CAMPUS PARTY p.24
A cultura digital é forte no Brasil

VOZ DA JUSTIÇA p. 22
Ampla implantação de VoIP no Tribunal de Justiça de SC

VOIP PROBLEMÁTICO p.30
No Brasil, o mercado de VoIP ainda tem problemas e falta de leis

LINUX NEW MEDIA
The Pulse of Open Source

51 Fevereiro 2009

LINUX MAGAZINE

A REVISTA DO PROFISSIONAL DE TI

VoIP SEGURO

A NOVA MODALIDADE DA TELEFONIA NÃO É UM PRIMOR DE SEGURANÇA. CONHEÇA AS BRECHAS DO SIP E DO RTP E APRENDA A REMEDIÁ-LAS ANTES QUE OS BANDIDOS O FAÇAM p.33

- SIP e RTP são inseguros. Há alternativas seguras? p.40
- Senhas descartáveis afastam invasores p.34
- Wi-Fi livre de intrusos p.46

SEGURANÇA: NMAP PARA QUÊ? p.68

Conheça o hping e veja como ele complementa (e até supera) os recursos diagnósticos do célebre Nmap

REDES: MSN FORENSE p.65

Até a polícia usa o brasileiro MSN Shadow para averiguar crimes via mensagens instantâneas

VEJA TAMBÉM NESTA EDIÇÃO:

- Acessibilidade total com desktops auditivos p.50
- O Minix 3 pode desbancar o Linux? p.56
- Scripts em Bash paralelos p.73
- Aprenda a montar redes Mesh p.60

Expediente editorial

Diretor Geral
Rafael Peregrino da Silva
rperegrino@linuxmagazine.com.br

Editor
Pablo Hess
phess@linuxmagazine.com.br

Revisora
Aileen Otomi Nakamura
anakamura@linuxmagazine.com.br

Editora de Arte
Paola Viveiros
pviveiros@linuxmagazine.com.br

Centros de Competência
Centro de Competência em Software:
Oliver Frommel: ofrommel@linuxnewmedia.de
Kristian Kißling: kkissling@linuxnewmedia.de
Peter Kreussel: pkreussel@linuxnewmedia.de
Marcel Hilzinger: hilzinger@linuxnewmedia.de
Centro de Competência em Redes e Segurança:
Achim Leitner: aleitner@linuxnewmedia.de
Jens-Christoph B.: jbrendel@linuxnewmedia.de
Hans-Georg Eßer: hgesser@linuxnewmedia.de
Thomas Leichtenstern: tleichtenstern@linuxnewmedia.de
Max Werner: mwerner@linuxnewmedia.de
Markus Feilner: mfeilner@linuxnewmedia.de
Nils Magnus: nmagnus@linuxnewmedia.de

Anúncios:
Rafael Peregrino da Silva (Brasil)
anuncios@linuxmagazine.com.br
Tel.: +55 (0)11 4082 1300
Fax: +55 (0)11 4082 1302

Petra Jaser (Alemanha, Áustria e Suíça)
anzeigen@linuxnewmedia.de

Penny Wilby (Reino Unido e Irlanda)
pwilby@linux-magazine.com

Amy Phalen (Estados Unidos)
aphalen@linuxmagazine.com

Hubert Wiest (Outros países)
hwiest@linuxnewmedia.de

Gerente de Circulação
Claudio Bazzoli
cbazzoli@linuxmagazine.com.br

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.es – Espanha
www.linux-magazine.pl – Polônia
www.linux-magazine.co.uk – Reino Unido
www.linux-magazin.ro – Romênia

Linux Magazine é publicada mensalmente por:

Linux New Media do Brasil Editora Ltda.
Av. Fagundes Filho, 134
Conj. 53 – Saúde
04304-000 – São Paulo – SP – Brasil
Tel.: +55 (0)11 4082 1300 – Fax: +55 (0)11 4082 1302

Impressão e Acabamento: Parma
Distribuída em todo o país pela Dinap S.A.,
Distribuidora Nacional de Publicações, São Paulo.

Atendimento Assinante

www.linuxnewmedia.com.br/atendimento
São Paulo: +55 (0)11 3512 9460
Rio de Janeiro: +55 (0)21 3512 0888
Belo Horizonte: +55 (0)31 3516 1280

ISSN 1806-9428 **Impresso no Brasil**

INSTITUTO VERIFICADOR DE CIRCULAÇÃO

Não é panaceia

Prezados leitores,

Motivos para adotar uma solução de voz sobre IP não faltam. Senão para cortar custos, o número de recursos oferecidos por um simples servidor Asterisk é surpreendente para quem se habituou às antigas centrais telefônicas convencionais. No entanto, o VoIP está longe de ser a panaceia que tanto se prega.

