisas.

O SSP também pode ser usado para compilar o kernel. Sempre é uma boa idéia incluir uma camada de proteção em volta do núcleo do sistema. Quanto ao desempenho, uma surpresa: as otimizações do SSP não deixam o kernel perceptivelmente mais pesado!

Existem técnicas mais elaboradas para fazer com que programas escritos na linguagem C sejam mais seguros.

Comparação: PaX versus outros patches.

A melhor coisa a respeito do *paxtest* é o fato de que foi possível usá-lo para testar a proteção de memória proporcionada pelos *outros* patches. Baixe o paxtest do site oficial do PaX [4] e compile-o em seu sistema. Se estiver usando Adamantix, basta um *apt-get install paxtest*. Disparar o paxtest contra um kernel "remendado" com o *OpenWall* mostra que apenas a pilha é protegida contra execução de código – coisa que, aliás, é o esperado. Em outras palavras, isso significa que quase não há proteção.

Quando o paxtest é rodado contra o *exec-shield*, obtemos diferentes resultados, dependendo da versão do paxtest em uso. Algumas falhas em versões antigas do paxtest fizeram com que as pessoas acreditassem que o *exec-shield* oferece uma proteção maior do que realmente o faz.

Na realidade, a proteção que o *exec-shield* oferece é realmente impressionante – só que para menos. Será interessante acompanhar a agitação dos desenvolvedores de exploits e ver quão rápido se adaptarão às fraquezas do *exec-shield*. Seria divertido se alguém portasse o paxtest para o OpenBSD. Não acredito que o i386 W^X – tão festejado pela comunidade OpenBSD – mostre resultados convincentes.

Uma delas é a famosa verificação de limites (*bounds checking*). Essa verificação significa que o acesso a *todo e qualquer dado* é testado. Pelo lado bom, isso traz mais segurança para qualquer programa. A desvantagem: o programa fica **pesado**. A velocidade é comparável à do Java – que, por acaso, faz verificação de limites por padrão. Essa queda brutal no desempenho é algo com que, na vida real, poucas pessoas querem (ou podem) conviver.

Aleatorização

Um recurso encontrado no PaX e em outros patches de proteção de memória é o chamado ASLR – "Address Space Layout Randomization", algo como *disposição aleatória do espaço de endereços*. Com o ASLR, diferentes partes do programa são gravadas em locais diferentes da memória. A posição de cada porção do programa na RAM muda a cada vez que ele é executado.

Isso não é um mecanismo de proteção, pois não há pontos de controle. Entretanto, dificulta bastante uma tentativa de exploração de uma falha conhecida, uma vez que o atacante nunca pode ter certeza da exata localização das coisas na memória. Os diversos patches de proteção de memória diferem na quantidade de entropia usada. Normalmente, quanto mais melhor, pois faz com que ataques de força bruta (*brute force*) sejam gradativamente mais difíceis. Atualmente, o PaX oferece a maior aleatoriedade dentre todos os patches de proteção de memória para Linux. Aliás, melhor até do que a do OpenBSD.

Compatibilidade

Não é necessário recompilar todos os seus programas para que eles funcionem num kernel com o PaX. A maioria deles rodará bem sem qualquer modificação. Já as bibliotecas podem causar algum estresse. Por experiência própria, sei que o Debian Woody possui algumas "pegadinhas" nessa área. Poucas bibliotecas são afetadas, mas algumas delas são extremamente importantes – a zlib é um belo exemplo. Conta-se que versões antigas do Red Hat possuem muitos problemas de compatibilidade com bibliotecas, embora eu nada tenha ouvido sobre as versões mais novas. Fiz testes preliminares com o Debian Sarge e, exceto pelo servidor gráfico (*XFree86*), tudo funcionou sem problemas com o kernel com PaX.

A maioria dos programas funciona sem nenhum problema com o PaX, mesmo que esteja configurado para ser bastante restritivo – como é o caso do Adamantix, por exemplo. Das várias centenas de pacotes já adaptados para uso no Adamantix, apenas uns poucos causam problemas com o PaX. Em muitos deles, o conserto foi bem fácil. As alterações mais comuns são de *flags* de compilação, especialmente para bibliotecas. Em alguns casos precisamos usar código em C em vez de assembly (por exemplo, na zlib e no gnupg). Em outros, algumas linhas de código tiveram que ser reescritas para se adequar ao método de trabalho do compilador.

Alguns programas – poucos, na verdade – não puderam ser adaptados. Todos eles precisam, por definição, criar código executável diretamente na memória. Um bom exemplo é o ambiente de Java da Sun, o SUN Java Runtime Environment (JRE).

Quando isso acontece, é possível usar o comando *chpax* para definir exceções. As configurações do *chpax* são gravadas no interior do próprio executável. Quando o executável é invocado, o PaX detecta essas configurações e desativa algumas das verificações. Um software problemático pode, então, ser colocado em funcionamento sem que seja necessário mudar uma única linha de código.

O duro teste da paz

Quando o PaX foi incluído no kernel do Adamantix e eu passei a recompilar os programas para o ASLR do sistema, tudo funcionou tão maravilhosamente bem que eu comecei a duvidar da honestidade dos desenvolvedores. Antes de começar, eu já estava preparado para encarar um grande número de programas deixando de funcionar. Nada disso pareceu ocorrer. Como isso era possível? Uma das alternativas era a de que o PaX estava inoperante, ou estava realmente ativado mas não fazia nada de nada. Em vez de especular, decidi encontrar alguma prova de (não) funcionamento. Para isso, desenvolvi o *paxtest*.

O *paxtest* é uma pequena coleção de programas de teste. Cada um deles verifica um aspecto funcional do PaX. Um teste escreve código de máquina em uma variável do tipo *string* e tenta executá-lo. Um PaX funcionando bem detectaria a tentativa e mataria o processo imediatamente. Outros criavam diversas situações vulneráveis, cada uma relativa a um aspecto que o PaX deveria proteger, e tentavam explorá-las.

Há também testes que verificam a profundidade de aleatorização do ASLR. Juntos, todos os programas nos dão uma idéia do nível de proteção oferecido pelo PaX. Quanto mais testes informam "killed" (ou seja, processo interrompido), melhor. Usando os dados obtidos com o *paxtest*, pude comprovar que o PaX estava se dando muito bem com o kernel do Adamantix. Melhor que isso: a ausência de programas indicava que o PaX estava funcionando muito melhor do que o esperado.

Conclusão

A proteção da memória usada pelos processos é assunto da mais alta importância, mas até agora não conseguimos encontrar nada que nos proteja contra as três classes de ataques ao mesmo tempo. O PaX, desde seu surgimento, nunca foi superado por outro patch de proteção de memória. Com o auxílio de controle de acesso (ACLs) e outros mecanismos auxiliares, o PaX consegue garantir proteção perfeita contra ataques de Classe (1).

Outros mecanismos, como o SSP, são necessários para evitar alguns ataques nas Classes (2) e (3). O custo de implementação é relativamente baixo. Qualquer distribuição de Linux que se importa com aspectos de segurança tem a obrigação de planejar a inclusão desse tipo de proteção. ■

INFORMAÇÕES

[1] *http://www.rsbac.org/*

[2] *http://www.adamantix.org/*

[3] *http://old.lwn.net/1998/0806/a/linus-noexec.html*

[4] *http://pageexec.virtualave.net/*

[5] *http://archives.neohapsis.com/archives/openbsd/2003-04/1362.html*

[6] *http://people.redhat.com/mingo/exec-shield/*

[7] *http://www.research.ibm.com/trl/projects/security/ssp/*

Compilando software para várias plataformas

Produto de exportação

Apesar de ser um conceito incomum para a maior parte das pessoas, aqueles que desenvolvem para outras plataformas rapidamente sentem uma necessidade de bons ambientes e suporte para cross-compiling. POR PITER PUNK

Não sou o que se pode chamar de "desenvolvedor multiplataforma", mas já dei meus pulinhos, compilando um kernel para PlayStation (desenvolvido por uma empresa que não existe mais, a *runix.ru*), um para ARM, outro para PowerPC e uma meia dúzia de softwares para Athlon64.

Se eu tenho todas essas máquinas? Não, não tenho. E, mesmo que tivesse, como eu iria compilar o kernel do PlayStation no próprio PlayStation, que não tem sequer linha de comando, quanto mais um compilador?

É nessas horas que usamos um "cross-compiler", ou *compilador cruzado*. Fazer uma compilação cruzada nada mais é que compilar programas de uma plataforma em outra. Aproveitando o exemplo que citei acima, podemos compilar um kernel para MIPS R3000A (processador do PlayStation 1) dentro do meu PC, que usa um processador x86.

Outro uso típico da compilação cruzada é em máquinas com baixa capacidade de processamento. Por exemplo, um PowerMac 5500 (com um processador PowerPC 603e a 250 MHz) demora cerca de uma hora para compilar um kernel da série 2.4, mas em menos de 20 minutos meu Athlon já "velhinho" resolve o problema. Ou seja, apesar de gastar um tempo inicial configurando o Athlon para geração de binários para máquinas PowerPC, terei uma grande economia de tempo depois. Em alguns casos, o dispositivo alvo sequer tem memória suficiente para compilar coisa alguma, como o PlayStation com seus parcos 2MB de RAM.

OK, você me convenceu...

Agora que já consegui convencê-lo a usar um cross-compiler, vamos às más notícias:

1. O processo envolve aproximadamente meia dúzia de passos exóticos;
2. Não funciona sempre, estando condicionado a diversos fatores externos, incluindo as fases da lua, o humor do presidente do Banco Central e a flutuação no valor de títulos voláteis da dívida externa do Zimbábue.

Quanto à primeira má notícia, a solução é razoavelmente simples: basta montar um "algoritmo" com os tais passos e seguilos sempre que necessário. A segunda má notícia envolve bem mais trabalho, incluindo buscas em listas de discussão, no Google, em ChangeLogs, em documentos obscuros na Biblioteca Nacional e até mesmo o sacrifício de virgens.

A primeira coisa que você vai precisar é baixar e instalar os softwares necessários para a compilação. Basicamente você vai precisar das *binutils*, o código fonte do kernel e o *gcc*, que são instalados por padrão na maioria das distribuições Linux.

Como estudo de caso, realizaremos a compilação de um binário para a arquitetura PowerPC em uma máquina x86. Isso irá determinar as versões de cada software que serão utilizadas (bem como os patches necessários). Sim, graças à segunda má notícia, não são todas as versões de software que funcionam com todas as plataformas de destino. Aliás, nem mesmo é possível a compilação cruzada entre certas plataformas – cada caso é um caso.

...mãos à obra!

Vale lembrar que todos os softwares utilizados são extremamente temperamentais. Vejamos os procedimentos básicos e dicas do que você pode precisar se quiser compilar programas para outras arquiteturas.

Utilizamos neste artigo o *binutils-2.15.90.0.3*, o *gcc-3.3.4*, a *glibc-2.3.2* e o kernel *2.4.18*. O código fonte dos três primeiros foi retirado do Slackware 10.0 e o kernel veio do Slackintosh 8.1. Você pode tentar com versões diferentes, mas é uma questão de "sorte".

Preparando o terreno

Descompacte o kernel que você baixou. Uma boa dica é remover o link */usr/src/linux* e descompactar seu kernel no */usr/src*. A seguir, entre no diretório do kernel e execute:

```
# make ARCH=$ARQUITETURA symlinks ➥
include/linux/version.h
```

Esse comando irá corrigir os links que estão dentro do */usr/src/linux/include* e apontá-los para a arquitetura correta. No meu caso, eu fiz:

```
# make ARCH=ppc symlinks ➥
include/linux/version.h
```

Com isso, o link */usr/src/linux/include/asm* que, originalmente, apontava para o *asm-i386* (instruções em assembler do x86) passa a apontar para o asm-ppc, com as instruções em assembler para o PowerPC.

O binutils

Agora começa a diversão. O binutils gera vários programas importantes para a compilação. Entre eles, dois essenciais:

- *as* – o assembler, que pega a saída do gcc e a converte para instruções da máquina; em seguida, as passa para o linker.
- *ld* – a função do linker é juntar vários arquivos objeto em um só, o que possibilita dividir programas extensos em pequenos pedaços e juntá-los no fim do processo em um único executável.

O *as* e o *ld* são específicos para cada arquitetura. Ou seja, o *as* para x86 não pode ser usado para gerar binários de PowerPC e vice-versa. Por isso, vamos compilá-los (junto com outros programas importantes) para ser executados em x86 e gerar código PowerPC.

Antes de descompactar o binutils, vamos prezar um pouco a organização do sistema e criar um diretório */opt/ppc*, para não misturar os nossos binários com os do sistema. Após criar esse diretório, descompacte o código-fonte do binutils, entre no diretório criado e digite:

```
# ./configure --prefix=/opt/ppc ➥
--target=powerpc-linux ➥
--disable-nls
```

Esse comando configura o binutils para ser instalado no diretório */opt/ppc*, tendo como plataforma alvo uma máquina PowerPC rodando Linux. O *--disable-nls* avisa para não compilar o suporte para outras linguagens além do inglês. Depois de tudo configurado, vamos aos já clássicos:

```
# make
# make install
```

Com isso podemos já compilar programas em assembly do PowerPC. Como a maior parte dos programas está em C, é uma boa idéia ter um compilador C, como por exemplo o gcc. Ah! Antes de passar para o próximo passo, inclua o diretório */opt/ppc/bin* no seu *PATH*, para que possamos utilizar os comandos que acabamos de compilar:

```
export PATH=$PATH:/opt/ppc/bin
```

O compilador C

Este é o momento de compilar o gcc. Na teoria é extremamente simples. A seguinte linha de comando faz a configuração correta:

```
# ./configure --target=powerpc-➥
linux --prefix=/opt/ppc ➥
--disable-nls --disable-threads ➥
--disable-shared --enable-➥
languages=c --with-newlib
```

As opções *--target*, *--prefix* e *--disable-nls* já são nossas conhecidas de quando compilamos o binutils. Como novidades temos o *--disable-threads* e o *--disable-shared*, que desabilitam, respectivamente, o suporte a threads (que depende da glibc, que nós ainda não temos) e a bibliotecas compartilhadas (idem).

O *--enable-languages* diz quais das linguagens do gcc iremos compilar. No nosso caso, pelo menos nesse primeiro momento, compilaremos apenas o C, não tendo motivo para dar suporte a outras linguagens. O último argumento faz com que o gcc não use a glibc (que nós ainda não temos). Depois de tudo isso, poderíamos terminar o assunto com um:

```
# make
# make install
```

Não é? Pode até ser que para outras arquiteturas isso seja verdade, mas para o PowerPC é necessário utilizar um patch. E, sim, é muito comum ter que usar patches diversos: lembre-se da má notícia número 2; ela nos perseguirá ainda por bastante tempo. No nosso caso, devemos aplicar o patch *gcc-3.3.1-crosspc.diff*, encontrado em [1]. Para aplicar o patch basta digitar:

```
# patch -p1 < gcc-3.3.1-➥
crosspc.diff
```

Depois disso, podemos rodar novamente o *make* e o *make install*. Dessa vez, tudo vai correr bem e teremos nosso compilador C funcionando.

Um bom teste para ele é compilar o kernel do Linux, que é complexo e não usa a glibc. A compilação é bem simples: o primeiro passo é editar o arquivo */usr/src/linux/Makefile*. Comente a linha que começa com *ARCH=$(sh...)* e inclua uma linha com o seguinte conteúdo:

```
ARCH=ppc
```

Ah! Os nomes utilizados na compilação de outros programas nem sempre são os mesmos nomes utilizados no kernel, como vimos agora com *powerpc* e *ppc*. Também devemos editar a linha com:

```
CROSS_COMPILE =
```

E colocar o parâmetro powerpc-linux- (é, com o "-" no final) após o sinal de igualdade. Agora siga a ordem normal de compilação:

```
# make menuconfig
# make dep
# make clean
# make vmlinuz
```

Opa! Essa última linha não era para ser *make bzImage*? Bom, uma das coisas que temos que fazer quando pensamos em compilar para outras arquiteturas é tentar se livrar dos vícios x86. O *zImage* e o *bzImage* só existem porque não é possível ler diretamente arquivos de um certo tamanho no PC. A solução encontrada foi compactar o arquivo e fazer com que ele fosse descompactado durante a inicialização. Daí vêm os arquivos zImage (*Zipped Image*) e o bzImage (*Big Zipped Image*).

No PowerPC, e na maioria das outras arquiteturas, geramos diretamente a imagem do kernel, o arquivo *vmlinux*. Podemos verificar se o binário foi gerado corretamente utilizando o comando "file" para ver o tipo do arquivo:

```
# file vmlinux
vmlinux: ELF 32-bit MSB executable,➥
PowerPC or cisco 4500, version➥
1 (SYSV), statically linked,➥
not stripped.
```

Pois bem: se quisermos compilar apenas o kernel, terminamos por aqui o nosso toolkit para compilação cruzada. Porém, nem só de kernel vive o homem, mas de todo software que puder ser compilado...

Compilando a glibc

Sim, se quisermos compilar outros softwares além do kernel, vamos precisar de uma libc. A mais utilizada no mundo Linux é a GNU libc, conhecida como *glibc*. Usei o código-fonte da *glibc-*

2.3.2, encontrado nos CDs do Slackware 10. Cuidado: o procedimento de compilação dela não é tão trivial.

Primeiro, devemos descompactar o código (arquivo *glibc-2.3.2.tar.bz2*); em seguida, entramos no diretório gerado (*glibc-2.3.2*). Lá dentro, devemos descompactar o código-fonte da linuxthreads (*glibc-linuxthreads-2.3.2.tar.bz2*):

```
# tar -xvjf ../glibc-➥
linuxthreads-2.3.2.tar.bz2
```

Ainda no diretório *glibc-2.3.2*, aplicamos um patch para possibilitar a compilação da glibc com o gcc 3.3.x:

```
# zcat ../glibc.gcc33x.diff.gz ➥
| patch -p1
```

Agora, vamos começar a trabalhar. Saia do diretório *glibc-2.3.2* e crie outro chamado "build" (ou o nome que você achar melhor). Entre nesse diretório e vamos configurar a glibc:

```
# ../glibc-2.3.2/configure ➥
--prefix=/opt/ppc --target=➥
powerpc-linux --host=powerpc-➥
linux --enable-add-ons=➥
linuxthreads --with-headers=➥
/usr/src/linux/include ➥
--with-binutils=/opt/ppc/bin
```

Os parâmetros significam:

--enable-add-ons=linuxthreads, que adiciona e faz com que seja compilado o suporte a threads;
--with-headers, que aponta para onde estão os includes do kernel que estamos utilizando; e
--with-binutils, que avisa onde estão instalados nossos binutils.

Depois de configurada a glibc, podemos compilá-la: *make all*. De preferência faça isso pouco antes de dormir, ou quando estiver passando um filme bom na TV, já que o processo demora bastante. Quando tudo tiver terminado, digite: *make install*.

Por algum motivo, a glibc instala vários dos seus arquivos em */opt/ppc/lib* e o gcc os procura em */opt/ppc/powerpc-linux/lib*. Resolvi isso copiando todos os arquivos de */opt/ppc/lib* para o */opt/ppc/powerpc-linux/lib*.

Aproveitando que estamos copiando coisas, vamos copiar os includes do kernel para o */opt/ppc/include*. Vamos fazer isso porque os programas devem ser compilados com os mesmos includes de kernel com que foi compilada a glibc, e nada melhor para garantir isso que copiar esses arquivos agora, logo depois de criarmos a glibc.

```
# cd /opt/ppc/include
# cp -a /usr/src/linux/include/➥
linux .
# cp -a /usr/src/linux/include/➥
asm-generic .
# cp -a /usr/src/linux/include/➥
asm-ppc asm
```

Repeteco

O gcc completo usa a glibc. Enquanto não havia a glibc, não era possível compilar o gcc com ela. Por isso usamos o parâmetro *--with-newlib* na configuração do gcc. Agora que temos uma glibc, podemos também ter um gcc completo.

Entre de novo no diretório onde você descompactou o gcc, e digite:

```
# ./configure --target=powerpc-➥
linux --prefix=/opt/ppc --enable-➥
shared --enable-threads ➥
--enable-languages=c
# make all
# make install
```

O gcc será compilado, mas agora com suporte a threads, bibliotecas compartilhadas e o que mais você achar interessante. Agora você também pode compilar outras linguagens além de C e fazer um sistema de desenvolvimento completo! Atingimos nosso objetivo!

Testando...

Depois de tudo pronto, o ideal é testar com um programa:

```
/* hello.c */
#include <stdio.h>
int main() {
  printf("Hello World!");
  return(0);
}
```

Esse programa imprime na tela *"Hello World!"* (sim, não estou sendo nem um pouco original). Antes de compilar, vamos exportar as seguintes variáveis:

```
# export C_INCLUDE_PATH=/opt/➥
ppc/include
# export CC=powerpc-linux-gcc
```

O primeiro comando indica onde devem ser encontrados os headers do sistema; afinal, queremos que o compilador use aqueles que nós criamos para PowerPC. Acredite, a falta dessa linha leva a desespero e noites mal dormidas. O segundo comando serve apenas para facilitar a vida se formos compilar várias coisas. A maioria dos programas e Makefiles já reconhece a variável CC e por isso usa o compilador indicado. O teste é simples:

```
# $CC -Wall hello.c
```

Ele vai dar algumas mensagens de erro, reclamando que o "main" não pode ser "void" (e não pode mesmo), mas vai compilar. Agora basta checar se o arquivo gerado é para PowerPC, o que pode ser feito com o comando "file", assim como fizemos com o kernel (*vmlinux*) que compilamos anteriormente.

Para garantir que nosso sistema funciona com programas "de verdade" vamos compilar um software qualquer. Escolhi o tar e peguei os sources do slackware 10.0. Vamos descompactar o arquivo *tar-1.14.tar.bz2* e compilá-lo:

```
# tar -xvjf tar-1.14.tar.bz2
# cd tar-1.14
# ./configure --host=powerpc-➥
linux --target=powerpc-linux ➥
--prefix=$HOME/ppc
# make
# make install
```

Para confirmar, novamente usamos o comando file. O resultado deve ser:

```
ELF 32-bit MSB executable, ➥
PowerPC or Cisco 4500, version ➥
1 (SYSV), dynamically linked➥
(uses shared libs), not stripped
```

Missão cumprida. Agora divirta-se compilando seus softwares favoritos para rodar numa torradeira, lavadora ou no microondas. ■

INFORMAÇÕES

[1] *http://www.openslack.org/~piterpk/cross/gcc-3.3.1-crossppc.diff*

linuxUSER

Bem-vindo à LinuxUser!

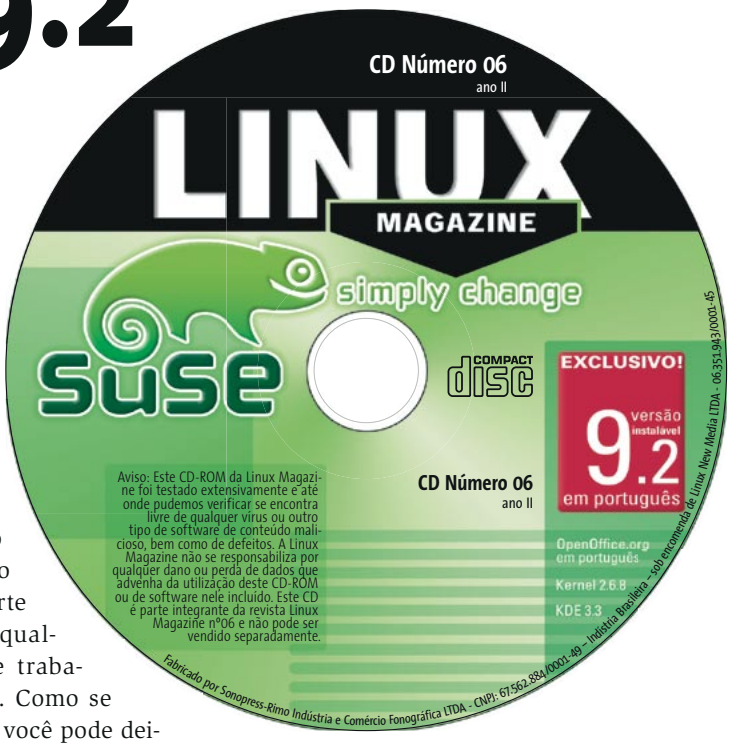
Esta é uma seção especial dedicada a destacar programas úteis e interessantes para ajudá-lo no seu trabalho diário com o Linux no desktop. Aqui você encontrará informações sobre como utilizar programas comuns de forma mais eficiente, obterá um valioso embasamento técnico e conhecerá as últimas novidades em software para seu sistema operacional favorito.

O camaleão esbanja tropicalidade

SuSE Linux 9.2

Fruto do trabalho conjunto da SuSE Linux AG e da Linux Magazine, este mês temos o prazer de lhe oferecer um CD com a versão nacionalizada de uma das distribuições mais cobiçadas do momento: o SuSE Linux 9.2.

POR RAFAEL PEREGRINO DA SILVA



A idéia nasceu durante uma conversa entre alguns dos diretores da SuSE Linux AG, da Linux New Media AG e da Linux New Media do Brasil, por ocasião da Linux World Conference and Expo em Frankfurt, em outubro de 2004: "Por que não criar uma versão para o Brasil do SuSE Linux, que coubesse em um único CD?" O resultado chega a suas mãos com esta edição.

Versão atual do sistema, o SuSE Linux 9.2 tem muito a oferecer: com um sistema desktop totalmente integrado e de interface consistente, bem como uma das instalações mais fáceis e poderosas do mercado, o produto da Novell é um forte competidor para qualquer ambiente de trabalho da atualidade. Como se isso não bastasse, você pode deixar seu dicionário de lado quando for usar o CD: tanto a instalação quanto o conjunto de pacotes pré-instalado – que conta, entre outros, com o kernel Linux 2.6.8, o X.org 6.8.1, o KDE 3.3, o OpenOffice.org 1.1.3 e uma série de aplicativos que oferecem tudo aquilo de que você precisa – estão em português.

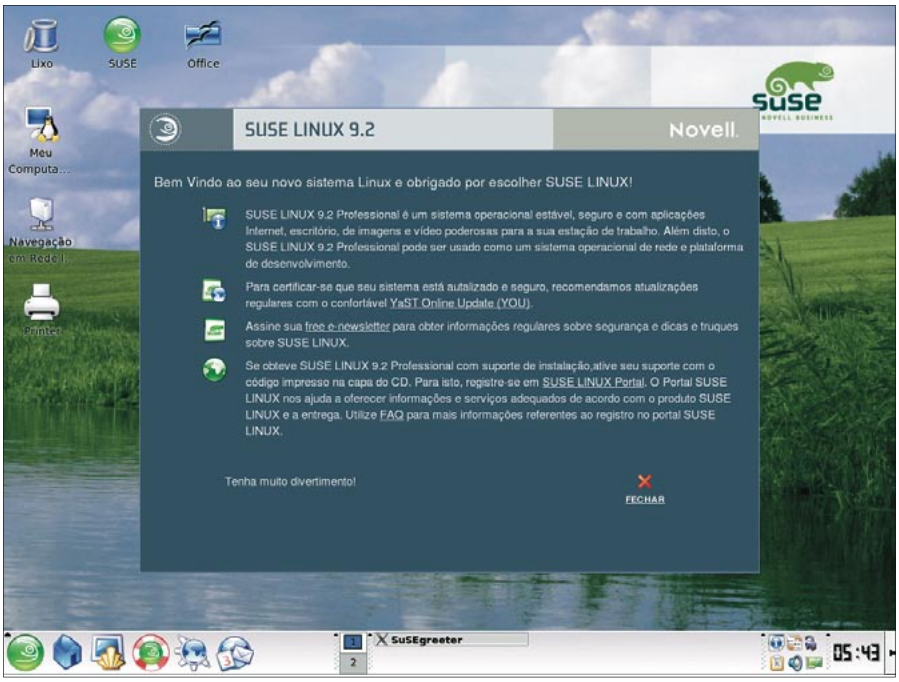


Figura 1: SuSE Linux a seu dispor! Esperamos que você aprecie a viagem...

Instalação

A instalação não poderia ser mais simples: introduza o CD no drive apropriado, inicie seu computador e siga as instruções na tela. O YaST, instalador da SuSE, é totalmente gráfico e inteligente. Não fique com receio se houver um Windows® ou outro sistema qualquer instalado no seu computador: o YaST vai lhe perguntar se você deseja mantê-lo ou utilizar todo o disco rígido para a instalação. Caso deseje mantê-lo – e se houver espaço, é claro – a rotina de instalação do sistema o guiará de maneira simples e intuitiva através das tarefas de redimensionamento e particionamento do seu disco. Naturalmente, aconselhamos realizar um backup de seus dados antes de prosseguir. Mas ao final da instalação dificilmente haverá surpresas indesejáveis: seus sistemas antigos estarão lá, prontinhos para ser iniciados, ao lado do novo sistema.

Todo o reconhecimento de hardware ocorre automaticamente. Além disso, todas as impressoras locais ou disponíveis via rede deverão estar listadas e prontas para ser usadas por todos os aplicativos do sistema. O mais interessante é que, ao final da instalação, não

Tabela 1: Principais aplicativos pré-instalados

Função	Aplicativos
Gráficos	OpenOffice Draw, GIMP
Internet	Konqueror, Kopete
Gerenciador de Informações Pessoais	Kontact (composto pelo KMail, KOrganizer, KAddressBook, KNotes e KNode)
Multimídia	Kaffeine, K3B, JuK e amaroK
Escritório	OpenOffice.org

é necessário reiniciar o computador. Aliás, isso não ocorre uma vez sequer em todo o processo.

Caso haja uma conexão com a Internet disponível, em um dos últimos passos da instalação o sistema chama o YOU (YaST Online Update) e verifica se o sistema instalado está atualizado. Por motivos de segurança, essa atualização é recomendada, mas prepare-se: mesmo com uma conexão de boa velocidade, a atualização do sistema pode demorar.

A figura 1 mostra o ambiente desktop da distribuição após a instalação ter sido finalizada.

Software adicional

A instalação deixa de fora alguns programas que podem ser interessantes, tais como o Mozilla Firefox, o Adobe Acrobat Reader etc. Esses programas não se encontram no CD (ver tabela 1: Principais aplicativos pré-instalados para saber o que está disponível por padrão após a instalação do sistema).

Para quem tem um link de banda larga, instalar software adicional no sistema não deve ser um problema. Basta chamar o YaST novamente, escolher o item *Modificar fonte da instalação* e inserir na caixa de diálogo que aparecer um dos diversos repositórios (FTP ou web) espelho da distribuição. Na figura 2 utilizamos como servidor, por pura falta de imaginação e originalidade (sem contar a preguiça de procurar outro) *ftp.suse.com*. O diretório para a versão 9.2 do sistema é *pub/suse/i386/9.2*. Vá tomar um café durante a atualização da lista de pacotes disponíveis e não se assuste se, ao voltar, ela ainda não tiver terminado. Aguarde e confie! Se lhe serve de consolo: isso só acontece quando você atualiza a fonte de instalação.

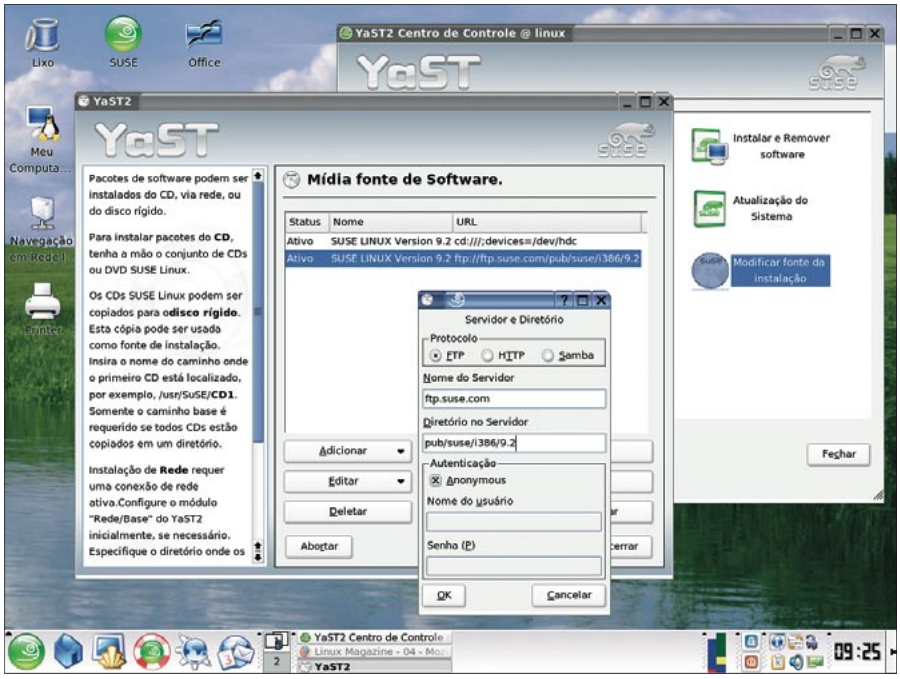


Figura 2: Estendendo o universo de pacotes instaláveis para além do seu CD.

Depois disso, clique em *Instalar e Remover software* (ver figura 3), procure os aplicativos que deseja instalar e siga as instruções na tela. Caso as coisas não estejam claras, sempre está disponível a Ajuda sensível ao contexto em português. No exemplo da figura 3 ilustramos a instalação do visualizador de documentos em formato PDF da Adobe.

Como todas as distribuições comerciais, a SuSE também não pode, por problemas de licença, fornecer um reprodutor de vídeos capaz de ler DVDs criptografados via CSS ("Content Scrambling System") ou que contenha decodificadores ("codecs") proprietários, que permitam tocar formatos como QuickTime ou WMV, apesar desses programas (*codecs*) estarem disponíveis na Internet. ■

Figura 3: Instalando um programa a partir do repositório FTP da distribuição.

Figura 4: Trailer de Guerra nas Estrelas: Episódio III reproduzido com o Kaffeine.

INFORMAÇÕES

[1] SuSE Linux: *http://www.novell.com/linux/suse/index.html*

[2] Codecs para o Kaffeine: *http://packman.links2linux.de/?action=046*

S.O.S

O CD-ROM que acompanha a Linux Magazine foi testado e, até onde pudemos constatar, se encontra livre de qualquer tipo de vírus ou conteúdo malicioso e de defeitos. Não nos responsabilizamos por qualquer perda de dados ou dano resultante do uso deste CD-ROM ou de software nele incluído. A Linux Magazine não oferece suporte técnico ao conteúdo deste CD.

SOBRE O AUTOR

Rafael Antonio Guido Peregrino da Silva foi chefe de pesquisa e desenvolvimento da Cyclades Europa. É um dos fundadores da Linux Magazine Brasil e atualmente atua, entre várias outras funções, como seu editor-chefe.

Emily Stone: National Science Foundation - EUA

Linux e seus sabores

Por que existem tantas distribuições? Parte II

Na edição passada, acompanhamos a história e principais características do Debian e do Fedora. Nesta edição, é a vez do Gentoo e do Slackware. POR SULAMITA GARCIA

O Gentoo [1] foi criado por Daniel Robbins, que começou usando Debian e criou uma versão desta distribuição chamada Enoch. O objetivo de Robbins era tornar o Enoch uma distro muito rápida, que tivesse criação e atualização de pacotes completamente automatizadas. Com o tempo, o Enoch se distanciou tanto do Debian que se tornou o Gentoo. Na época em que o Gentoo chegaria à versão 1.0, Robbins adquiriu uma máquina que tinha uma incompatibilidade de hardware com o Linux. Por causa disso, o projeto ficou parado por um tempo. Nesse período, Robbins começou a usar o FreeBSD e se tornou fã do ports, um repositório de pacotes extremamente eficiente, que compila os pacotes na máquina por padrão antes de instalá-los. Essa experiência serviu de inspiração para o Portage, repositório de pacotes do Gentoo. Logo depois o problema com o hardware foi consertado e Daniel voltou ao Linux. Com a ajuda de outros desenvolvedores, como Achim Gottinger, o Gentoo voltou à ativa.