Conexões de Internet lentas ou, pior ainda, com alta latência, prejudicam enormemente o serviço de telefonia via Internet, mesmo no caso de operadoras que certamente não fazem *traffic shaping*. Para uma boa instalação de VoIP, há muito mais variáveis do que na telefonia convencional – certamente surpreendendo mais uma vez os profissionais mais antigos da área.

A segurança é mais um dos novos problemas que chegam com as implantações de voz sobre IP. Na realidade, como problema a segurança não é exatamente nova, mas as possibilidades de ataques abertas pelo uso da nova telefonia via Internet devem assustar os administradores que ainda não haviam considerado o tópico.

Além disso, a própria falta de uma legislação específica para os serviços de voz sobre IP no Brasil permite que operadoras e provedores de acesso se isentem de qualquer papel na manutenção de um ambiente seguro e confiável para a nova forma de comunicação.

Mesmo com essas questões, certamente o uso do VoIP tende a aumentar. E, com ele, as ameaças à segurança das empresas. Enquanto a legislação não favorece esse mercado, os clientes ficam responsáveis por seus próprios serviços e segurança.

Cuide-se e aja. ■

Pablo Hess
Editor

CAPA

COLUNAS

NOTÍCIAS

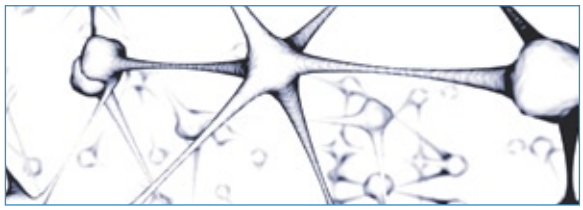
CORPORATE

TUTORIAL

ANÁLISE

REDES

SEGURANÇA



PROGRAMAÇÃO

SERVIÇOS

Emails para o editor

Permissão de Escrita

Se você tem dúvidas sobre o mundo Linux, críticas ou sugestões que possam ajudar a melhorar a nossa revista, escreva para o seguinte endereço: ***cartas@linuxmagazine.com.br****. Devido ao grande volume de correspondência, torna-se impossível responder a todas as dúvidas sobre aplicativos, configurações e problemas de hardware que chegam à Redação, mas garantimos que elas são lidas e analisadas. As mais interessantes são publicadas nesta seção.*

Samba e Windows Distributed Filesystem

Trabalho como Administrador de Sistemas Linux e atualmente estou enfrentando um problema, cuja dúvida é de raro material na Internet. Como resolver o problema da mensagem "Object is Remote" num volume CIFS (sobre o Distributed Filesystem, DFS) montado em qualquer sistema operacional Linux ou Unix?

Em vez de navegarmos no conteúdo montado do DFS dentro de um Linux, a mensagem perdura, dizendo "Object is Remote".

Como devemos montar sistemas de arquivos do tipo Windows DFS?

Por favor, ajudem-me.

Weber Morais

Prezado Weber, ao trabalhar com sistemas DFS no Linux, é importante usar sempre a versão mais recente do pacote Samba disponível para sua distribuição. Segundo informações disponíveis na Web, a montagem de volumes DFS com o Samba exige que se conheça o nó central do volume, ou seja, o servidor que agrega os demais volumes remotos. De posse do endereço desse servidor, basta montá-lo com o comando "mount" apontando para essa máquina.

Fica também a sugestão para que nossos leitores escrevam sugerindo soluções mais específicas para o problema.

Os melhores produtos VoIP

Linha profissional com os melhores preços

digium | Asterisk

Produtos IP para mercado corporativo
Tecnologia com segurança, alta qualidade de voz e funcionalidade customizáveis
Compatíveis com SIP e outros protocolos

- Vídeo Phone com alta qualidade de vídeo e compatível com IP-PABX Asterisk
- Gateways FXO até 8 portas possibilitam que um telefone IP / ATA faça ligações usando a rede pública
- Gateways FXS até 24 portas possibilitam que telefones analógicos integrem a uma rede VoIP sem a necessidade de um ATA
- PABX-IP com 4 ou 8 linhas analógicas, 30 ligações simultâneas e 100 ramais IP

Livetech Bahia | Air Live Powered by OvisLink Corp. | UBIQUITI NETWORKS | PROXIM WIRELESS NETWORKS | MikroTik | aperto networks | SkyPilot NETWORKS | RADWIN The Wireless Choice

digifort | EverFocus | KBC NETWORKS | NVT | Rainbow | PELCO by Schneider Electric | VERINT

SP (11)3035-3777 | RJ (21)4062-0078
PR (41) 4062-0045 | BA (71) 2626-2784

WDC Networks

www.wdcnet.com.br
comercial@wdcnet.com.br

COLUNA

Coluna do Klaus

Pergunte ao Klaus!

O professor Klaus responde as mais diversas dúvidas dos leitores.