O Gentoo possui um *Contrato Social* baseado no do Debian, porém simplificado. Esse contrato traz basicamente as mesmas premissas do documento que o inspirou: "O Gentoo é e permanecerá Software Livre", "o código será devolvido à comunidade" e "não esconderemos os problemas". Atualmente a distribuição tem suporte às arquiteturas x86, AMD64, PowerPC, UltraSparc, Alpha and MIPS. Possui mais de 8000 pacotes no seu repositório, o Portage, e está na versão 2004.3.

Uma definição aproximada do Gentoo seria dizer que ele é um *Linux From Scratch* automatizado. A documentação disponível no site é a melhor que já vi, extremamente didática e detalhada. Usuários que queiram começar a usar o Linux, mesmo que com outra distribuição, irão se beneficiar com os tópicos sobre particionamento, sistemas de arquivos e formatação de discos, configuração e conceitos básicos de rede, além de compreender as variáveis envolvidas na otimização de um sistema, como por exemplo a definição de otimizações adequadas ao processador da máquina. Essa documentação foi recentemente atualizada, em 17 de janeiro, com uma versão voltada para a instalação a partir do terceiro estágio (*stage 3*) de um Live CD. O grupo GentooBR [2] lançou há pouco tempo uma versão em português dessa documentação. Vale a pena conferir e também colaborar com outros projetos do grupo.

A distribuição concentra-se na otimização do desempenho do sistema,

Figura 1: O Gentoo é ideal para quem quer o máximo de desempenho do sistema.

usando para isso muita automatização. O processo gira em torno da configuração de parâmetros no arquivo */etc/make.conf* através das variáveis *CXXFLAGS* e *USE*. Os *CXXFLAGS* são parâmetros passados ao gcc. Com eles é possível selecionar, entre outras coisas, seu processador. Como este artigo trata de características gerais das distribuições, não iremos nos aprofundar nisso. A variável *USE*, por outro lado, irá definir como serão compilados os pacotes a instalar. Há uma longa lista de recursos para ativar ou desativar ao gosto do freguês, entre eles: Java, ALSA, criptografia, X, QT e GTK. Assim, esses recursos serão automaticamente ativados ou desativados, de acordo com suas escolhas, durante a compilação. Obviamente, se um pacote não tiver, por exemplo, suporte à criptografia, ativá-lo ou desativá-lo não terá efeito algum.

A instalação do Gentoo se divide em estágios – os *stages*. Eles definem de onde você vai começar a instalação. É possível começar diretamente do *stage 3* se você quiser, porém a diversão está em começar pelo menos do *stage 2*.

No *stage 1*, você irá executar o script *bootstrap.sh*, que irá compilar a *libc*, o *gcc* e vários outros programas essenciais do sistema, todos otimizados para seu processador. Isso demora bastante.

O *stage 2* é a compilação e instalação do sistema básico, de acordo com as configurações escolhidas ao se configurar a variável *USE*. Você pode baixar um arquivo de *stage 2* já preparado para seu processador nos mirrors do Gentoo e assim pular a parte do bootstrap. No *stage 3*, você pode escolher entre o kernel da série 2.4 ou 2.6 e se quer um kernel puro ou se gostaria de usar um conjunto de patches de otimização. E todo o resto, interface gráfica, clientes de irc, icq, serviços de rede…

Tudo isso é feito através do *emerge*, a ferramenta que resolve as dependências, verifica os recursos ativados na variável *USE* e instala os pacotes de acordo com as opções ali listadas. Através dele é possível atualizar toda a listagem do sistema e os pacotes, com os comandos *emerge sync* e *emerge world*. Com o emerge, também se podem baixar pacotes binários, o que às vezes é aconselhável, como no caso do OpenOffice. Os pacotes são constantemente atualizados, coisa que pode ser problemática: por ser uma distro preocupada com otimização e constante atualização, o Gentoo não é lá extremamente estável. Li num site que isso se deve a um pobre sistema de QA – *Quality Assurance* (Controle de Qualidade) – que idealmente deveria pegar os bugs antes do sistema ser liberado. Passei por problemas com a versão 2004.2 em um Athlon XP instalado a partir de um *stage 1* e só consegui resolvê-los usando o *stage 2*.

A instalação é dolorida – nunca espere gastar menos de alguns dias para ter um sistema funcional. Você pode gerar pacotes com o software que está instalando para reaproveitar mais tarde, sem ter de compilar tudo novamente. Porém, dois sistemas com algumas poucas configurações diferentes não irão aproveitar os pacotes uma da outra. Por exemplo, um usuário que goste de GTK e Python não vai aproveitar os pacotes compilados com suporte a QT e Java.

Todavia, após alguns dias de instalação, o resultado é impressionante. Foi emocionante ver o Mozilla e o OpenOffice.org nem sequer pensarem para abrir, numa máquina em que a única coisa não compilada era o próprio OpenOffice. Se você gosta de emoções fortes, de controlar à unha seu sistema e fazê-lo render ao máximo, porém com ferramentas automatizadas e scripts de inicialização prontos, o Gentoo é uma ótima pedida. Pode parecer incrível, mas existem pessoas, como uma colega minha, que quando não têm o que compilar buscam os últimos pacotes adicionados ao Portage só para ver seu sistema em ação. Há loucos pra tudo. Se você é um desses, leia o Gentoo Handbook [9] e divirta-se.

Slackware

O nascimento do Slackware [3] já foi contado na primeira edição desta revista. Resumidamente, ele é a mais antiga distribuição Linux em atividade, criada por Patrick Volkerding. Seu criador atualmente enfrenta sérios problemas de saúde, porém conta com o apoio e trabalho de vários grupos ao redor do mundo, liderados pelo GUS-BR [4], na manutenção do sistema e na torcida pelo seu pronto restabelecimento.

Na primeira vez que ouvi falar do Slackware, soube que era uma distribuição para hackers. Um pouco mais

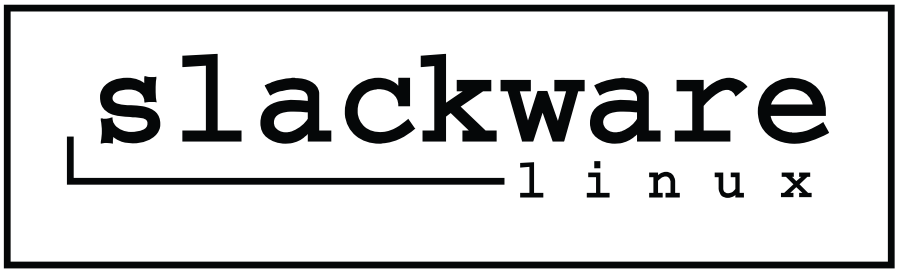


Figura 2: Até hoje o Slackware é "trabalho de um homem só", Patrick Volkerding.

adiante, que era uma distribuição para machos. Estou aqui para demonstrar que essa segunda premissa não é verdadeira. O Slackware simplesmente assume que o administrador tem o controle do sistema e que sabe o que está fazendo. Como disse Piter Punk em uma de suas palestras, ele "resgata o prazer da leitura". O sistema é indicado para usuários novatos ou experientes, desde que tenham, porém, vontade de entender o funcionamento do sistema e desejem total controle sobre o que está sendo instalado e configurado em sua máquina. Tudo isso com extrema simplicidade.

Conhecido por sua estabilidade, um dos grandes segredos do Slackware se deve ao fato de ser a única distribuição que não aplica *patches* nem adaptações aos softwares que distribui. E também de sempre lançar versões com pacotes atualizados e manter um rápido sistema de atualização, caso surjam problemas de segurança.

A instalação é toda em modo texto e os pacotes são divididos em séries: *A* (instalação base), *AP* (aplicações básicas de sistema), *D* (desenvolvimento), *K* (fontes do kernel), *N* (rede), *KDE*, *GNOME*, *L* (bibliotecas), *X*, *XAP* (aplicações gráficas). Depois de escolher as séries que deseja instalar, você seleciona o nível de controle que quer ter sobre os pacotes instalados: modo *expert* (todos os pacotes podem ser selecionados ou desativados), *menu* (pacotes essenciais ao sistema não são mostrados), *full* (instalação completa) ou por *tagfiles*. A instalação pode ser feita por CD ou através da rede, via FTP ou NFS.

O desenvolvimento do Slackware é feito exclusivamente por Volkerding. Ele toma as decisões do que entra ou não na distribuição. O que implica, por exemplo, no fato da distro não utilizar o PAM. Volkerding tem uma profunda aversão ao PAM e prefere deixar tudo simplificado pelo NSS. O ciclo de desenvolvimento é de aproximadamente 6 meses entre as versões. A filosofia da distribuição é sempre estar concentrada na simplicidade e estabilidade. A versão atual, 10.0, trouxe algumas novidades, como o suporte ao *UDEV* (que irá substituir o *devfs*), sistema */sys* (que irá substituir o */proc*) e atualizações gerais. O kernel 2.6 continua no diretório *testing*, não tendo sido oficialmente incorporado à distribuição.

Os scripts de inicialização do Slackware seguem o padrão BSD; todos ficam no diretório */etc/rc.d/*. Os scripts principais, que carregam o sistema de acordo com o runlevel escolhido, verificam as permissões de execução dos scripts de serviço (como *rc.httpd*, *rc.sshd*) e os executam, se permitido. Para desativar a execução de um serviço no boot, basta remover a permissão de execução do script correspondente.

O gerenciamento de pacotes é feito pelo pkgtool e seus agregados: *installpkg*, *removepkg*, *upgradepkg*, *makepkg* e *explodepkg*. Essas ferramentas já foram melhor detalhadas nas edições 1 e 2 da Linux Magazine Brasil. Resumidamente, servem para instalar, remover, atualizar ou construir pacotes do Slackware. Esses pacotes têm um formato bastante simples: são arquivos *tar.gz* contendo os binários, bibliotecas e demais arquivos que compõem o software, mais um script para executar algum comando necessário após a instalação e a descrição do software. Uma listagem desses arquivos, junto com a descrição, é colocada no diretório */var/log/packages*, num arquivo que leva o nome do pacote. Isso serve para consultas rápidas sobre versões e arquivos instalados.

O Slackware, por definição, não verifica as dependências entre pacotes. Alguns programas, entretanto, fazem essa verificação por conta e risco do autor, como o *swaret* e o *slapt*. Outro software, o *slackpkg* [5], apenas consulta a listagem de pacotes de um mirror escolhido, verifica versões, instala ou atualiza pacotes, de acordo com o comando escolhido.

Os pacotes são feitos com base nos *SlackBuilds*, scripts que compilam o software e formatam um pacote de acordo com o padrão oficial. Eles estão sempre disponíveis junto aos fontes do Slackware. Com eles, é fácil modificar alguma opção de compilação faltante ou excedente, como por exemplo suporte a algum recurso que não venha por padrão em um pacote.

O Slackware também tem uma versão super reduzida, chamada *ZipSlack*, que pode ser instalada em qualquer partição FAT (ou FAT32) com 100Mb de espaço livre e utiliza o sistema de arquivos UMSDOS. Isto significa que você não precisa reparticionar seu HD e instalar tudo para testar o Linux e conhecer a cara do Slackware.

Com tudo isso, espero ter fornecido uma visão geral das distribuições e que alguma tenha lhe chamado a atenção. Ao download!

INFORMAÇÕES

[1] Gentoo: *http://www.gentoo.org*
[2] Gentoo-BR: *http://www.gentoobr.org*
[3] Slackware: *http://www.slackware.org*
[4] GUS-BR: *http://gus-br.linuxmag.com.br/*
[5] Slackpkg: *http://slackpkg.sourceforge.net/*
[6] Swaret: *http://swaret.sourceforge.net*
[7] Slapt-get: *http://software.jaos.org*
[8] Trabalhando com o Portage: *http://www.gentoo.org/doc/pt_br/handbook/handbook-x86.xml?part=3&chap=0*
[9] Gentoo Handbook (em Português): *http://www.gentoo.org/doc/pt_br/handbook/index.xml*

SOBRE A AUTORA

Sulamita Garcia é formada em Ciências da Computação pela UFSC, onde conheceu o Linux. Participa do projeto LinuxChix e mantém a seção sobre Alta Disponibilidade do site UnderLinux. Possui certificação LPIC II e trabalha como Software Designer na Cyclades.

O Gerenciador de Janelas Fluxbox

Leve, *rápido*, prático

Velocidade é um traço de família. O Fluxbox herdou as melhores características de seu predecessor, o Blackbox, e acrescenta alguns recursos úteis à área de trabalho. POR ANDREA MUELLER

Ter dons de programação é algo fantástico. Não importa se seu aplicativo favorito é chato demais, extravagante ou carece de algumas opções – um punhado de linhas de código trará os recursos de que você precisa. Henrik Kinnunen viu-se exatamente nessa situação há algum tempo. O Blackbox [1] era seu gerenciador de janelas favorito, mas ele precisava acrescentar algumas coisas para ficar realmente satisfeito. Em vez de reinventar a roda, Henrik começou a trabalhar no código fonte do Blackbox 0.61.1, adicionou alguns recursos convenientes e apresentou sua criação, o Fluxbox [2].

Instalando o Fluxbox

O modo mais rápido de instalar o Fluxbox é usar pacotes RPM (ou DEB, se você usa o Debian), se estiverem disponíveis. Boa notícia para os usuários do SuSE Linux 9.0 – embora a SuSE não disponibilize RPMs, os pacotes do Mandrake funcionarão de maneira bastante satisfatória. Os usuários do Debian ficarão felizes em saber que o Fluxbox está no repositório oficial, acessível com o comando *apt-get install fluxbox*.

Não dá para compilar o Fluxbox 0.9.9 (ou versões anteriores) com o ***gcc*** 3.3 ou posterior. Se você tem a versão 8.2 ou 9.0 do SUSE Linux ou 9.2 ou 10.0 do Mandrake Linux, ou qualquer outra distro baseada no gcc 3.3, deve usar a versão 0.9.10 ou posterior (quando escrevíamos este artigo a mais atual era 0.9.11). Siga os passos de costume para compilar e instalar o Fluxbox:

```
./configure ; make ; su -c ➥
 make install-strip
```

Glossário

gcc: *O compilador C da GCC (GNU Compiler Collection). Traduz o código fonte de programas para um formato compreensível para a máquina, criando assim um executável a partir de um arquivo de texto.*

Dockapps do WindowMaker: *Mini-aplicativos [3] especialmente projetados para o ambiente gráfico WindowMaker (*http://www.windowmaker.org*/), criado pelo brasileiro Alfredo Kojima. Esses aplicativos rodam em segundo plano e podem ser acessados por um ícone em uma barra de programas comumente chamada de* dock *(doca ou atracadouro).*

GnuPG: *O "GNU Privacy Guard" (GnuPG ou GPG) é um programa que criptografa de forma segura vários tipos de dados. O sistema permite que apenas pessoas autorizadas consigam decifrar esses dados, protegendo documentos e mensagens de email contra o "olho grande" de espiões durante o trajeto – da mesma forma que um envelope protege o conteúdo de uma carta no correio "de verdade". Para que isso seja possível, é necessário o uso de* chaves de criptografia GnuPG *[4] para cifragem e decifragem das mensagens.*

Gerenciador de Login: *Programa que pede aos usuários que informem nome e senha. Caso estejam cadastrados, libera acesso ao ambiente gráfico escolhido, que de outra maneira estaria bloqueado.*

O gerenciador de janelas em si tem caráter frugal e demanda apenas os pacotes de desenvolvimento do *XFree86*, da *libpng* e da *freetype*. Se você usa regularmente os aplicativos do KDE, pode preferir substituir o *./configure* por *./configure --enable-kde*. Isso faz com que o Fluxbox exiba os programas do KDE que geralmente são iconizados no *kicker*, a barra de tarefas do KDE. Só que, como não estamos no KDE mas no Fluxbox, usa-se o *Slit*, uma barra para os **dockapps do WindowMaker**. Assim, aplicativos "K" como o gerenciador de área de transferência *klipper* ou o guarda-chaves *kgpg*, para o sistema de assinatura digital **GnuPG**, podem ser usados e minimizados numa bandeja como de costume.

Apertem os cintos: decolagem autorizada!

Depois de instalar o Fluxbox, temos que integrar esse novo ambiente gráfico ao nosso **gerenciador de login** (veja artigo à página 80). A boa notícia é que se você o instalou por meio de um pacote oficial da sua distribuição, isso já deve ter sido feito automaticamente.

Se instalou manualmente, entretanto, será necessário editar, também manualmente, a configuração de seu gerenciador de login.

Se você entra no sistema em modo texto, adicione a linha:

```
exec /usr/local/bin/fluxbox
```

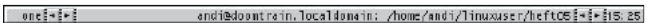


Figura 1: A barra de tarefas do Fluxbox, com seu seletor de áreas de trabalho e o infalível relógio no cantinho.

em seu arquivo *~/.xinitrc*. Desta forma, o comando *startx* carregará o Fluxbox.

Primeiros Passos

O Fluxbox é bastante direto em sua comunicação com o usuário. Há uma barra de tarefas com um relógio e um seletor de áreas de trabalho, normalmente quatro (ver figura 1). Programas iconizados são também exibidos nessa área. É possível usar os botões com setas para navegar entre as diversas áreas de trabalho; mas, se o usuário estiver realmente com pressa, pode usar a "rodinha" do mouse, caso possua um modelo desse tipo.

Para modificar esses padrões, clique com o botão direito do mouse na barra de tarefas para ter acesso ao menu de configuração. Os itens sob *Placement* definem a posição da barra de tarefas. Há seis posições predefinidas, desde o canto superior esquerdo (*Top Left*) até o canto inferior direito (*Bottom right*), passando pelo centro (ver figura 2).

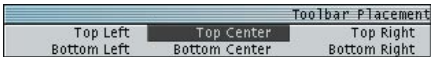


Figura 2: Onde você quer a barra mesmo?

Para que as janelas dos aplicativos não escondam a barra de tarefas, é interessante habilitar a opção *Always on top* (sempre no topo). A opção *Auto hide* (esconder automaticamente) faz com que a barra de tarefas desapareça se não estiver em uso. Não se preocupe: ela surgirá magicamente se o ponteiro do mouse passar próximo à posição que ela ocupa. A ativação simultânea dessas duas opções permitirá o uso integral da tela pelos aplicativos, aumentando a área útil. Sempre que preciso, bastará mover o mouse para o canto da tela e lá estará a barra de tarefas.

Para definir nomes diferentes para cada uma das áreas de trabalho – permitindo assim que ela seja facilmente distinguível das demais – selecione a opção *Edit current workspace name* (editar o nome da área de trabalho atual). No seletor, os nomes padrão (*one*, *two* etc.) serão substituídos pelo novo nome que você informar.

Meus programas

O **menu de contexto**, que possui uma bela seleção de programas pré-configurados, facilita o acesso ao que o usuário realmente precisa (ver figura 3). Os dois primeiros itens chamam um emulador de terminal e um navegador de Internet. Qual navegador e emulador de terminal será chamado depende do que estiver instalado no sistema.

O Fluxbox possui um programinha chamado *fluxbox-generate_menu*. Ele "fareja" os menus existentes atrás de aplicativos, que são gravados no arquivo *~/.fluxbox/menu*. Isso ocorre automaticamente na primeira vez em que o usuário entra no Fluxbox.

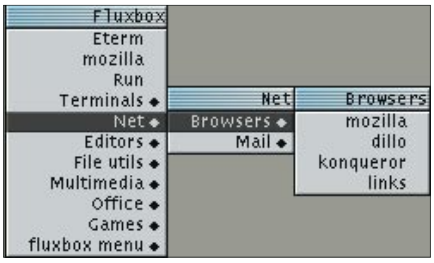


Figura 3: Durante o processo de compilação, o Fluxbox modifica o menu para refletir os programas e configurações que estão realmente instalados no sistema.

Uma ferramenta pequena e muito útil, o *fbrun*, não é encontrada por padrão. Ela mostra uma caixa de diálogo para rodar comandos arbitrários – como o menu "Executar" do Windows e do KDE. Para forçar sua instalação, digite:

```
fluxbox-generate_menu -p ➥
/usr/local/share
```

Isso colocará o *fbrun* no nível mais alto do menu. A opção *-p /usr/local/share* é importante se o Fluxbox for compilado em casa. Por padrão, o ambiente gráfico procura por suas configurações e temas no diretório */usr/share*. Entretanto, como compilamos a partir do código-fonte (e, portanto, não é um pacote oficial da distribuição) o Fluxbox está instalado no diretório */usr/local/share*. Se o caminho não for explicitamente indicado, nenhum dos diversos **temas** incluídos será mostrado no menu de configurações.

Glossário

Menu de contexto: *Um menu que surge quando se pressiona o botão direito do mouse sobre uma região específica dentro de uma janela. O menu mostrado contém opções exclusivas daquele objeto clicado. Normalmente, uma das opções mostradas reflete exatamente o que o usuário quer fazer.*

Tema: *Um método de alterar a aparência geral da área de trabalho e dos programas que rodam nela. Os temas são como uma coleção de roupas para o ambiente gráfico.*

Se a janelinha preta com letras brancas do shell e o navegador de Internet escolhidos pelo Fluxbox não lhe agradam, mais uma vez o comando *fluxbox-generate_menu* – desta vez com os parâmetros *-t* e *-b* – salvará o dia:

```
fluxbox-generate_menu -t ➥
konsole -b konqueror -p ➥
/usr/local/share
```

Com isso, o emulador de terminal *konsole* e o navegador Konqueror, ambos programas do KDE, serão adicionados ao menu.

Grupos

Com relação ao comportamento das janelas, o Fluxbox é muito semelhante a qualquer outro ambiente gráfico. Um clique transfere o foco para a janela. As barras de título das janelas possuem botões bem conhecidos para travar, minimizar, maximizar e fechar.

As abas na parte superior esquerda da barra de janelas (ver figura 4) são um recurso especial que permite associar múltiplos programas a um único "invólucro". O princípio de funcionamento é o mesmo do *pwm* [5], um gerenciador de janelas mais antigo e limitado. Arraste a aba com o botão do meio do mouse até a janela de destino. Ao soltá-la, a janela agora conterá duas abas, cada uma com um aplicativo diferente. Para desacoplar a aba de uma janela, simplesmente arraste-a e solte sobre a área de trabalho.

Figura 4: As abas permitem que vários programas sejam agrupados em uma única janela.

Figura 5: Adaptável sem perder o estilo: seu ambiente gráfico preferido vestido com o tema da *maison* Fluxarnation.

O trabalho de arrastar as janelas para agrupá-las é meio chato, não é? Para facilitar, as abas possuem atalhos úteis. Clique com o botão direito sobre elas para mostrar o menu principal. Os programas iniciados por esse menu não serão abertos em suas próprias janelas, mas numa aba da mesma janela.

Não está fácil o bastante? Se você quer que alguns programas sempre compartilhem a mesma janela, crie um arquivo chamado *.fluxbox/groups* no seu diretório pessoal (home). Adicione os nomes das classes de janelas que quer criar, um para cada programa. Cada nome deve estar em uma linha. A ferramenta *xprop* informa o nome da classe para um determinado programa. Digite:

```
xprop WM_CLASS
```

O ponteiro do mouse transforma-se numa mira. Clique na janela cujo nome de classe quer descobrir. O resultado mostrado no exemplo refere-se ao editor HTML do KDE, o Quanta:

```
WM_CLASS(STRING) = ➥
"quanta", "quanta"
```

e, para o Konqueror

```
WM_CLASS(STRING) = ➥
"konqueror", "konqueror"
```

O Mozilla causou problemas ao *xprop*, mostrando uma mensagem de *classe desconhecida* (*class unknown*). A linha abaixo no arquivo *groups* permite que o Konqueror e o Quanta automaticamente compartilhem uma janela.

```
quanta konqueror
```

Agora basta informar ao Fluxbox que o arquivo contém as definições de grupo. O lugar apropriado para isso é o *~/.fluxbox/init*. A última linha já deve possuir a declaração *sessiongroupFile:*. Só falta, então, incluir o caminho para o arquivo *groups* que acabamos de criar:

```
sessiongroupFile: ➥
/home/andi/.fluxbox/groups
```

O comando *fluxbox-menu* | *Reload Config*, presente no menu principal do Fluxbox, permite que o ambiente gráfico releia seus arquivos de configuração e aplique as mudanças sem que seja necessário reiniciar.

Configurações

O item *fluxbox-menu* no menu principal também é o lugar para alterar o comportamento do gerenciador de janelas. Os apreciadores do modelo de "foco desordenado", em que a janela ganha foco ao passar o mouse em cima sem clicar, vão gostar da opção *configure* | *Focus Model* | *Sloppy Focus*. Para que as janelas que ganham o foco sejam enviadas para a frente de todas as outras, é necessário ativar a opção *Auto Raise*.

O menu *Tab Placement* permite que as abas sejam posicionadas em 16 locais diferentes. As mais interessantes são as que incluem a palavra *Relative*. Há um item desses para cada borda da janela. A opção *Left Relative* irá atracar as abas na borda esquerda da janela. Se dois programas forem agrupados, cada aba ocupará 50% da altura da janela.

Listagem 1: Atalhos de teclado para o arquivo ~/.fluxbox/keys

```
# Use [Alt-Ctrl-Seta para a direita] e [Alt-Ctrl-Seta para a esquerda]
# para alternar entre áreas de trabalho.
Mod1 Control Right      :NextWorkspace
Mod1 Control Left       :PrevWorkspace
# [Alt-Ctrl-Enter] abre uma janela de terminal
Mod1 Control Return     :ExecCommand xterm
# [Alt-x] fecha a janela ativa
Mod1 x  :Close
# [Alt-m] maximiza e restaura uma janela
Mod1 m  :MaximizeWindow
# [Alt-s] "enrola" a janela, reduzindo-a a sua barra de título
Mod1 s  :ShadeWindow
# [Alt-d] liga e desliga as decorações das janelas
Mod1 d  :ToggleDecor
# Para maníacos por Emacs: também é possível associar
# seqüências de teclas. Em nosso exemplo, [Alt-y] [Alt-c]
# chamará o Mozilla.
Mod1 y Mod1 c   :ExecCommand mozilla
```

Glossário

Monitor do sistema ACPI: *Um programa que mostra a carga da bateria ou a temperatura do processador de um laptop, por exemplo. O hardware tem que, obrigatoriamente, ser compatível com o sistema ACPI ("Advanced Configuration and Power Interface") para tal e, necessariamente, precisa de uma distribuição Linux atual, configurada com suporte a ACPI. Veja a edição número 05 da Linux Magazine.*

Se preferir não usar abas, desabilite o item *Use Tabs*. Para ver o conteúdo da janela enquanto a estiver movendo, ative *Opaque Window Moving*.

Requinte e estilo

Os itens sob *fluxbox-menu* | *Styles* permitem alterar a aparência do ambiente. O Fluxbox possui nativamente uma coleção de mais de 20 temas. Os fãs do Blackbox apreciarão o tema *Artwiz*. Já *Lemon Space* parece exalar a fragrância refrescante da primavera, enquanto *Blue* é um tema futurista. Se nenhum deles lhe agradar, um mundo de novos temas pode ser baixado de [6], alguns deles maravilhosos (ver figura 5).

Será necessário criar um diretório chamado *~/.fluxbox/styles* para gravar os temas baixados da Internet, embora alguns deles façam isso automaticamente ao serem descompactados. Se o tema possuir seu próprio papel de parede, será necessário também criar o diretório *~/.fluxbox/backgrounds*. Depois de criar o diretório e colocar o tema dentro, basta acessar as opções *fluxbox-menu* | *System Styles* e *fluxbox-menu* | *User Styles* do menu principal.

Figura 6: Combine o Fluxbox com utilitários como o *gkrellm* e *desklaunch* para conseguir um desktop bonito, leve e funcional.

Feitiçaria e teclados

Amantes do teclado vão se apaixonar pelo Fluxbox. Uns poucos atalhos de teclado estão definidos por padrão, como por exemplo [Alt-Tab] para alternar entre janelas e [Alt-F1] até [Alt-F12] para alternar entre áreas de trabalho. Entretanto, basta editar o arquivo *~/.fluxbox/keys* com o editor de textos de sua preferência e adicionar tantos atalhos quantos sua imaginação e necessidades exigirem.

Cada atalho de teclado deve estar em uma linha, no formato:

```
atalho de teclado      :ação
```

O *atalho de teclado* deve conter pelo menos uma tecla modificadora como o [Alt] (o arquivo de configuração o chama de *Mod1*) ou *Control* (*Ctrl* em alguns teclados) e uma tecla adicional. Consulte a seção *KEYS FILE* na página de manual do Fluxbox para descobrir quais ações são reconhecidas. A configuração com comentários mostrada na listagem 1 traz alguns truques úteis para o arquivo *keys*.

O estranho mundo dos applets

No início do artigo, nos referimos a um tal de Slit, a barra que abriga dockapps do WindowMaker [3] e applets do Afterstep [7] e do KDE. Não o discutimos em detalhe então; afinal, o Slit está lá, aguardando fielmente em seu posto no canto inferior direito. Quando um programa é iniciado – por exemplo, o monitor do sistema *bubblemon* [8] e o **monitor do sistema ACPI** do KDE, o *akpi* [9], ambos os programas se acomodarão alegremente no Slit.

Se não gostar da posição padrão do Slit, na vertical e no canto direito da tela, isso também pode ser mudado: o applet *Direction* altera a orientação, entre vertical ou horizontal. *Placement* possui as mesmas opções discutidas antes para a barra de tarefas, permitindo escolher a posição na tela. Infelizmente, os práticos recursos de *"Auto hide"* e *"Always on top"* não estão disponíveis no Slit.

Todas essas conveniências fazem do Fluxbox mais do que uma belíssima opção para apreciadores de ambientes gráficos leves. De fato, o Fluxbox é uma tentação mesmo para os fanáticos pelos onipresentes KDE e Gnome. ■

SOBRE A AUTORA

Após quase dois anos como jornalista independente, Andrea Mueller agora trabalha como editora para a Linux New Media AG.

Quando não está lidando com artigos ou empacotando software, ela gosta de ir além do Linux, e se envolve com outros sistemas operacionais, como o QNX, BeOS e NetBSD.

INFORMAÇÕES

[1] Blackbox: *http://sourceforge.net/projects/blackboxwm/*

[2] Fluxbox: *http://fluxbox.sourceforge.net/*

[3] Artigo sobre dockapps: Joachim Moskalewski, "DockApps", Linux Magazine Internacional, Edição 3, página 128: *http://www.linux-magazine.com/issue/03/Dockapps.pdf*

[4] Artigo sobre GnuPG: Patricia Jung, "Key Signing Party", Linux Magazine Internacional, Edição 35, página 45: *http://www.linux-magazine.com/issue/35/Using_GnuPG_Keys.pdf*

[5] PWM: *http://modeemi.cs.tut.fi/~tuomov/pwm/*

[6] Temas para o Fluxbox: *http://fluxbox.sourceforge.net/themes.php*

[7] Applets para o Afterstep: Andrea Müller, "Docked", Linux Magazine Internacional, Edição 41, abril de 2004, página 78

[8] Utilitários para desktop: Andrea Müller, "Aquatic Utilities", Linux Magazine Internacional, Edição 31, página 78: *http://www.linux-magazine.com/issue/31/BubbleMon_WMFishTime.pdf*

[9] Akpi: *http://akpi.scmd.at/*

[10] gkrellm: *http://www.gkrellm.net*

[11] Desklaunch: *http://www.oroborus.org*

Configurando o KDM e GDM

A porta de entrada

Tanto o KDE quanto o GNOME possuem gerenciadores de login. Vamos olhar o que o KDM e o GDM têm a oferecer e aprender a configurar a tela de boas-vindas de seu sistema Linux. POR HAGEN HÖPFNER

Jenny W.: www.sxc.hu

Na maioria dos sistemas operacionais, o gerenciador de login é o primeiro programa gráfico que o usuário vê. Muitas distribuições iniciam o servidor gráfico (o X Window System) automaticamente após o boot. O usuário digita seu nome e senha e entra no sistema, depois de selecionar o ambiente gráfico de sua preferência. No gerenciador de login também é possível disparar outras ações, como desligar a máquina ou reiniciá-la.

O KDM, do projeto KDE, e seu "rival" no Gnome, o GDM, são os dois gerenciadores de login mais populares do mundo Linux, mas há outras alternativas, como o ancião XDM e o minúsculo WDM. O gerenciador de login não tem influência alguma nos ambientes gráficos instalados na máquina; em outras palavras, mesmo os maiores aficionados pelo KDE podem usar o GDM para se logar no sistema usando seu ambiente gráfico favorito. A única desvantagem é que instalar o GDM coloca uma grande quantidade de bibliotecas do Gnome no disco rígido, ocupando espaço valioso. Este artigo explica como mudar o gerenciador de login padrão no Fedora Core 2, SuSE Linux 9.1 e Mandrake Linux 10.0. Também explicamos como adicionar mais gerenciadores de janelas e ambientes gráficos ao menu de inicialização.

Primeiros Passos

A maioria das distribuições Linux configura um gerenciador de login já na instalação. O SuSE e o Mandrake usam o KDM por padrão (figura 1); o Fedora usa o GDM (figura 2). Para experimentar um novo gerenciador de login, deve-se, primeiro, instalá-lo. Enquanto para o GDM pode-se simplesmente instalar o pacote *gdm* no Fedora Core 2, SuSE e Mandrake, o KDM é encontrado numa miríade de pacotes RPM. No SuSE, por exemplo, o pacote é chamado *kdebase3-kdm*; o Mandrake o chama *kdebase-kdm* e os usuários do Fedora devem procurar por *kdebase*.

No Mandrake e no SuSE, é possível que o usuário nem veja o gerenciador de login, pois há a opção de selecionar um usuário para login automático quando da instalação do sistema. Isso pode ser aceitável para máquinas com apenas um usuário, já que provavelmente esse usuário vai usar sempre o mesmo ambiente. Entretanto, há algumas desvantagens. Por exemplo, se mudar de idéia, é difícil para o usuário alterar o ambiente gráfico em uso. Outro exemplo: qualquer um tem acesso a seus dados e docu-

Figura 1: A tela de login do KDM rodando no SuSE.

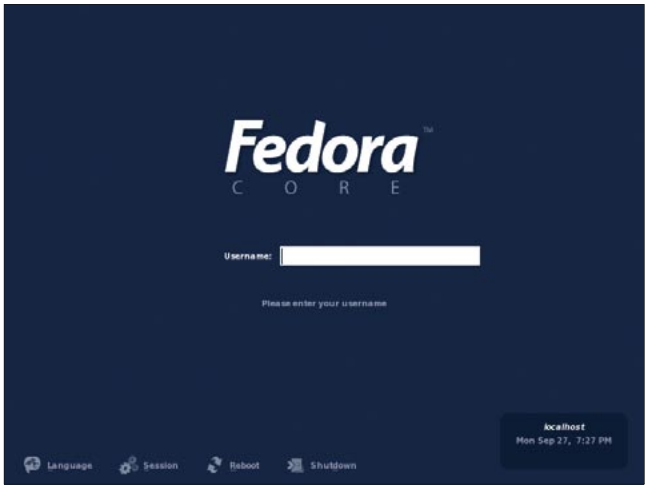


Figura 2: O GDM dá as boas-vindas aos usuários no Fedora Core 2.

mentos apenas ligando a máquina, já que nenhuma senha é necessária.