Becape de disco inteiro

Às vezes eu uso o comando `dd` para fazer becape de um disco inteiro:

```
dd if=/dev/sda of=/mnt/disco.img
```

Essa técnica funciona bem, mas preciso retornar a imagem inteira para o disco antes de montar qualquer uma das suas partições. Se eu fizesse uma imagem de cada partição, bastaria depois montar o arquivo imagem como dispositivo de *loop*. Existe alguma forma de fazer isso com a imagem do disco inteiro?

Resposta

Sim, existe. Para montar sistemas de arquivos no Linux, não faz (quase) nenhuma diferença se este se encontra num disco ou num arquivo.

Exemplo 1: Saída do fdisk

```
01 You must set cylinders.
02 You can do this from the extra functions menu.
03
04 Disk becape.img: 0 MB, 0 bytes
05 255 heads, 63 sectors/track, 0 cylinders, total 0 sectors
06 Units = sectors of 1 * 512 = 512 bytes
07 Disk identifier: 0x00000000
08
09      Device Boot  Start       End    Blocks  Id  System
10 becape.img1 *        63   8401994   4200966   7  HPFS/NTFS
11 becape.img2     8401995  16595144   4096575  83  Linux
12 Partition 2 has different physical/logical endings:
13     phys=(1023, 254, 63) logical=(1032, 254, 63)
14 becape.img3    16595145  18699659  1052257+  82  Linux
➥ swap / Solaris
```

Para montar uma partição única como dispositivo de loop, basta o comando:

```
sudo mount -o loop,ro root_filesystem /media/teste
```

No caso de uma imagem de disco inteiro, a técnica é a mesma, mas é necessário saber onde começam e terminam as partições na imagem.

Por exemplo, caso a imagem se chame `becape.img` e tenha sido criada conforme o comando que você informou, é possível descobrir onde começam e terminam as partições com o seguinte comando:

```
fdisk -u -l becape.img
```

A saída é mostrada no **exemplo 1**.

Não se preocupe com diferenças de término físico e lógico agora; bastam os valores de início, que são fornecidos em setores de 512 bytes pela opção `fdisk -u -l becape.img`, e então pode-se calcular o início da partição 2 (de tipo *Linux*) dessa forma:

```
echo $((8401995 * 512))
4301821440
```

Agora que já sabemos que o sistema de arquivos começa no byte 4301821440 da imagem, podemos montá-lo com:

```
sudo mount -o loop,ro,offset=4301821440 becape.img /
➥media/test
```

Se você não quiser montar a partição em modo somente-leitura, basta retirar a opção `ro` do comando. ■

Klaus Knopper é o criador do Knoppix e co-fundador do evento *Linux Tag*. Atualmente trabalha como professor, programador e consultor.

INTEROP®

SÃO PAULO | 2 – 3 SETEMBRO, 2009

TRANSAMÉRICA EXPO CENTER

FAÇA PARTE DO EVENTO LÍDER GLOBAL EM BUSINESS TECHNOLOGY

A **INTEROP** é o único evento no Brasil, com foco em Business Technology, que abrange todas as áreas de TI nas pequenas, médias e grandes empresas e canais de distribuição.

A partir deste ano, a **INTEROP** conta com a parceria da IT Mídia em sua organização.

ENCONTRE EM UM SÓ LUGAR:

- **Mais de 5.000 visitantes altamente qualificados** em busca de soluções: diretores, CIO's, CTO's, CSO's, COO's.
- **Mais de 100 empresas e marcas expositoras** nacionais e internacionais
- **Mais de 6.000m²** de área de exposição
- Completo **programa de conteúdo** organizado pela IT Mídia e Techweb USA, que trarão os **principais temas internacionais, adaptados para o mercado brasileiro**.

www.interopsaopaulo.com.br

Organização

United Business Media

techweb
A Division of United Business Media LLC

it•mídia
mídias de negócios

Para mais informações:
11 4689.1935 • cfacc@cmpi.com.br

COLUNA

Coluna do Charly

Listas negras antispam

Na Universidade do Baixo Reno, futuros administradores se defendem do spam atraindo a Máfia do Viagra. Os resultados são listas negras populosas e conhecimento avançado de métodos para combater essa ameaça.

Um projeto na Universidade do Baixo Reno[1] em Krefeld, Alemanha, prepara alunos para a vida profissional, o trabalho em equipe e o cotidiano louco, parte do qual é a emergência do spam. O spam pode ser combatido pelo uso de vários métodos e um deles é a lista negra de spam (*spam blacklist*, ou SBL). Agora, os alunos da universidade estão fazendo a implementação e a manutenção de uma SBL.