Para desabilitar o autologin no SuSE, selecione a opção *etc/sysconfig/Editor* no YaST. Ela está na categoria *System* (Sistema). Edite a opção *Desktop | Display manager | DISPLAYMANAGER_AUTOLOGIN*, apagando o nome do usuário no campo à direita.

No Mandrake usa-se o *drakconf*. Clique em *Autologin* na seção *Boot* e altere o padrão para *No, I don't want autologin* (Desabilitar autologin). Certifique-se de que a opção *Launch the graphical environment when your system starts* (Entrar em modo gráfico quando o sistema inicia) esteja ligada. Se essa opção estiver desligada, o Mandrake não vai iniciar em modo gráfico, forçando o usuário a logar-se em modo texto e digitar *startx* para usar o KDE.

GDM e KDM: uma questão de escolha

Para alterar o gerenciador de login padrão, é preciso fazer algumas alterações "à mão" tanto no Mandrake quanto no Fedora Core 2. Ambas as distribuições usam o arquivo */etc/X11/prefdm* para chamar o programa de login gráfico. O *prefdm* lê as configurações para o gerenciador de login a partir de uma variável de ambiente chamada *DISPLAYMANAGER*. O arquivo */etc/sysconfig/desktop* configura o valor dessa variável. A linha que define o programa de login gráfico será *DISPLAYMANAGER="kdm"* para o gerenciador de login do KDE ou *DISPLAYMANAGER="gdm"* para o do Gnome. Assim que o servidor gráfico for acionado, o gerenciador de login que você configurou será mostrado.

O Mandrake esconde um "segredinho". A distribuição francesa oferece duas variantes do KDM: uma versão modificada, específica do Mandrake e usada por padrão (Figura 4) e a versão original, intocada. Se o valor *kdm* for atribuído à variável *DISPLAYMANAGER*, o KDM original é mostrado. A versão modificada do Mandrake será usada se a linha apropriada em */etc/sysconfig/desktop* for modificada para *DISPLAYMANAGER="KDE"*.

Os usuários do SuSE devem usar o */etc/sysconfig/Editor* no YaST. Essa ferramenta é, na verdade, uma interface para acesso fácil ao diretório */etc/sysconfig/*. Se você preferir um editor de textos comum, simplesmente edite a variável *DISPLAYMANAGER* no arquivo */etc/sysconfig/displaymanager*.

Domando o KDM

Cada distribuição ajusta o KDM para que se comporte de uma determinada maneira. No SuSE, por exemplo, o KDM, além de pedir aos usuários que informem seu *login* e *senha*, possui um menu que permite selecionar o *tipo de sessão*, além de funções para reiniciar o servidor X e desligar (*Shutdown*) a máquina (figura 3).

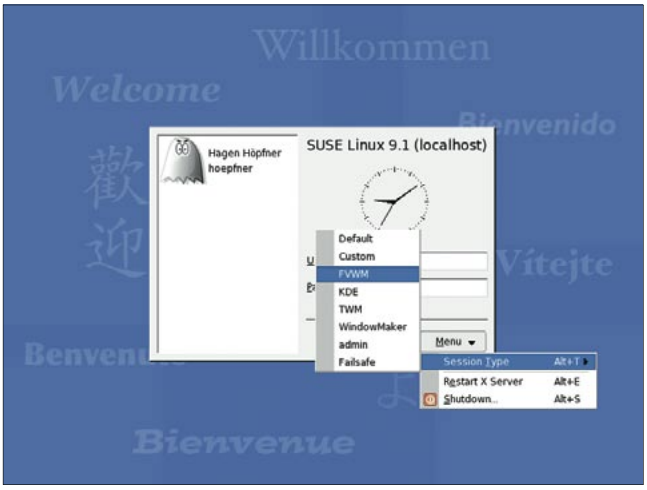


Figura 3: O SuSE oferece aos usuário todas as opções do KDM em uma única tela.

O Mandrake espera que o usuário selecione da lista um nome de login e, em seguida, mostra uma caixa de diálogo (figura 4) para que os usuários escolham o ambiente gráfico que desejam usar. O KDM do Fedora é o que possui a interface mais simples, mas comporta-se como o do SuSE.

Todas as três variações permitem que o usuário se registre no sistema, mas há pelo menos uma boa razão para mudar o padrão. O tipo de sessão é baseado nos ambientes gráficos pré-instalados (KDE, Gnome, Window Maker, IceWM...). Se o usuário instalar outro ambiente, o gerenciador de login não o mostrará na lista de ambientes disponíveis, a menos que a própria distribuição tenha incluído um script para isso. KDM e GDM ignoram ambientes sem esse script.

Por exemplo, suponha que o usuário instalou o gerenciador de janelas *qlwm*. Para dizer ao KDM para que adicione o *qlwm* na lista de ambientes disponíveis, é preciso criar um novo tipo de sessão. O KDE costumava ter um menu em seu Centro de Controle para essas eventualidades, mas infelizmente ele foi removido do KDE 3.2. Agora, é necessário que o usuário entre como root e crie manualmente um script de sessão para o novo ambiente gráfico – em nosso caso, o qlwm. Esse script é simplesmente um arquivo texto contendo parâmetros para o ambiente em questão. O SuSE armazena esses scripts em */opt/kde3/share/apps/kdm/sessions/*; o Mandrake e o Fedora Core 2 procuram em */usr/share/apps/kdm/sessions/*. Crie um arquivo para o novo ambiente – *qlwm.desktop* em nosso exemplo (a extensão *.desktop* é obrigatória) – e, dentro dele, digite os dados a seguir:

```
[Desktop Entry]
Encoding=UTF-8
Type=XSession
Exec=/usr/local/bin/qlwm
TryExec=/usr/local/bin/qlwm
Name=QLWM
```

Figura 4: O KDM no Mandrake – passo a passo.

Pronto! Da próxima vez que alguém se logar no sistema, o KDM oferecerá a opção de usar o *QLWM*. O tipo de sessão é definido pelo parâmetro *Name* = – em nosso caso, *QLWM*. O sistema assume que há um arquivo chamado */usr/local/bin/qlwm* e que o arquivo é executável (*TryExec*). Agora basta selecionar *QLWM* como tipo de sessão ao informar nome e senha e o KDM irá iniciar o ambiente gráfico apropriado – ou seja, o programa indicado na linha que começa com *Exec*. O KDM classifica a lista de ambientes gráficos em ordem alfabética de acordo com o nome do script. Para que o *QLWM* seja o primeiro item da lista, renomeie o script para *00qlwm.desktop*.

De roupa nova

Se você não está muito satisfeito com a maneira como o KDM se apresenta, que tal "maquiá-lo" um pouquinho? No Centro de Controle do KDE, escolha *System administration* | *Login manager* (Administração do sistema | Gerenciador de login – veja a figura 5). Além do ajuste fino na aparência (*Appearance / Font / Background* – *Aparência / Fonte / Fundo*), há três abas com opções gerais. A aba *Shutdown* permite especificar quais usuários podem ligar, desligar e reiniciar o computador. Já em *Users* – *Usuários* você define quais das contas aparecerão na lista de usuários que o KDM mostra.

Para mais detalhes sobre a configuração do KDM, há mais material disponível em [1]. Infelizmente, o manual do KDM não está atualizado, embora supostamente devesse abordar a versão 3.3 do KDE.

GLOSSÁRIO

XDMCP: *Acrônimo para X Display Manager Control Protocol [2], um protocolo de controle desenvolvido para possibilitar acesso a servidores X em outras máquinas da rede. Usuários remotos podem trabalhar como se estivessem sentados na frente do servidor e usando os programas disponíveis nele. Por razões de segurança – e para diminuir o tráfego na rede – o protocolo SSH é preferível em lugar do XDMCP.*

Tema: *Um tema é um conjunto de elementos que força uma determinada aparência em um programa que possua uma interface gráfica. Além dos diversos ambientes gráficos e seus gerenciadores de login, programas como o Mozilla ou o XMMS podem ser personalizados com temas.*

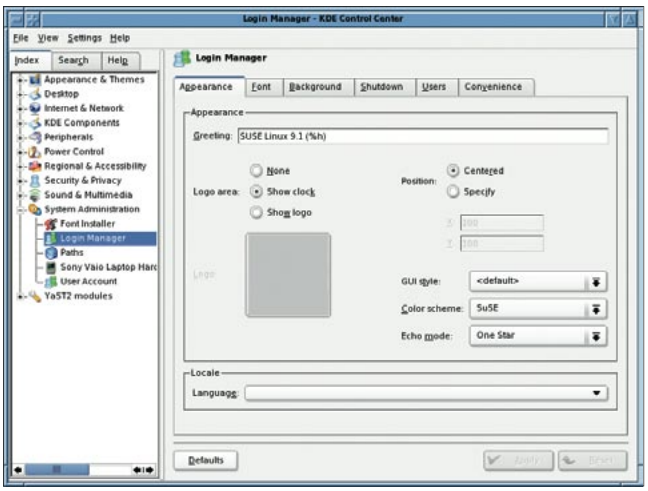


Figura 5: Modificando a aparência e o comportamento do KDM.

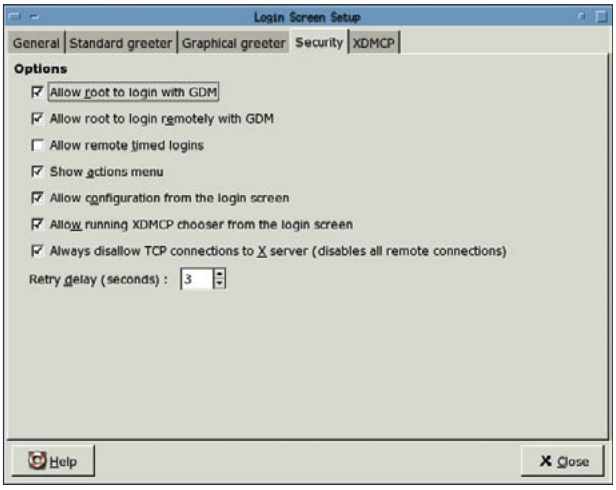


Figura 6: O programa de configuração do GDM.

Login à moda do Gnome

Assim como o KDM, a aparência do GDM depende da distribuição. O SuSE possui um seletor de idiomas (*Language*) que permite escolher a língua a ser usada após o login. Já *Session* mostra aos usuários um cardápio com os ambientes gráficos disponíveis. Por último, *System* possui opções para desligar e reiniciar o computador, além de poder configurar o servidor **XDMCP** [2]. Há ainda a opção de chamar o programa de configuração do GDM (figura 6).

O GDM no Mandrake é exatamente igual, à exceção do menu *System*, que aqui é chamado de *Actions*.

No Fedora, que usa o GDM por padrão, não é possível chamar a ferramenta de configuração por ele. O usuário root deve executar o utilitário *gdmsetup* para acessar essas opções.

Na aba *Security*, é possível definir se a ferramenta de configuração vai ficar ou não disponível na tela de login. Entretanto, a opção *Allow configuration from the login screen* (Permitir configuração a partir da tela de login) no Fedora Core 2 não faz o que promete: ainda não há um botão para o *gdmsetup* no login. O **tema** do GDM para o Fedora Core 2 é o responsável. Se um tema diferente for escolhido – o que pode ser feito na aba *Graphical greeter*, o GDM do Fedora exibirá a ferramenta na tela. Como a variedade de temas pré-instalados é pequena, procure em [3] um que lhe agrade. É preciso rodar o *gdmsetup* para instalar os temas baixados da Internet.

É simples dizer ao GDM para mostrar, no menu *Sessão*, um novo ambiente gráfico – tão simples como no KDM. A única diferença é o caminho para o script de sessão. No SuSE, o GDM espera encontrar os scripts em */opt/gnome/share/xsessions*. Mandrake e Fedora Core 2 usam o diretório */usr/share/xsessions/*. Para saber mais sobre a configuração do GDM, leia o manual de referência do programa, encontrado em [4]. ■

INFORMAÇÕES

[1] Manual do KDM: *http://docs.kde.org/pt_BR/HEAD/kdebase/kdm/*

[2] HOWTO para o XDMCP: *http://www.tldp.org/HOWTO/XDMCP-HOWTO/*

[3] Temas para o GDM: *http://art.gnome.org/themes/gdm_greeter/*

[4] Manual de Referência do GDM: *http://www.jirka.org/gdm-documentation/t1.html*

Bloqueando propagandas com seu navegador ou um proxy

Surfando sem banners

Propagandas em sites podem dificultar sua navegação e consomem muita banda. Neste artigo, conheceremos ferramentas que permitem bloquear banners em sua própria máquina ou rede local. POR OLIVER FROMMEL

Atualmente, há pouquíssimos sites que não contenham anúncios publicitários. Além dos banners que cobrem toda a extensão da tela, há uma tendência crescente de utilizar grandes imagens que substituem o cabeçalho, dificultando a navegação dos usuários (ver figura 1).

Embora isso possa ser compreensível do ponto de vista do provedor de conteúdo, a maioria dos usuários não acha nem um pouco divertido trombar com propagandas, especialmente se elas afetam a velocidade de navegação. De fato, é bastante comum que anúncios espalhafatosos, sejam imagens ou apresentações em Macromedia Flash que consomem mais memória do que o verdadeiro conteúdo do site.

Esses problemas levaram ao desenvolvimento de alguns programas que evitam que banners indesejados apareçam em sites – e mesmo que sejam carregados. As ferramentas de bloqueio filtram os dados, permitindo que o conteúdo solicitado passe e rejeitando a propaganda indesejável. Eles funcionam porque o navegador vai buscar a página direto no servidor.

Figura 1: Uma página da web cheia de propaganda atrapalha a visualização do conteúdo real.

O software inteligente busca imagens de propaganda na página e remove os pedaços ofensores do arquivo HTML. Só então o navegador reúne os elementos necessários para exibir a página.

Há diversos métodos. Alguns navegadores têm um recurso integrado ou podem usar um plugin ou extensão. Programas de filtragem que rodam como **servidores proxy** independentemente do navegador mas na mesma máquina são uma alternativa. Esse tipo de software pode rodar numa máquina separada se necessário, servindo de proxy para os computadores de sua rede doméstica ou empresarial.

Integrado com o navegador

Os usuários do Mozilla têm uma opção simples, um plugin que bloqueia propaganda. A ferramenta, apropriadamente chamada AdBlock, filtra com base na **URL**. É fácil instalar o plugin – basta clicar no link na homepage do projeto [1]. Não são precisos privilégios de administrador para fazê-lo, já que o AdBlock é instalado no diretório pessoal (home) do usuário atual em *~/.mozilla*.

É necessário reiniciar o navegador para habilitar o plugin. O programa está disponível na seção *AdBlock*, no menu *Tools* (*Ferramentas*). Se você acrescentar o tipo de padrões mostrados na figura 2, o plug-in bloqueará um número bastante respeitável de anúncios. Um menu permite visualizar os elementos na página atual. Se você descobrir um anúncio que fugiu do filtro, pode usar sua URL para criar um novo filtro.

Também é possível clicar no anúncio com o botão direito e selecionar *Adblock Image* no menu drop-down para acessar os mesmos recursos. Isso fará com que se abra uma pequena caixa de diálogo com o endereço daquela imagem. Você pode usar um **caractere curinga** (*) para modificar a URL e capturar endereços semelhantes. Vamos supor que a seguinte URL seja exibida:

```
http://img-cdn.mediaplex.com/➥
ads/2399/9556/DE_DE_mofg_dim➥
4600_dhs_q1w0304_300x200_15k_➥
FL_gif.gif
```

Remova a seção que vem depois do nome do servidor e o diretório *ads* e digite um asterisco como caractere curinga no lugar:

```
http://img-cdn.mediaplex.com➥
/ads/*
```

Essa linha manda que o plugin bloqueie todos os arquivos do diretório *ads* no servidor Mediaplex. Agora clique em *Reload* (*Atualizar*) e *voilà*, o anúncio desaparece. Candidatos prováveis são fáceis de encontrar usando o menu mencionado anteriormente, ou inspecionando o código fonte do arquivo HTML (ver figura 3). Usuários do navegador Mozilla Firefox (antes conhecido como Firebird) vão gostar de saber que o plugin também funciona ali (fizemos o teste com o Firefox 0.8).

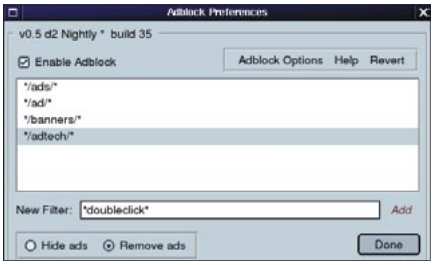


Figura 2: Usando o menu do Mozilla para configurar o AdBlock.

Alguns anúncios podem ser bloqueados sem o plugin. O modo mais fácil de fazê-lo é habilitar a opção *Load Images | for the originating site only* (*Carregar Imagens | apenas do site de origem*) presente no menu *Options | Web Features* (*Opções | Web*). Isso funciona muito bem, pois a maior parte dos banners publicitários não são realmente gerados pelo servidor web que hospeda a página, mas fornecidos por terceiros, que gerenciam para o provedor de conteúdo as compras dos internautas ocorridas por meio dos anúncios no site. Para usar esse recurso durante a navegação, clique no anúncio com o botão direito e selecione *Block images from server* (*Bloquear imagens do servidor*).

Note que esse método pode ser um tiro no pé, já que vai impedir que você baixe quaisquer imagens que não estejam armazenadas no servidor original. Além disso, ele não bloqueará os anúncios que estejam no site original. O plugin AdBlock certamente tem configurações mais granulares. Se mesmo assim você não estiver satisfeito, pode preferir usar um proxy mais flexível e poderoso, que rode em outros navegadores como o Konqueror e o Opera.

Café saboroso

O Muffin é um desses proxies. Fica entre o servidor web e seu navegador e filtra o tráfego das propagandas pela web. Por ser um pacote Jar, pode ser rodado diretamente com o Java, não necessitando de instalação ou compilação. É necessário ter o Java Runtime Environment (JRE), mas a maioria das distribuições instala o JRE por padrão. Ao baixá-lo de [2], lembre-se de clicar com o botão direito no arquivo Jar e escolher *Save link to disk*. De outra forma, seu navegador pode decidir iniciar o arquivo Jar diretamente, o que seria implorar para ter problemas.

Se o interpretador *java* não estiver em seu caminho de execução (path), adicione o diretório em que o programa está armazenado, por exemplo:

```
export PATH=$PATH:/usr/java➥
/j2sdk1.4.2_02/bin
```

Inicie então o Muffin digitando *java -jar muffin-0.9.3a.jar* (ver figura 4). O Muffin usa o filtro *NoThanks* por padrão. O filtro utiliza regras simples para bloquear anúncios. O proxy inclui alguns filtros adicionais, que não estão habilitados por padrão, como o *Animationkiller* para remover animações gif e o *Cookiemonster*, que dá um jeito nos onipresentes **Cookies**. O filtro NoThanks não funciona com as configurações padrão. É necessário antes baixar um pequeno arquivo de configuração, chamado killfile. Uma amostra do killfile está disponível no site do Muffin, no menu *Samples*.

Para baixar o killfile, selecione a entrada para o filtro *NoThanks* em *Enabled Filters* e clique em *Preferences*. Na caixa de diálogo que surge, selecione *Browse* em *Kill File* e localize o *Killfile* em seu diretório Muffin. Clique em *Apply*, e em seguida em *Load*, para mandar o Muffin usar o killfile. O botão *Save* armazena essa configuração permanentemente, o que é uma boa idéia, a não ser que você prefira repetir eternamente esse procedimento a cada vez que o Muffin for iniciado. O programa de filtro irá criar um diretório *Muffin* em sua pasta pessoal e o usará para armazenar sua configuração e um arquivo de registro (*log*).

Figura 3: É possível inspecionar o código fonte HTML para identificar o endereço das propagandas. Muitos sites, na verdade, usam "ad" como parte de seu nome.

Glossário

Servidores proxy *(ou simplesmente proxies) ficam entre um cliente (p. ex., um navegador web) e um servidor. Visto da perspectiva do cliente, o proxy é um servidor, enquanto do ponto de vista do servidor ele é um cliente. Os proxies armazenam os websites em cache, aumentando assim a velocidade de acesso. Em alguns casos, um proxy é necessário para permitir que clientes sem conexão direta possam acessar a internet.*

Caractere Curinga: *Muitos programas do Linux (como o shell) usam caracteres específicos para representar uma ou mais letras. Por exemplo, o asterisco (*) tipicamente representa um conjunto de caracteres arbitrário, embora possa significar qualquer número de repetições de um determinado caractere (em expressões regulares). Quando você digita ls *.jpg, o shell exibirá todos os nomes de arquivo com o sufixo .jpg, quaisquer que sejam as iniciais desses nomes.*

URL: *Um Uniform Resource Locator (Localizador Padronizado de Recursos) compreende um serviço representado por um acrônimo (http, ftp, ..), o endereço de um servidor da Internet e diretórios e nomes de arquivo opcionais. Isso permite que os endereços da internet sejam individualmente identificados, por exemplo:* http://www.linuxmagazine.com.br/issue/01/KDE_32_Tips.pdf.

Cookies: *pequenos fragmentos de informação em texto associadas pelo browser a um website. Os provedores de conteúdo usam cookies para armazenar informações sobre os costumes de navegação pela internet do usuário entre duas visitas a um site ("Quando esse usuário visitou pela última vez este site?")*

Port: *Uma vez que diversos programas servidores podem rodar numa só máquina, uma combinação do número da porta e do endereço IP é usada para identificar cada uma das conexões. Os padrões da Internet atribuem portas determinadas para serviços específicos; por exemplo, a porta 80 para a World Wide Web (HTTP) e a porta 25 para o sistema de envio de email (SMTP).*

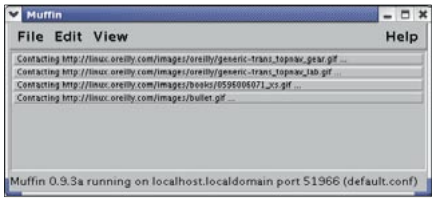


Figura 4: O Muffin, baseado em java, tem sua própria interface gráfica, e informa quais os arquivos com que está lidando.

Tabela 1: Visão geral dos filtros web

	Adblock	Muffin	Privoxy	Webwasher
Possibilidade de instalação por usuários comuns	sim	sim	sim	não
Proxy	não	sim	sim	sim
Porta padrão	–	51966	8118	9090
Em desenvolvimento?	sim	não	sim	sim
Pré-configurado	não	não	sim	sim
Licença	livre	livre	livre	restrita, livre para uso privado

Para que a nova ferramenta funcione, é preciso mudar as configurações de seu browser ao instalá-la como um proxy. Isso se aplica a quaisquer ferramentas de proxy que você pretenda usar. Se seu navegador for o Mozilla, abra as configurações em *Edit | Preferences* (*Editar | Preferências*) e clique no pequeno triângulo *Advanced* (*Avançado*). Procure por *Proxies*; ali você pode habilitar a configuração manual e digitar os valores corretos para *HTTP Proxy* e *Port* (ver figura 5). O primeiro campo, tipicamente, é *localhost*. Digite o número da porta de seu proxy no segundo campo (ver tabela 1). O Opera tem configurações similares no menu *Network | Proxy Servers*; o Firefox usa *Ferramentas | Opções*, e finalmente *Conexão*.

Um filtro com história

Esse software baseia-se numa ferramenta clássica, o Junkbuster [3]. O Privoxy [4] também ainda está em desenvolvimento ativo. O website do Privoxy tem pacotes pré-compilados para diversas distribuições. Por padrão, o pacote será instalado para todos os usuários através da conta do administrador. O Privoxy pode usar uma conta de usuário não-privilegiado, mas para isso exige que se façam algumas mudanças "à mão".

Figura 5: Configuração do proxy no Mozilla mostrando as configuração do filtro web Muffin rodando na porta 51966 da mesma máquina (*localhost*).

Quando o Privoxy é instalado no Red Hat, o sistema operacional roda o filtro na inicialização, junto com outros programas servidores. Você pode fazer isso manualmente usando */etc/rc.d/init.d/privoxy start*. Os arquivos de configuração estão em */etc/privoxy*; o arquivo principal é o */etc/privoxy/config*. Use esse arquivo para os ajustes mais granulares; porém, o Privoxy funciona bastante bem com as configurações padrão.

Depois de definir a porta como 8118 em seu navegador, como descrito antes, o Privoxy começará a filtrar anúncios nas páginas web que você visitar. O programa tem um recurso útil de marcar o lugar de onde removeu a propaganda, e pode exibir tanto a imagem filtrada como a regra de filtragem na qual ela se encaixa. Assim, você pode conferir se a ferramenta está apenas removendo propaganda ou também bloqueando o acesso a imagens que você gostaria de ver. Também é possível configurar o Privoxy diretamente a partir do navegador usando um endereço especial: *http://p.p* (ver figura 6).

Lava mais branco

Os usuários domésticos podem querer dar uma olhada no Webwasher [5], uma ferramenta livre da empresa Webwasher. A versão comercial tem filtros especiais para bloquear a execução de código em Javascript, que não estão disponíveis na versão livre. É preciso ter privilégios de administrador para instalar o Webwasher, e não há como contornar isso. A empresa fornece dois formatos de pacote, um RPM e um arquivo tar compactado com o gzip, que contém um script de instalação. A tarefa de instalação é assim reduzida a um simples comando executado pelo administrador.

O software é imediatamente iniciado, sem que lhe peçam para fazê-lo, e atende a pedidos na porta 9090. Os arquivos de configuração estão em */etc/*

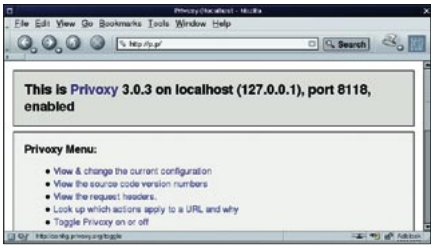


Figura 6: A página de configuração do Privoxy em um navegador, onde você pode clicar para habilitar ou desabilitar o filtro.

wwasher e os arquivos de log em */var/log/wwasher*. O registro de eventos (log) fica desativado por padrão, mas pode ser ativado usando um configurador baseado em web – você pode acessá-lo com seu navegador de Internet por um endereço especial: *http://-web.washer-/*. O nome de usuário padrão é *admin* e a senha é *webwasher*.

Mas qual escolher?

A escolha de uma ferramenta de filtragem é basicamente questão de gosto pessoal. Se você detesta Java, não vai apreciar o Muffin. Se insiste em utilizar apenas software completamente livre, pode descartar o Webwasher. Os usuários de Mozilla e Firefox podem baixar o plugin AdBlock, embora ele não seja tão poderoso quanto outros filtros. O Privoxy é uma ferramenta madura e em desenvolvimento ativo, disponível como pacote pré-compilado para muitas distribuições. E não é difícil instalar os programas que analisamos.

INFORMAÇÕES

[1] AdBlock: *http://adblock.mozdev.org*

[2] Muffin: *http://muffin.doit.org*

[3] Junkbuster: *http://internet.junkbuster.com/*

[4] Privoxy: *http://www.privoxy.org*

[5] Webwasher: *http://www.webwasher.com/client/download/private_use/linux*


Dave Hamilton - www.sxc.hu

Curso de Shell Script

Papo de Botequim VI

Blocos de código e laços (ou *loops*, como preferem alguns) são o tema do mês em mais uma lição de nosso curso de Shell Script. Garçom, salta uma boa redondinha, que tô a fim de refrescar o pensamento! POR JULIO CEZAR NEVES

Fala, cara! E aí, já tá sabendo tudo do comando *for*? Eu te deixei um exercício para treinar, se não me engano era para contar a quantidade de palavras de um arquivo... Você fez?

– Claro! Tô empolgadão com essa linguagem! Eu fiz da forma que você pediu, olha só...
– Êpa! Peraí que eu tô sequinho pra tomar um chope. Aê Chico, traz dois por favor. Um sem colarinho!
– Como eu ia dizendo, olha como eu fiz. É muito fácil...

```
$ cat contpal.sh
#!/bin/bash
# Script meramente pedagógico
# cuja função é contar a
# quantidade de palavras de
# um arquivo. Supõe-se que as
# palavras estão separadas
# entre si por espaços, <TAB>
# ou <ENTER>.
if  [ $# -ne 1 ]
then
    echo uso: $0 /caminho/do/➥
arquivo
    exit 2
fi
Cont=0
for Palavra in $(cat $1)
do
    Cont=$((Cont+1))
done
echo O arquivo $1 tem $Cont ➥
palavras.
```

Ou seja, o programa começa, como sempre, verificando se a passagem de parâmetros foi correta; em seguida o comando *for* se incumbe de pegar cada uma das palavras (lembre-se que o *$IFS* padrão é branco, *TAB* e *ENTER*, que é exatamente o que desejamos para separar as palavras), incrementando a variável *$Cont*. Vamos relembrar como é o arquivo *ArqDoDOS.txt*.

```
$ cat ArqDoDOS.txt
Este arquivo
foi gerado pelo
DOS/Rwin e foi
baixado por um
ftp mal feito.
```

Agora vamos testar o programa passando esse arquivo como parâmetro:

```
$ contpal.sh ArqDoDOS.txt
O arquivo ArqDoDOS.txt tem 14
palavras.
```

Funcionou legal! Se você se lembra, em nossa última aula mostramos o loop *for* a seguir:

```
for ((; i<=9;))
do
    let i++
    echo -n "$i "
done
```

Uma vez que chegamos neste ponto, creio ser interessante citar que o Shell trabalha com o conceito de "Expansão Aritmética" (*Arithmetic Expansion*), que é acionada por uma construção da forma *$((expressão))* ou *let expressão*.

No último loop *for* usei a expansão aritmética das duas formas, mas não podemos seguir adiante sem saber que a expressão pode ser uma das listadas na tabela 1.

Mas você pensa que o papo de loop (ou laço) se encerra no comando *for*? Ledo engano, amigo, vamos a partir de agora ver mais dois comandos.

O comando *while*

Todos os programadores conhecem este comando, porque é comum a todas as linguagens. Nelas, o que normalmente ocorre é que um bloco de comandos é executado, enquanto (enquanto, em inglês, é "while") uma determinada condição for verdadeira.

Tabela 1: Expressões no Shell

Expressão	Resultado
id++ id--	pós-incremento e pós-decremento de variáveis
++id --id	pré-incremento e pré-decremento de variáveis
**	exponenciação
* / %	multiplicação, divisão, resto da divisão (módulo)
+ -	adição, subtração
<= >= < >	comparação
== !=	igualdade, desigualdade
&&	E lógico
||	OU lógico

Pois bem, isso é o que acontece nas linguagens caretas! Em programação Shell, o bloco de comandos é executado enquanto um comando for verdadeiro. E é claro, se quiser testar uma condição, use o comando *while* junto com o comando *test*, exatamente como você aprendeu a fazer no *if*, lembra? Então a sintaxe do comando fica assim:

```
while comando
do
    cmd1
    cmd2
    ...
    cmdn
done
```

e dessa forma, o bloco formado pelas instruções *cmd1, cmd2,...* e *cmdn* é executado enquanto a execução da instrução *comando* for bem sucedida.

Suponha a seguinte cena: tinha uma tremenda gata me esperando e eu estava preso no trabalho sem poder sair porque o meu chefe, que é um pé no saco (aliás chefe-chato é uma redundância, né?), ainda estava na sala dele, que fica bem na minha passagem para a rua. Ele começou a ficar cabreiro depois da quinta vez que passei pela sua porta e olhei para ver se já havia ido embora. Então voltei para a minha mesa e fiz, no servidor, um script assim:

```
$ cat logaute.sh
#!/bin/bash
# Espero que a Xuxa não tenha
# copyright de xefe e xato :)
while who | grep xefe
do
    sleep 30
done
echo O xato se mandou, não ➥
hesite, dê exit e vá à luta
```

Neste scriptzinho, o comando *while* testa o pipeline composto pelos comandos *who* e *grep*, que será verdadeiro enquanto o *grep* localizar a palavra *xefe* na saída do comando *who*. Desta forma, o script dormirá por 30 segundos enquanto o chefe estiver logado (Argh!). Assim que ele se desconectar do servidor, o fluxo do script sairá do loop e te mostrará a tão ansiada mensagem de liberdade. Mas quando executei o script, adivinha o que aconteceu?

```
$ logaute.sh
xefe  pts/0   Jan  4 08:46 ➥
(10.2.4.144)
xefe  pts/0   Jan  4 08:46 ➥
(10.2.4.144)
...
xefe  pts/0   Jan  4 08:46 ➥
(10.2.4.144)
```

Isto é, a cada 30 segundos a saída do comando grep seria enviada para a tela, o que não é legal, já que poluiria a tela do meu micro e a mensagem tão esperada poderia passar despercebida. Para evitar isso, já sabemos que a saída do pipeline tem que ser redirecionada para o dispositivo */dev/null*.

```
$ cat logaute.sh
#!/bin/bash
# Espero que a Xuxa não tenha
# copyright de xefe e xato :)
while who | grep xefe > /dev/null
do
    sleep 30
done
echo O xato se mandou, não ➥
hesite, dê exit e vá a luta
```

Agora quero montar um script que receba o nome (e eventuais parâmetros) de um programa que será executado em background e que me informe do seu término. Mas, para você entender este exemplo, primeiro tenho de mostrar uma nova variável do sistema. Veja estes comandos executados diretamente no prompt:

```
$ sleep 10&
[1] 16317
$ echo $!
16317
[1]+  Done                    sleep 10
$ echo $!
16317
```

Isto é, criei um processo em background que dorme por 10 segundos, somente para mostrar que a variável *$!* guarda o PID (*Process ID*) do último processo em background. Mas observe a listagem e repare, após a linha do *Done*, que a variável reteve o valor mesmo após o término desse processo.

Bem, sabendo isso, já fica mais fácil monitorar qualquer processo em background. Veja só como:

```
$cat monbg.sh
#!/bin/bash
# Executa e  monitora um
# processo em background
$1 &     # Coloca em backgroud
while ps | grep -q $!
do
    sleep 5
done
echo Fim do Processo $1
```

Esse script é bastante similar ao anterior, mas tem uns macetes a mais, veja só: ele tem que ser executado em background para não prender o prompt mas o *$!* será o do programa passado como parâmetro, já que ele foi colocado em background após o *monbg.sh* propriamente dito. Repare também na opção *-q* (quiet) do grep, que serve para fazê-lo "trabalhar em silêncio". O mesmo resultado poderia ser obtido com a linha: *while ps | grep $! > /dev/null*, como nos exemplos que vimos até agora.

Vamos melhorar o nosso velho *musinc*, nosso programa para incluir registros no arquivo *musicas*, mas antes preciso te ensinar a pegar um dado da tela, e já vou avisando: só vou dar uma pequena dica do comando read (que é quem pega o dado da tela), que seja o suficiente para resolver este nosso problema. Em uma outra rodada de chope vou te ensinar tudo sobre o assunto, inclusive como formatar tela, mas hoje estamos falando sobre loops. A sintaxe do comando read que nos interessa por hoje é a seguinte:

```
$ read -p "prompt de leitura" var
```

Onde "prompt de leitura" é o texto que você quer que apareça escrito na tela. Quando o operador teclar tal dado, ele será armazenado na variável *var*. Por exemplo:

```
$ read -p "Título do Álbum: " Tit
```

Bem, uma vez entendido isso, vamos à especificação do nosso problema: faremos um programa que inicialmente lerá o nome do álbum e em seguida fará um loop de leitura, pegando o nome da música e o artista. Esse loop termina quando for informada uma música com nome vazio, isto é, quando o operador

Dica

Leitura de arquivo significa ler um a um todos os registros, o que é sempre uma operação lenta. Fique atento para não usar o while quando for desnecessário. O Shell tem ferramentas como o sed e a família grep, que vasculham arquivos de forma otimizada sem que seja necessário o uso do while para fazê-lo registro a registro.

der um simples <ENTER>. Para facilitar a vida do operador, vamos oferecer como default o mesmo nome do artista da música anterior (já que é normal que o álbum seja todo do mesmo artista) até que ele deseje alterá-lo. Veja na listagem 1 como ficou o programa.