Seguindo a ideia de "combate ao spam com spam", nós criamos deliberadamente contas de email IMAP e POP3 desprotegidas de spam. As contas atuaram como *honeypots* para capturar spam. Para atraí-lo, os alunos espalharam os endereços de email do honeypot o máximo que conseguiam. Para isso, ignoraram todas as regras relacionadas ao uso responsável de endereços de email e publicaram-nos em redes sociais; também postaram em grupos de discussão de testes e visitaram os cantos mais sombrios da Web.

Não levou muito tempo para alcançarmos resultados satisfatórios: as contas rapidamente se encheram com toneladas de spam. O trabalho dos alunos era criar um sistema para determinar a origem das mensagens recebidas o mais rápido possível (por meio da identificação do IP do servidor de envio) e adicionar o spam à lista negra por um período de tempo determinado.

O objetivo disso era permitir que o servidor de email comaprasse os endereços IP dos servidores de envio com a lista negra ao verificar suas contas regulares. Se o servidor de email percebesse que um servidor de envio estava recentemente envolvido na distribuição de spam, ele se recusaria a aceitar o email (**figura 1**).

O primeiro passo foi consultar as contas IMAP automaticamente em intervalos regulares. Em seguida, rotinas extraem do cabeçalho do email os detalhes do servidor que o entregou. Os trechos relevantes se localizam nas linhas `Received`:

```
Received: from bhixhv
(wsip-70-183-106-183.sd.sd.cox.net [70.183.106.183])
➥by islay.kuehnast.com (Postfix) with ESMTP id
➥B3B1F5D7A3; Tue, 21 Oct 2008 20:17:43 +0200 (CEST)
```

O endereço publicado, *islay.kuehnast.com*, aponta para um dos servidores do honeypot. Portanto, é bastante óbvio que qualquer máquina que envie emails para ele é uma distribuidora de spam – em muitos casos é uma máquina comprometida que está sendo usada como zumbi numa *botnet* [2]. O nome usado pela máquina, *bhixhv*, é forjado – o que ocorre com praticamente todos os spams.

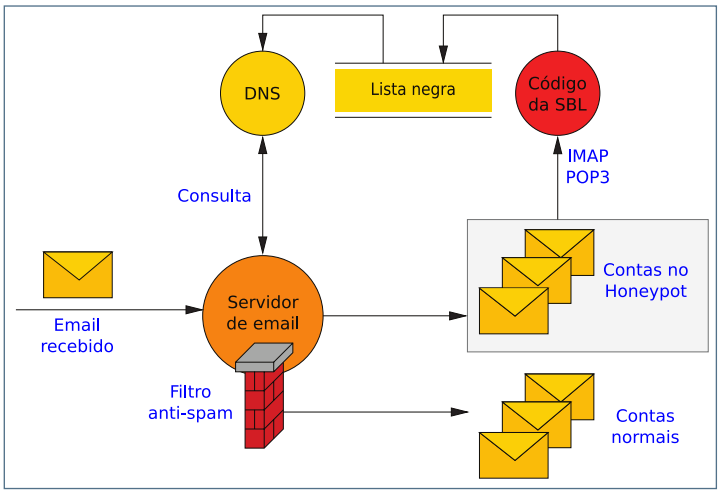


Figura 1 As mensagens enviadas por um servidor listado são filtradas.

Uma consulta ao DNS reverso desse IP, *wsip-70-...*, não é necessária; o IP não tem problemas. Esse IP é listado no log do *Postfix* entre colchetes, o que facilita sua extração por expressões regulares. A mesma linha contém a hora do servidor, relevante para estender automaticamente as entradas de agressores recorrentes ou para retirar automaticamente um IP da lista caso ele não envie spam dentro de um período definido.

Exemplo 1: Configuração do Policyd-weight

```
01 ## DNSBL settings
02 @dnsbl_score = (
03 #HOST,                BAD SCORE,   GOOD SCORE,  LOG NAME
04 'list.dsbl.org'       3.5,         0,           'DSBL_ORG',
05 'ix.dnsbl.manitu.net' 3.5,         0,           'IX_MANITU',
06 'sbl.hsnr.de',        3.5,         0,           'HSNR_DE',
07 );
```

Criação de zona DNS

De um ponto de vista técnico, a SBL é uma zona DNS combinada com uma consulta DNS para descobrir se um servidor está listado na SBL.

Agora, a configuração acrescenta à zona da SBL no DNS o endereço IP descoberto:

```
183.106.183.70.sbl.hsnr.de  IN A 127.0.0.10
                            IN TXT   "Spam from
➥this IP received: 2008-10-21-20.17h"
```

Como as consultas à zona da SBL envolvem uma consulta reversa, é preciso informar os octetos individuais em ordem inversa. O DNS resolverá o nome para o IP 127.0.0.10; o último octeto, *.10*, foi escolhido arbitrariamente, sendo que a principal característica é ser maior que *1*. Poderiam ser usados números diferentes para classificar os resultados – por exemplo, para avaliar qual honeypot forneceu aquela entrada.

O texto do registro TXT é armazenado nos logs do servidor de email quando ele se recusa a aceitar um email devido a uma consulta à SBL. No caso mais simples, diríamos ao servidor para cancelar o link de comunicação com o servidor de envio no caso de este se encontrar na SBL. A configuração no Postfix fica assim:

```
smtpd_recipient_restrictions = [outras_regras],
➥reject_rbl_client sbl.hsnr.de, permit
```

Independentemente da fé depositada na eficácia da SBL do tipo "faça você mesmo", ainda não é uma boa ideia descartar emails com base simplesmente num item de uma lista. Ferramentas como o *Policyd-weight* [3] do Postfix oferecem uma técnica mais ampla para aplicar múltiplos critérios de detecção de spam. Por exemplo, o Policyd-weight pode consultar múltiplas SBLs e realizar outras verificações nos cabeçalhos. O *daemon* gera um valor a partir dos resultados e compara-o a um limite configurável para decidir aceitar ou descartar o email.

O **exemplo 1** mostra uma configuração do Policyd-weight com três SBLs. Se o valor-limite for, por exemplo, *8.0*, um servidor precisa estar presente em todas as três listas para o Policyd-weight classificá-lo como spammer.

Igualdade

Os alunos podiam escolher livremente as armas para implementar sua SBL. A variedade de soluções propostas por eles simplesmente prova o velho ditado, "pergunte a dez cientistas da computação e obtenha 11 soluções". Por exemplo, as rotinas para extração dos dados relevantes a partir dos emails do honeypot incluía soluções tão diferentes quanto um script *Bash*, *PHP*, *.NET* e *C*.

O *Bind 9* venceu a competição dos servidores de nomes por sua configuração simples e suporte a múltiplas plataformas. Porém, alguns participantes queriam controle total sobre o código e decidiram programar um servidor de nomes em miniatura que oferecia os recursos necessários e nada mais.

O projeto de incinerador de lixo ainda está em andamento. Imagino o que os alunos vão criar quando chegarmos à etapa final, "monitoramento e relatórios". Mal posso esperar para ver. ■

Mais informações

[1] Universidade do Baixo Reno (em alemão): http://www.hs-niederrhein.de/fb03.html

[2] Charly Kühnast, "Gangue virtual": http://www.lnm.com.br/article/684

[3] Policyd-weight: http://www.policyd-weight.org

Charly Kühnast é administrador de sistemas Unix no datacenter Moers, perto do famoso rio Reno, na Alemanha. Lá ele cuida principalmente dos firewalls.

COLUNA

Coluna do Zack

Crônicas do Kernel

Mudanças na árvore linux-next e uma simplificação dos relógios do sistema.

Nova política na linux-next

Stephen Rothwell recentemente esclareceu as exigências para inclusão de código na árvore *linux-next*. Para começar, uma árvore só pode ser incluída na linux-next se seus *patches* forem postados na lista de emails apropriada e passarem pela revisão adequada. A árvore também precisa ter passado em testes de unidade e é preciso que seja desejo de seu mantenedor incluí-la no kernel oficial durante a próxima "janela" de inclusão (ou *merge window*, no jargão dos desenvolvedores).

Stephen também explicou quais eventos podem conspirar para ocasionar a exclusão temporária de uma árvore (por exemplo, até os problemas serem resolvidos). Isso inclui qualquer conflito com a árvore de Linus Torvalds que não tenha solução trivial. Qualquer árvore que não possa ser compilada com sucesso também será excluída. Isso está de acordo com a ideia de que as árvores na linux-next devem estar prontas para o uso imediato, já que visam à inclusão na árvore de Linus. Outra forma de ter sua árvore excluída é se ela conflitar de forma não trivial com outras que já estejam na linux-next. Nesse caso, o que se supõe é que se a sua árvore conflita com outra, o problema foi causado pela sua e portanto é nela que ele deve ser resolvido. Na prática, os mantenedores conflitantes podem se acertar para solucionar o conflito.

Tudo isso representa uma leve mudança na política de Stephen. Até recentemente, ele estava disposto a criar patches pessoalmente para permitir a compilação com sucesso, mas não mais. Qualquer árvore incapaz de ser compilada, ele disse, será excluída até ser consertada. Por outro lado, Stephen disse que está disposto a solucionar pequenos conflitos entre árvores e não pretende ser tão rígido com relação a exclusões por erros pequenos. Durante a janela de inclusão, ele espera ver quase tudo passar da linux-next para a árvore principal do kernel. Fora da janela, ele afirmou que deseja manter a árvore num estado que seja conveniente para Andrew Morton usar na criação de suas árvores *-mm*.