Nosso exemplo começa com a leitura do título do álbum. Caso ele não seja informado, terminamos a execução do programa. Em seguida um *grep* procura, no início (^) de cada registro de músicas, o título informado seguido do separador (^) (que está precedido de uma contrabarra [\] para protegê-lo da interpretação do Shell).

Para ler os nomes dos artistas e as músicas do álbum, foi montado um loop *while* simples, cujo único destaque é o fato de ele armazenar o nome do intérprete da música anterior na variável *$oArt*, que só terá o seu conteúdo alterado quando algum dado for informado para a variável *$Art*, isto é, quando não for teclado um simples *ENTER* para manter o artista anterior.

O que foi visto até agora sobre o *while* foi muito pouco. Esse comando é muito utilizado, principalmente para leitura de arquivos, porém ainda nos falta bagagem para prosseguir. Depois que aprendermos mais sobre isso, veremos essa instrução mais a fundo.

O comando *until*

Este comando funciona de forma idêntica ao *while*, porém ao contrário. Disse tudo mas não disse nada, né? É o seguinte: ambos testam comandos; ambos possuem a mesma sintaxe e ambos atuam em loop; porém, o *while* executa o bloco de instruções do loop enquanto um comando for bem sucedido; já o until executa o bloco do loop até que o comando seja bem sucedido. Parece pouca coisa, mas a diferença é fundamental. A sintaxe do comando é praticamente a mesma do while. Veja:

```
until comando
do
    cmd1
    cmd2
    ...
    cmdn
done
```

e dessa forma o bloco de comandos formado pelas instruções *cmd1, cmd2,...* e *cmdn* é executado até que a execução da instrução *comando* seja bem sucedida.

Como eu te disse, *while* e *until* funcionam de forma antagônica, e isso é muito fácil de demonstrar: em uma guerra, sempre que se inventa uma arma, o inimigo busca uma solução para neutralizá-la. Foi baseado nesse principio belicoso que meu chefe desenvolveu, no mesmo servidor em que eu executava o *logaute.sh*, um script para controlar o meu horário de chegada.

Um dia tivemos um problema na rede. Ele me pediu para dar uma olhada no micro dele e me deixou sozinho na sala. Resolvi bisbilhotar os arquivos – guerra é guerra – e veja só o que descobri:

```
$cat chegada.sh
#!/bin/bash
until who | grep julio
do
    sleep 30
done
echo $(date "+ Em %d/%m às ➥
%H:%Mh") > relapso.log
```

Olha que safado! O cara estava montando um log com os meus horários de chegada, e ainda por cima chamou o arquivo de *relapso.log*! O que será que ele quis dizer com isso?

Nesse script, o pipeline *who | grep julio*, será bem sucedido somente quando *julio* for encontrado na saída do comando *who*, isto é, quando eu me "logar" no servidor. Até que isso aconteça, o comando *sleep*, que forma o bloco de instruções do *until*, colocará o programa em espera por 30 segundos. Quando esse loop encerrar-se, será enviada uma mensagem para o arquivo *relapso.log*. Supondo que no dia 20/01 eu me "loguei" às 11:23 horas, a mensagem seria a seguinte:

Listagem 1

```
$ cat musinc.sh
#!/bin/bash
# Cadastra CDs (versao 4)
#
clear
read -p "Título do Álbum: " Tit
[ "$Tit" ] || exit 1 # Fim da execução se título vazio
if grep "^$Tit\^" musicas > /dev/null
then
    echo "Este álbum já está cadastrado"
    exit 1
fi
Reg="$Tit^"
Cont=1
oArt=
while true
do
    echo "Dados da trilha $Cont:"
    read -p "Música: " Mus
    [ "$Mus" ] || break       # Sai se vazio
    read -p "Artista: $oArt // " Art
    [ "$Art" ] && oArt="$Art" # Se vazio Art anterior
    Reg="$Reg$oArt~$Mus:"     # Montando registro
    Cont=$((Cont + 1))
    # A linha anterior tb poderia ser ((Cont++))
done
echo "$Reg" >> musicas
sort musicas -o musicas
```

Em 20/01 às 11:23h

Voltando à nossa CDteca, quando vamos cadastrar músicas seria ideal que pudéssemos cadastrar diversos CDs de uma vez só. Na última versão do programa isso não ocorre: a cada CD cadastrado o programa termina. Veja na listagem 2 como melhorá-lo.

Nesta versão, um loop maior foi adicionado antes da leitura do título, que só terminará quando a variável *$Para* deixar de ser vazia. Caso o título do álbum não seja informado, a variável *$Para* receberá um valor (coloquei 1, mas poderia ter colocado qualquer coisa) para sair desse loop, terminando o programa. No resto, o script é idêntico à versão anterior.

Atalhos no loop

Nem sempre um ciclo de programa, compreendido entre um *do* e um *done*, sai pela porta da frente. Em algumas oportunidades, temos que colocar um comando que aborte de forma controlada esse loop. De maneira inversa, algumas vezes desejamos que o fluxo de execução do programa volte antes de chegar ao *done*. Para isso, temos respectivamente os comandos *break* (que já vimos rapidamente nos exemplos do comando *while*) e *continue*, que funcionam da forma mostrada na figura 1.

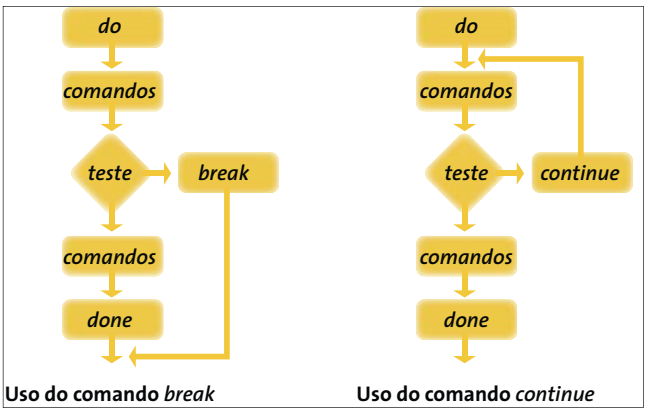


Figura 1: A estrutura dos comandos *break* e *continue*, usados para controlar o fluxo de execução em loops.

O que eu não havia dito anteriormente é que nas suas sintaxes genéricas eles aparecem da seguinte forma:

```
break [qtd loop]
```

e também:

```
continue [qtd loop]
```

Onde *qtd loop* representa a quantidade dos loops mais internos sobre os quais os comandos irão atuar. Seu valor por *default* é 1.

Duvido que você nunca tenha apagado um arquivo e logo após deu um tabefe na testa se xingando porque não devia tê-lo removido. Pois é, na décima vez que fiz esta besteira, criei um script para simular uma lixeira, isto é, quando mando remover um (ou vários) arquivo(s), o programa "finge" que deletou, mas no duro o que ele fez foi mandá-lo(s) para o diretório /tmp/LoginName_do_usuario. Chamei esse programa de *erreeme* e no arquivo */etc/profile* coloquei a seguinte linha, que cria um "apelido" para ele:

```
alias rm=erreeme
```

Veja o programa na listagem 3. Como você pode ver, a maior parte do script é formada por pequenas críticas aos parâmetros informados, mas como o script pode ter recebido diversos arquivos a remover, a cada arquivo que não se encaixa dentro do especificado há

Listagem 2

```
$ cat musinc.sh
#!/bin/bash
# Cadastra CDs (versao 5)
#
Para=
until [ "$Para" ]
do
    clear
    read -p "Título do Álbum: " Tit
    if  [ ! "$Tit" ] # Se titulo vazio...
    then
        Para=1     # Liguei flag de saída
    else
        if grep "^$Tit\^" musicas > /dev/null
        then
            echo "Este álbum já está cadastrado"
            exit 1
        fi
        Reg="$Tit^"
        Cont=1
        oArt=
        while [ "$Tit" ]
        do
            echo Dados da trilha $Cont:
            read -p "Música: " Mus
            [ "$Mus" ] || break       # Sai se vazio
            read -p "Artista: $oArt // " Art
            [ "$Art" ] && oArt="$Art" # Se vazio Art anterior
            Reg="$Reg$oArt~$Mus:"     # Montando registro
            Cont=$((Cont + 1))
            # A linha anterior tb poderia ser ((Cont++))
        done
        echo "$Reg" >> musicas
        sort musicas -o musicas
    fi
done
```

Listagem 3: erreeme.sh

```
$ cat erreeme.sh
#!/bin/bash
#
# Salvando cópia de um arquivo antes de removê-lo

# Tem de ter um ou mais arquivos a remover
if  [ $# -eq 0 ]
then
    echo "Erro -> Uso: erreeme arq [arq] ... [arq]"
    echo "O uso de metacaracteres e' permitido. Ex.➥
erreeme arq*"
    exit 1
fi

# Variável do sistema que contém o nome do usuário.
MeuDir="/tmp/$LOGNAME"
# Se não existir o meu diretório sob o /tmp...
if  [ ! -d $MeuDir ]
then
    mkdir $MeuDir      # Vou criá-lo
fi

# Se não posso gravar no diretório...
if  [ ! -w $MeuDir ]
then
    echo "Impossivel salvar arquivos em $MeuDir. ➥
Mude as permissões..."
    exit 2
fi

# Variável que indica o cod. de retorno do programa
Erro=0
# Um for sem o in recebe os parametros passados
for Arq
do
# Se este arquivo não existir...
    if  [ ! -f $Arq ]
    then
        echo "$Arq nao existe."
        Erro=3
        continue       # Volta para o comando for
    fi

# Cmd. dirname informa nome do dir de $Arq
    DirOrig=`dirname $Arq`
# Verifica permissão de gravacao no diretório
    if  [ ! -w $DirOrig ]
    then
        echo "Sem permissão no diretorio de $Arq"
        Erro=4
        continue         # Volta para o comando for
    fi

# Se estou "esvaziando a lixeira"...
    if  [ "$DirOrig" = "$MeuDir" ]
    then
        echo "$Arq ficara sem copia de seguranca"
        rm -i $Arq      # Pergunta antes de remover
        # Será que o usuário removeu?
        [ -f $Arq ] || echo "$Arquivo removido"
        continue
    fi

# Guardo no fim do arquivo o seu diretório original➥
para usá-lo em um script de undelete
    cd $DirOrig
    pwd >> $Arq
    mv $Arq $MeuDir  # Salvo e removo
    echo "$Arq removido"
done

# Passo eventual número do erro para o código
# de retorno
exit $Erro
```

um *continue*, para que a seqüência volte para o loop do *for* de forma a receber outros arquivos.

Quando você está no Windows (com perdão da má palavra) e tenta remover aquele monte de lixo com nomes esquisitos como HD04TG.TMP, se der erro em um dos arquivos os outros não são removidos, não é? Então, o *continue* foi usado para evitar que uma impropriedade dessas ocorra, isto é, mesmo que dê erro na remoção de um arquivo, o programa continuará removendo os outros que foram passados.

– Eu acho que a esta altura você deve estar curioso para ver o programa que restaura o arquivo removido, não é? Pois então aí vai vai um desafio: faça-o em casa e me traga para discutirmos no nosso próximo encontro aqui no boteco.
– Poxa, mas nesse eu acho que vou dançar, pois não sei nem como começar...
– Cara, este programa é como tudo o que se faz em Shell: extremamente fácil. É para ser feito em, no máximo, 10 linhas. Não se esqueça de que o arquivo está salvo em */tmp/$LOGNAME* e que sua última linha é o diretório em que ele residia antes de ser "removido". Também não se esqueça de criticar se foi passado o nome do arquivo a ser removido.
– É eu vou tentar, mas sei não...
– Tenha fé, irmão, eu tô te falando que é mole! Qualquer dúvida é só passar um email para julio.neves@gmail.com. Agora chega de papo que eu já estou de goela seca de tanto falar. Me acompanha no próximo chope ou já vai sair correndo para fazer o script que passei?
– Deixa eu pensar um pouco...
– Chico, traz mais um chope enquanto ele pensa!

SOBRE O AUTOR

Julio Cezar Neves é Analista de Suporte de Sistemas desde 1969 e trabalha com Unix desde 1980, quando participou do desenvolvimento do SOX, um sistema operacional similar ao Unix produzido pela Cobra Computadores. Pode ser contatado no e-mail julio.neves@gmail.com

Inclusão digital e economia

Caminho livre para a Administração Pública

A Fundação Centro de Tecnologia Aplicada vem atuando junto a prefeituras e outros órgãos municipais, através do seu Núcleo de Pesquisa e Desenvolvimento (NuPD), criando soluções em software para a administração pública. POR JOE KLIMENT

Em 2002, o NuPD iniciou um processo de avaliação junto a municípios no Estado do Rio de Janeiro com o objetivo de identificar a real situação das prefeituras em estrutura de hardware, software, rede e conectividade. Essa avaliação evidenciou a existência de uma situação comum à maioria dos municípios: por causa da falta de recursos para implementação de um Plano Diretor de Informática apropriado, as prefeituras se viam às voltas com redes de transmissão de dados ineficientes, hardware inadequado ao sistema operacional proprietário utilizado e utilização de softwares e aplicativos de gerenciamento proprietários que pecavam pela falta de integração, devido à inexistência ou inadequação da estrutura de rede. O acesso à Internet muitas vezes era obtido através de uma linha discada; o suporte e a manutenção dos sistemas eram quase sempre feitos por empresas terceirizadas, que precisavam deslocar técnicos para prestar suporte no local, aumentando o custo dos serviços. Sem contar o fato de que, muitas vezes, havia softwares com licenciamento inadequado ou inexistente.

Ficou evidente que qualquer solução realmente viável para a administração municipal deveria necessariamente possuir características de custo e desempenho bastante diferentes das tradicionalmente utilizadas. De posse dessa avaliação realizada em campo, a Fundação CTA foi buscar no mercado um profissional para dirigir o projeto de remodelação da estratégia de desenvolvimento de sistemas para a área pública. O projeto foi então apresentado a mim. De posse da pesquisa realizada pela Fundação, e já familiarizado com o Linux e as realidades do software livre, o NuPD

Figura 1: Joe Kliment – Diretor de TI da Fundação CTA.

apresentou ao CTA uma idéia inovadora: desenvolver todos os softwares necessários para a administração pública utilizando uma plataforma web, a linguagem de programação PHP, o banco de dados livre PostgreSQL – e rodar tudo isso em servidores Linux. Os desktops das prefeituras deveriam, preferencialmente, utilizar o sistema operacional Linux. Aplicativos de processamento de texto e planilhas eletrônicas deveriam também ser Software Livre.

A idéia era fazer com que as prefeituras pudessem melhorar a capacidade de utilização de seu parque de hardware com um sistema operacional que exigisse menos recursos, evitar o pagamento desnecessário de licenças de software básico e de banco de dados e construir um sistema de administração pública que pudesse ser aprendido com facilidade pelo funcionalismo e que tivesse a possibilidade de receber suporte e manutenção remota, de forma rápida, o que baixaria ainda mais os custos operacionais. Faltava colocar o projeto em prática.

Aproveitando a existência de uma escola técnica gerida pela Fundação em Padre Miguel, na Zona Oeste do Rio de Janeiro, foi montado um laboratório de informática a baixíssimo custo, utilizando a tecnologia de terminais baseados em Linux LTSP (www.ltsp.org). Nesse laboratório foram ministrados cursos de Linux, PHP e Bancos de Dados. Os cursos eram apresentados à comunidade a preço de custo; os alunos do curso de PHP, por exemplo, pagavam 20 Reais ao mês para aprender a linguagem de programação. O resultado foi assombroso. Como os alunos aprendiam rápido, os técnicos do NuPD – e também professores – sugeriram que a Fundação contratasse os formandos do curso. Os resultados não tardaram a aparecer.

Figura 3: Delmo Morani – Diretor de Relações Institucionais da CTA.

A equipe de analistas de sistemas do CTA começou então a executar o projeto. Montaram-se equipes de trabalho; alunos da escola eram inseridos nessas equipes e trabalhavam junto com os técnicos do NuPD, adquirindo a experiência necessária para o mercado de trabalho. Os softwares foram desenvolvidos e testados exaustivamente. Tendo elegido a cidade de Resende, na região sul Fluminense, como palco para o lançamento do projeto, o CTA forneceu treinamento em Linux e Software Livre às equipes de processamento de dados da prefeitura. Os sistemas administrativos foram instalados e entraram em produção, e os funcionários aprenderam a operá-los em um curto período. Quem sabia utilizar um navegador Web sabia usar o sistema, o que diminuiu drasticamente a curva de aprendizado e os custos de treinamento.

Figura 2: Alunos do CTA premiados pelo prefeito da cidade de Resende.

Em Maio de 2003 ocorreu o I Encontro de Governo Eletrônico, totalmente patrocinado pelo CTA, na cidade de Resende-RJ. Com a presença de cerca de 300 autoridades do Estado do Rio de Janeiro, foram apresentadas as soluções desenvolvidas. Os estandes com sistemas de Compras e Licitações, Protocolo, Pregão Eletrônico e Planejamento e Execução Orçamentária foram especialmente concorridos. Na ocasião, num gesto de agradecimento e reconhecimento pelo esforço dos alunos do CTA, o prefeito da cidade os agraciou com medalhas (ver figura 2). Foi um momento inesquecível para aqueles meninos e meninas de Padre Miguel, uma região carente do Rio de Janeiro.

Hoje o CTA dispõe de sistemas completos de administração pública, incluindo contabilidade, tesouraria, orçamento, licitações, protocolo e muitos outros. Um caso de sucesso, que demonstra a capacidade dos profissionais de informática brasileiros para aplicar os recursos disponíveis em Linux e Software Livre para o benefício de todos, gerando emprego e trazendo soluções para quem precisa. ■

Poloneses salvam Europa das Patentes de Software

Obrigado, Polônia!

Não posso deixar de agradecer o ato bastante positivo e de muita coragem da Polônia. Graças a ela, a Europa está livre das patentes de software. Aproveitando que o assunto virou notícia no mundo todo, vamos falar um pouco sobre patentes e os problemas que podem trazer se não forem bem aplicadas ou caírem nas mãos erradas. POR CHRISTIANO ANDERSON

O Sr. Wlodzimierz Marcinski, ministro da ciência e tecnologia da informação da Polônia (membro da União Européia desde 2004), foi até Bruxelas, na Bélgica, tratar de um assunto polêmico: as patentes de software. O ministro, em um ato corajoso, conseguiu anular a adoção das patentes de software para todos os países que compõem o quadro da UE [1].

Essa medida é bastante positiva, principalmente para o Software Livre. As grandes corporações detêm boa parte das patentes de software de todo o mundo, o que prejudica o desenvolvimento de programas por empresas de pequeno e médio porte. Isto não quer dizer que a Europa está livre do problema das patentes, mas o assunto foi adiado até uma nova reunião sobre o tema. Enquanto isso, aqueles que são contra as patentes terão mais tempo para pensar em uma alternativa ou se unir e cancelar de vez o assunto.

O mais estranho é que a maioria dos países da UE evitava comentar o tema; alguns chegaram a dizer que eram contra as patentes de software, mas não fizeram nada para interromper o processo – pelo contrário, votaram a favor. O lobby sobre esse assunto é enorme. Assim como há políticos que são contra a idéia, outros são a favor. É uma briga longa e temos que nos unir para não deixar o problema chegar até nós.

Figura 1: A União Européia é o novo front na batalha contra as patentes de software.

Entenda melhor o assunto

O que o leitor deve estar pensando é: o que nós, brasileiros, temos a ganhar com isso? Certamente temos muito o que comemorar e aprender com experiências desse tipo, nem que seja para evitar que o mesmo aconteça no Brasil. Não, não estamos totalmente livres desse problema; ele poderá acontecer a qualquer momento. O governo brasileiro parece estar no rumo certo deixando a ALCA em segundo plano – isso sim pode ser uma grande dor de cabeça para empresas que trabalham com software e desenvolvimento.

Enquanto o Brasil não aceitar a ALCA da maneira que foi estabelecida, estaremos livre de problemas com as patentes de software. Caso o Brasil entre no grupo, estaremos incorporando as patentes já existentes nos EUA, o que pode prejudicar o desenvolvimento de um modelo de software nacional e independente – coisa que, diga-se de passagem, já vem ocorrendo. Perderíamos muito com a ALCA: nossas empresas perderiam o poder de barganhar novos mercados e seríamos afetados por toda a política de patentes e copyrights existente na América do Norte.

Vamos pensar um pouco no atual modelo de proteção, que serve tanto para software quanto para música, textos etc. Vamos nos concentrar no software, que é nossa principal ferramenta. Quando o desenvolvedor cria um programa qualquer, automaticamente os direitos autorais pertencem a esse autor. É ele quem vai decidir qual licença utilizar, se vai liberar o programa sob alguma licença livre ou se vai ceder os direitos a algum outro autor ou empresa. Essa decisão é somente dele. Ninguém mais poderá interferir nesse trabalho, a não ser que ele permita. Se alguém fizer uma cópia ou uso não autorizado deste software, o autor poderá entrar com recursos legais para cobrar por esse uso indevido. Resumindo: o autor está protegido. O ponto mais irônico é que esse modelo foi criado pelas megacorporações, como Microsoft, Oracle, IBM etc. Quando essas empresas criam seus softwares, o direito de uso está diretamente relacionado ao modelo de licenciamento dessas empresas. Isso tudo é chamado de direitos autorais.

Então, o que as patentes têm a ver com isso?

As patentes são mecanismos para uma empresa poderosa ficar ainda mais poderosa. Alguns escritórios de advocacia são especializados em patentes; detêm os direitos de milhares de aplicativos registrados e o que querem é obter mais dinheiro e poder. Os direitos autorais são incorporados automaticamente no momento da criação de um aplica-

tivo; o desenvolvedor não precisa se preocupar com registrar ou correr atrás de uma empresa que faça isso para ele. Segundo o INPI[2], os programas de computador são protegidos pelo Direito Autoral e não pelo Direito Patentário. Com isso, concluímos que o modelo válido na maioria dos países do globo é também válido no Brasil.

O que as empresas especializadas em patentes querem é poder para processar desenvolvedores e empresas que fazem softwares que venham a ferir alguma patente já registrada por seus clientes. Estas empresas querem extorquir dinheiro de pessoas inocentes, que só desejam desenvolver seus aplicativos. Querem anular qualquer concorrência para seus clientes, que já possuem inúmeras patentes registradas. Querem aumentar o monopólio e prejudicar pessoas honestas e trabalhadoras.

O modelo de patentes pode ser como um campo minado para um programador; ele terá de seguir um rumo e tomar cuidados para não pisar em bombas ou encostar em cercas de arame farpado. O caminho das patentes é árduo, diferente daquilo a que estamos acostumados: ter uma idéia, sentar na frente do computador, colocar uma boa música para tocar e fazer nosso software, sem nos preocuparmos com estar fazendo algo ilegal. Atualmente, na Europa, mais de 30 mil patentes de software já foram registradas e há outras milhares na fila aguardando registro.

A Foundation for a Free Information Infrastructure (FFII)[3] criou uma loja virtual [4] (figura 2) que possui mais de 20 itens patenteados, isso só na página principal. A loja possui itens como a palavra "Shopping Cart" (Carrinho de Compras), patenteada pela Sun Microsystens e "Picture Link", patente da IBM, entre outras. O site é apenas uma simulação, mas serve como exemplo do que poderia acontecer: e se todas as empresas citadas entrassem com um processo contra o autor da loja virtual? E se o autor fosse obrigado a pagar pelos termos utilizados em seu website?

Imagine se um desenvolvedor de software livre tivesse que se preocupar com cada detalhe, cada item que eventualmente pudesse ferir uma patente? Esse exemplo da FFII mostra muito bem como funciona o modelo perigoso.

Caso o modelo fosse válido no Brasil, o desenvolvedor seria obrigado a consultar uma base de dados de patentes para se certificar de que tudo o que está utilizando já não esteja patenteado. O pior disso tudo é ter de criar patentes sobre seus itens para não correr o risco de um espertinho registrá-los antes e depois ainda vir cobrar direitos sobre seus próprios produtos. É um assunto bastante complexo. ■

Figura 2: Infelizmente, a loja virtual patenteada criada pela FFII pode se tornar uma triste realidade.

INFORMAÇÕES

[1] *http://pt.wikipedia.org/wiki/União_Europeia*
[2] *http://www.inpi.gov.br*
[3] *http://www.ffii.org/*
[4] *http://webshop.ffii.org/*

SOBRE O AUTOR

Christiano Anderson (anderson@gnu.org) é desenvolvedor autônomo de Software Livre, participa do Projeto Software Livre Brasil (PSL-BR) e contribui com o Projeto GNU. Sua principal missão é difundir o Software Livre e sua filosofia, principalmente no mercado educacional.

Eventos em Destaque

II Fórum Maranhense de Software Livre

Data: 13 de Abril de 2005
Local: Imperatriz, Maranhão
Website: -

III CONISLI

Data: 10 de Novembro de 2005
Local: São Paulo, São Paulo
Website:
www.conisli.org.br

Telefonia IP com Software Livre

Data: 30 de Abril de 2006
Local: São Paulo, São Paulo
Website:
eventos.temporeal.com.br/?area=5

Calendário de Eventos

Evento	Data	Local	Website
II Festival de Software Livre da Bahia	11 a 13 de Março	Salvador, Bahia	festival.softwarelivre.org
II Fórum Maranhense de Software Livre	13 de Abril	Imperatriz, MA	–
III CONISLI	10 de Novembro	São Paulo, SP	www.conisli.org.br
II Latinoware	27 de Março, 2006	Foz do Iguaçu, PR	www.latinoware.org
FOSDEM 2005	26 a 27 de Fevereiro	Bruxelas, BE	www.fosdem.org
Gentoo UK 2005	12 de Março	Manchester, UK	dev.gentoo.org/~stuart/2005
LinuxCorp	05 e 06 de Julho	São Paulo, SP	www.rpmbrasil.com.br
TubaInstall Linux Fest	05 de Março	São Caetano, SP	nxpangea.no-ip.org/installfest
Telefonia IP com Software Livre	30 de Abril	São Paulo, SP	eventos.temporeal.com.br/?area=5

Índice de Anunciantes

Anunciante	Site	Página
4Linux	www.4linux.com.br	95
A Casa do Linux	www.casadolinux.com.br	95
AS Informática	www.asinformática.com.br	95
Alternativa Linux	www.alternativalinux.com.br	15
Green Treinamento e Consultoria	www.green.com.br	73
Infomedia	www.infomediatv.com.br	17
IBM	www.ibm.com.br	99 (terceira capa)
Linux Magazine	www.linuxmagazine.com.br	51, 95
Linux Professional Institute	www.lpi.com.org	95
MySQL Brasil	www.mysqlbrasil.com.br	02 (contra-capa)
Novatec Editora	www.novateceditora.com.br	95
Revera_	www.reveralinux.com.br	9, 29
Oracle	www.oracle.com.br	100 (quarta capa)
Visuelles	www.visuelles.com.br	79
Unicial / LinuxPress	www.unicial.com.br	47
utah	www.utah.com.br	95

Escreva para a Linux Magazine

Estamos sempre à procura de novos artigos e autores. Se você acha que um assunto é importante, ou que precisa ser melhor explicado, fale conosco.

Precisamos de tutoriais, análises, estudos de caso e notícias. Se você é membro de um grupo de usuários, porque não nos conta sobre os eventos que estão sendo planejados? Preferimos que os artigos sejam enviados via e-mail, e screenshots são sempre bem-vindos. Para facilitar as coisas, mencione no assunto de sua mensagem o tema do artigo.

Artigos têm em média 4.500 caracteres por página (contando os espaços), mas listagens de código e imagens reduzem o espaço disponível para o texto. Se possível, escreva páginas inteiras. Como estamos presentes em vários países, evite o uso de gírias e expressões regionais.

As imagens devem estar na maior resolução possível. No caso de fotos digitais, recomendamos que elas sejam tiradas com uma câmera de 3 Megapixels e resolução igual ou superior a 1024x768 pixels. Prefira formatos como TIF e EPS.

Uma revista passa por muitos estágios durante sua produção, portanto alguns meses podem se passar desde que seu artigo seja recebido até que a revista chegue às bancas. Portanto, nunca envie na última hora material ou notícias sobre encontros e eventos.

Envie suas colaborações para o endereço *material@linuxmagazine.com.br*. Evite enviar mais de 4 MB em arquivos anexos. Caso o material para seu artigo ultrapasse esse limite, coloque-o em algum site na Internet e nos informe a URL.

Expediente Editorial

Editores
Rafael Antonio Guido Peregrino da Silva:
rperegrino@linuxmagazine.com.br
Rafael Pereira Rigues:
rrigues@linuxmagazine.com.br

Centros de Competência
Centro de Competência em Software:
Oliver Frommel, *ofrommel@linux-magazine.com*
Centro de Competência em Hardware:
Mirko Dölle, *mdoelle@linux-magazine.com*
Centro de Competência em Redes e Segurança:
Achim Leitner, *aleitner@linux-magazine.com*

Correspondentes & Colaboradores
Achim Leitner, Alexandre Barbosa, Andrea Mueller, Augusto Campos, Christian Ney, Christiano Anderson, Fabrizio Ciacchi, Hagen Höpfner, Holger Junge, James Mohr, Jason Walsh, Joe Casad, Joe Kliment, Julio Cezar Neves, Oliver Frommel, Peter Busser, Rafael Rigues, Rafael Peregrino da Silva, Ralf Spenneberg, Sulamita Garcia, Zack Brown.

Confecção do CD-ROM
Ricardo D. Dalceno
rdalceno@linuxmagazine.com.br

Tradução e Revisão
Henrique Cesar Ulbrich: *henrique.cesar@ermida.com*
Julia Vidili: *julia.vidili@ermida.com*

Direção de Arte
Luciano Hagge Dias:
lhagge@linuxmagazine.com.br

Design da Capa
Pinball: *info@pinball-werbeagentur.de*

Anúncios: *www.linuxmagazine.com.br/Advertise*

Brasil
Unicial – União Comercial e Editora Ltda.
A/c. Sérgio Alday
anuncios@linuxmagazine.com.br
Tel./Fax: +55 (0)11 3051 3159
Cel.: +55 (0)11 9935 1368

Todos os países (exceto Alemanha, Áustria e Suíça)
Brian Osborn, *ads@linux-magazine.com*
Tel.: +49 (0)6509 910 495
Fax: +49 (0)6509 910 497

Alemanha, Áustria e Suíça
Osmund Schmidt
anzeigen@linux-magazine.com
Tel.: +49 (0)6335 9110
Fax: +49 (0)6335 7779

Diretoria
Rafael Antonio Guido Peregrino da Silva,
rperegrino@linuxmagazine.com.br

Linux Magazine
Av. Engº. Luís Carlos Berrini, 1500
Conj. 103 – Brooklin Novo
04571-000 – São Paulo – SP – Brasil
Tel.: +55 (0)11 3345 1002
Fax: +55 (0)11 3345 1081

Assinaturas: www.linuxmagazine.com.br/Subs
Preço: (12 edições incluindo CD mensal): R$143,00
Email: assinaturas@linuxmagazine.com.br

Preço Unitário: R$14,90

Na Internet:
www.linuxmagazine.com.br – Brasil
www.linux-magazin.de – Alemanha
www.linux-magazine.com – Portal Mundial
www.linuxmagazine.com.au – Austrália
www.linux-magazine.ca – Canadá
www.linux-magazine.co.uk – Reino Unido
www.linux-magazine.pl – Polônia
www.linux-magazin.ro – Romênia

Linux Magazine é publicada mensalmente por:
Linux New Media do Brasil Editora Ltda.,
São Paulo/SP, Brasil.

Distribuído por Fernando Chinaglia Distribuidora.

Impressão e Acabamento: Editora Gráficos Burti LTDA

ISSN 1806-9428 Impresso no Brasil

INSTITUTO VERIFICADOR DE CIRCULAÇÃO
em processo de filiação

Sétima edição

Na próxima edição...

Ubuntu

Ubuntu é uma palavra do idioma Nguni, falado na África do Sul, que significa "tratar humanamente ao próximo" ou "sou o que sou por aquilo que todos nós somos". É também o nome de uma distribuição Linux recém-chegada que tem chamado muita atenção por sua ênfase na usabilidade e integração de tecnologias que facilitam a vida do usuário, como o Gnome 2.8, excelente detecção de hardware e montagem automática de mídia removível, como cartões de memória e máquinas fotográficas digitais. Veja em nosso artigo como é o processo de instalação, como funciona o suporte a hardware mais antigo, a quantas anda o suporte ao idioma Português do Brasil e muito mais.

Gimp 2.2

Depois da avalanche de mudanças na versão 2.0 (apresentadas ao leitor em nossa quinta edição), chegou a hora de conhecer os ajustes e novidades do Gimp 2.2. Interface refinada, previews em tempo real para vários filtros, um editor de atalhos de teclado e filtros novos como *Cartoon* e *Photocopy* são algumas das mudanças que vale a pena conhecer.

Syslog, a próxima geração

Por mais útil e onipresente que o syslog seja, já começa a mostrar sua idade. Sistemas Unix modernos (e assemelhados) são consideravelmente mais complexos do que eram quando ele foi inventado. Os limitados recursos de rede e registro do syslog já não são mais suficientes para atender às exigências de rebustez, flexibilidade e velocidade atuais – sem falar na segurança.

O Syslog-ng ("Syslog, a nova geração"), criado e mantido por Balazs Scheidler, é uma tentativa de aumentar a flexibilidade do syslog, com a adição de melhor filtragem de mensagens, melhores recursos de rede e, por fim, melhor criptografia e verificação de integridade das mensagens. No artigo em nossa próxima edição você aprenderá a instalar e configurar um servidor de logs baseado nesta poderosa ferramenta.

Redes sem fio

Redes sem fio são muito práticas – depois de terem sido corretamente configuradas. Mas a variedade de dispositivos, além dos problemas para integrar o hardware wireless com seu software ou sistema operacional, pode intimidar até mesmo os mais experientes administradores. Vamos mostrar a você algumas estratégias de configuração de uma máquina Linux para se conectar a redes wireless.

Os artigos de nossa matéria de capa sobre redes sem fio abrangem um grande conjunto de tópicos sobre sistemas wireless no Linux. Começamos com uma olhada nos fundamentos dessas redes. Os artigos seguintes discutem roteadores DSL WLAN e adaptadores wireless USB. Terminamos mostrando como configurar uma rede privada usando OpenVPN em uma rede wireless, mas o tema continua. Apresentamos o KWiFiManager, um utilitário para o KDE que ajuda a monitorar e gerenciar suas conexões à redes WiFi; ensinamos também a montar um sistema de autenticação via Radius para suas redes locais, inclusive redes wireless.


CD Número 06
ano II
LINUX
MAGAZINE
simply change
suse
COMPACT
disc
EXCLUSIVO!
versão instalável
9.2
em português
OpenOffice.org em português
Kernel 2.6.8
KDE 3.3
CD Número 06
ano II
Aviso: Este CD-ROM da Linux Magazine foi testado extensivamente e até onde pudemos verificar se encontra livre de qualquer vírus ou outro tipo de software de conteúdo malicioso, bem como de defeitos. A Linux Magazine não se responsabiliza por qualquer dano ou perda de dados que advenha da utilização deste CD-ROM ou de software nele incluído. Este CD é parte integrante da revista Linux Magazine nº06 e não pode ser vendido separadamente.
Fabricado por Sonopress-Rimo Indústria e Comércio Fonográfica LTDA - CNPJ: 67.562.884/0001-49 – Indústria Brasileira - sob encomenda de Linux New Media LTDA - 06.351.943/0001-45