Relógios demais

Jonas Bonn tem pensado muito em relógios (*clocks*). Qualquer sistema possui vários relógios, incluindo o da CPU, o do barramento e os de vários dispositivos externos. As ferramentas disponíveis para lidar com eles, como *cpuidle* e *cpufreq*, tendem a oferecer recursos semelhantes, segundo Jonas, enquanto requerem processamento especial com base em cada sistema. Jonas propôs criar uma única API para permitir que o programador defina várias restrições, como o consumo de energia desejado do sistema, enquanto lida com a coordenação de todos os relógios do sistema por trás dos panos para que o desenvolvedor não precise pensar nisso.

Várias pessoas como Jon Smirl se manifestaram a favor dessa ideia geral, mas também indicaram que ela é semelhante a outros esforços já em andamento. Como Mark Brown lembrou, Dmitry Baryshkov vem trabalhando numa versão genérica da atual API *clk*, e Alan Jenkins mencionou o projeto OMAP originado em julho no Ottawa Linux Power Management Summit. Depois de examinar as alternativas, Jonas decidiu que a melhor atitude seria melhorar a API no código clk atual. Ele disse: "Implementei essa interface para o S3C2410 e ela oferece a funcionalidade que eu quero: que os relógios se tornem cidadãos de primeira classe e possam ser ajustados em conhecimento específico sobre os dispositivos que podem ou não estar utilizando-o". ■

A lista de discussão Linux-kernel é o núcleo das atividades de desenvolvimento do kernel. **Zack Brown** consegue se perder nesse oceano de mensagens e extrair significado! Sua newsletter Kernel Traffic esteve em atividade de 1999 a 2005.

azzu
.com.br

Realizar suas ligações agora ficou muito mais fácil... **e de graça*!**

Número Especial, Siga-me Temporário, Secretária Eletrônica, Voice Mail, Tele-Reunião e Multi-Azzu.

*Ligações gratuitas entre Azzu (ou Número Especial Contratado) e Ramais do Brasil Telecom.

COLUNA

Coluna do Kurt

O ataque das requisições cross-site

Às vezes, até mesmo ING, YouTube, New York Times e Google se enganam

Ataques do tipo *cross-site request forgery* (CSRF ou XSRF) estão se tornando rapidamente um sério problema de segurança desconhecido da maioria dos programadores e usuários. O CSRF é um ataque baseado na Web que deriva do (mas ainda permanece semelhante ao) tradicional ataque *cross-site scripting*, ou XSS. Num ataque XSS, o agressor fornece conteúdo malicioso numa aplicação web (por exemplo, criando uma URL mal formada ou embutindo código hostil numa caixa de resposta) que resulta em conteúdo hostil tal como JavaScript inserido em códigos até então seguros que serão servidos às vítimas. Ataques CSRF levam esse comportamento um passo adiante, inserindo conteúdo hostil que resulta numa ação por parte do navegador do usuário, como alterar a configuração de um filtro no webmail ou efetuar uma transferência bancária pelo site do banco.

Exemplo de ataque

Então você entra em sites de rede social para conversar com amigos. Infelizmente, o site em questão permite que os usuários insiram imagens em conversas via Web (por exemplo, avatares de um fórum). Em vez de usar uma URL como:

```
<img src="http://site-aleatório/imagem.jpg">
```

o agressor usa a URL:

```
<img src="http://site-social/changepassword?newpassw
ord=senha000">
```

Assim, quando o navegador web do usuário tenta carregar a imagem, ele se conecta ao site da rede social e executa um comando para alterar a senha.

Esse ataque também pode ser executado a partir de outros sites. Por exemplo, se um usuário permanece logado no site da rede social enquanto navega em outra aba e uma imagem de outro site aponta para a URL de mudança de senha, essa aba executa o comando e altera a senha do usuário, a menos que o site tenha proteções específicas contra CSRF.

Os ataques CSRF se popularizaram por três motivos simples. O primeiro é o crescimento de serviços baseados na Web tais como email, comércio online, Internet banking etc. Ataques CSRF podem resultar em envio de dinheiro para um agressor por sites de bancos ou de compra e venda de ações. Webmails permitem que o agressor altere ou peça cópias das senhas de vários serviços, como registros DNS e sites de comércio eletrônico. Os agressores podem monetizar esses ataques pelo direcionamento do acesso a contas bancárias, alteração da senha do usuário etc. Algo tão simples quanto alterar uma senha pode resultar no uso da conta, domínio ou serviço como refém. Por uma pequena taxa, o agressor altera a senha e devolve-a ao usuário.

O segundo motivo é a presença da navegação por abas. Quando os navegadores Web surgiram, navegar pela Internet era uma tarefa totalmente serial. Não me ocorreu durante algum tempo que eu algum dia precisaria de mais de uma sessão, porque o conteúdo não era tão significativo que me fizesse conservá-lo (a navegação era literalmente navegação). Entretanto, com o advento do webmail, agora tenho três sessões simplesmente logadas (mas paradas) para eu conseguir enviar emails rapidamente e ser notificado quando chegarem novas mensagens. Isso significa que um ataque CSRF tem muito mais chance de sucesso porque estou sempre logado em meu webmail (eu uso um navegador separado para o email, justamente para evitar esse problema).

O terceiro motivo é que a maioria das aplicações web não tem qualquer segurança. Elas são absolutamente terríveis na filtragem dos dados fornecidos pelo usuário, e com isso permitem que agressores injetem conteúdo malicioso (como *JavaScript*) por meio de diversas vulnerabilidades

a XSS. Embora eu raramente visite websites hostis, visito vários sites "confiáveis" que sei (por constatação) que não possuem segurança adequada e podem resultar em ataques XSS, culminando na viabilização de ataques CSRF.

Além disso, poucas aplicações web implementam proteções contra CSRF para evitar tais ataques.

Defesas para programadores

A forma de vencer o CSRF é conceitualmente simples, mas, dependendo da aplicação web, sua implementação pode variar entre fácil e quase impossível. Para vencer ataques CSRF, uma aplicação precisa simplesmente assegurar que todas as requisições sejam realizadas de forma adequada; em outras palavras, sua aplicação precisa manter seu estado para que o conteúdo da aba 'A' que está logada no seu webmail seja o único com permissão de enviar emails, e uma requisição feita pelo conteúdo da aba 'B', que está num website hostil, não resulte no envio ou leitura de emails. Para isso, sua aplicação web precisa manter informações de estado. Mas a Web foi projetada como um sistema sem estado desde o princípio, então qualquer acréscimo de estado requer truques técnicos, pois o navegador em si não tem como ajudar diretamente.

Para vincular o conteúdo de uma página à requisição feita de forma correta a partir de outra página (isto é, o usuário preenche um formulário e clica em *Enviar*), é preciso passar um *one-time token* no conteúdo, que o navegador então repassa com a requisição, permitindo que se confirme que a requisição veio do local correto.

Envio e recepção de tokens

Esse one-time token pode ser enviado e recebido de diversas maneiras:

- campos ocultos em formulários

```
<input type="hidden" name="token"
➥value="textoaleatório" />
```

  - A vantagem, nesse caso, é que muitas aplicações suportam a inclusão de campos de formulário e a lógica para processá-los. A desvantagem é que páginas web que não usam formulários mas ainda permitem interação não têm uma solução tão fácil assim;
- componentes de URL (seja dentro da URL ou como parâmetros)

```
http://exemplo.org/novasenha?nova=senha&token=texto
➥aleatório
```

  - Esta tem a vantagem de disponibilizar os dados para o servidor para que se possa, em teoria, usar um módulo do Apache para validar todas as requisições e simplesmente bloquear as inválidas, evitando assim que a aplicação sequer as enxergue. A desvantagem (ou potencialmente uma vantagem, dependendo do ponto de vista) é que os usuários não podem mais adicionar a página aos favoritos, já que o one-time token não será mais válido.
- cookies

```
PHP: setcookie("Cookie_do_token", $textoaleatório);
```

  - Os cookies precisam estar ativados para isso funcionar, e potencialmente podem ser roubados por um agressor inteligente (várias falhas de navegadores já permitiram o roubo de cookies ao longo dos anos). A vantagem dessa técnica, contudo, é ser invisível para o usuário e não exigir que o HTML seja exibido para o usuário ou que a URL a ser usada precise de modificações;
  - exigências do *back-end*: todos esses exemplos requerem alguma forma de back-end para armazenar os dados de sessão e criar tokens para elas, compará-los e permitir ou rejeitar requisições com base neles. Além disso, aplicações baseadas na Web podem precisar de modificações (por exemplo, se campos ocultos de formulário forem usados para passar os dados). A boa notícia é que cada vez mais aplicações web estão implementando essa proteção por padrão. Por exemplo, o popular framework *Joomla* agora possui a função `JRequest::checkToken()`.

Defesas para usuários

A boa notícia é que há um grande número de defesas contra ataques CSRF. Uma muito comum é o plugin *NoScript* do Firefox. Infelizmente, para o NoScript ser eficaz é necessário desativar o JavaScript por padrão e então ativá-lo seletivamente para os sites confiáveis. Isso leva a questões óbvias de usabilidade, pois muitos sites nem funcionam sem JavaScript. Além disso, essa técnica não impede que um agressor realize um ataque XSS em um site em que você confia.

Um navegador que incorporou essa estratégia é o *Google Chrome*. Cada aba do Chrome é um processo separado, e não uma *thread* com o mesmo contexto das outras abas. Portanto, as abas não podem interferir entre si, o que impede a maioria dos ataques CSRF. Para sofrer um ataque, seria preciso fazer login num serviço baseado na Web e em seguida usar essa mesma aba para visitar um site hostil. ■

Kurt Seifried é consultor de segurança da informação especializado em redes e Linux desde 1996. Ele frequentemente se pergunta como a tecnologia funciona em grande escala mas costuma falhar em pequena escala.

COLUNA

Coluna do Pablo

Kernel novo

Além do suporte ao novo sistema de arquivos Ext4, a versão mais recente do kernel trouxe novidades em várias outras áreas.

Reduzindo significativamente o tempo de desenvolvimento e o número de alterações com relação às versões anteriores, a natalina versão 2.6.28 do kernel Linux foi liberada por Linus Torvalds no dia 24 de dezembro. Nos 76 dias de desenvolvimento dessa versão (contra 88 dias das versões 2.6.27 e 2.6.26), mais de 11.000 arquivos foram alterados (contra 15.127 da versão anterior).

Uma característica marcante dessa versão é que boa parte das principais novidades já são prontamente utilizáveis pelos usuários.

Armazenamento

O sistema de arquivos Ext4, criado para substituir o Ext3 como padrão do Linux, finalmente deixou a fase de desenvolvimento e agora se chama *Ext4dev*. Embora ele já esteja pronto para uso em desktops, não é saudável empregar um sistema de arquivos tão recente para os dados mais importantes.

Quem possui um disco rígido com proteção contra choques finalmente já pode desfrutar dessa segurança no Linux, pois foi acrescentado o suporte a essa tecnologia.

Com relação ao processamento de I/O, o Linux ganhou também a possibilidade de controlar a associação entre processadores e essas operações.

Gráficos

O servidor gráfico *Xorg* já dispõe de recursos muito interessantes, mas a arquitetura gráfica do Linux ainda pode melhorar bastante, principalmente em vista dos avanços na área de hardware de vídeo na última década.

O primeiro passo para uma nova arquitetura gráfica acaba de ser dado com a inclusão do GEM (*Graphics Execution Manager*), um novo gerenciador de memória de GPUs. Ele servirá de base para outros componentes gráficos que prometem revolucionar o desktop Linux, como o *modesetting* no kernel, o DRI2 (descendente do DRI) e a UXA (descendente da EXA).

Tudo sem fio

A nova tecnologia *Ultra Wide Band* (UWB) está pronta para eliminar os cabos USB e firewire (IEEE 1394), de rede e outros. Ela fornece uma base para comunicação sem fio em alta velocidade (480 Mbps a dois metros de distância e 110 Mbps a dez metros). E o Linux acaba de ganhar drivers para suportar não apenas o UWB, como também *Wireless USB* e *WiMedia*, dois dos protocolos que dependem da nova tecnologia.

Além disso, o novo protocolo de rede *PhoNet*, desenvolvido pela Nokia para seus modems celulares, também já pode ser usado no sistema do pinguim.

Desempenho puro

Além da memória gráfica, a memória principal do sistema agora é gerenciada com mais competência, principalmente no caso de sistemas com muita memória e vítimas do lento *swap* em disco.

Quem pretende acelerar a inicialização do sistema agora também conta com o *Boot tracer*, uma estrutura que registra o tempo necessário para cada tarefa executada pelo kernel. Esse complemento ao competente *bootchart* deve ajudar todos os usuários na busca da inicialização rápida.

Drivers e outros

Como de costume, inúmeros novos drivers foram acrescentados ao Linux. Um dos mais célebres adiciona suporte ao *touchpad* Elantech presente nos netbooks Asus Eee PC, mas há outros em todas as áreas, como dispositivos de blocos, redes sem fio, USB, ACPI, captura de vídeo, rede etc. ■

Coluna do Augusto

Genealogia das distribuições

Distribuições Linux têm uma dinâmica bastante particular, mas ainda é possível demonstrá-la com uma árvore genealógica

Genealogia é um estudo que pode ser fascinante, a ponto de ser encarado como hobby por bastante gente, dedicando suas horas livres a desencavar laços entre famílias e as origens históricas das linhagens. Sua técnica básica é a exposição cronológica, usualmente tomando a forma de um diagrama, da filiação de um indivíduo ou da origem e ramificações de uma família.

Uma das ferramentas essenciais das análises das relações entre linhagens e dinastias ao longo do tempo é o cladograma, diagrama em árvore que apresenta a história evolutiva de um conjunto de famílias.

Em sentido figurado, a genealogia também pode significar o estudo da história do desenvolvimento de um gênero qualquer da atividade humana, apontando a procedência ou origem de cada um de seus ramos – por exemplo, o estudo das origens e descendentes das várias dezenas de distribuições Linux com alguma relevância existentes hoje.

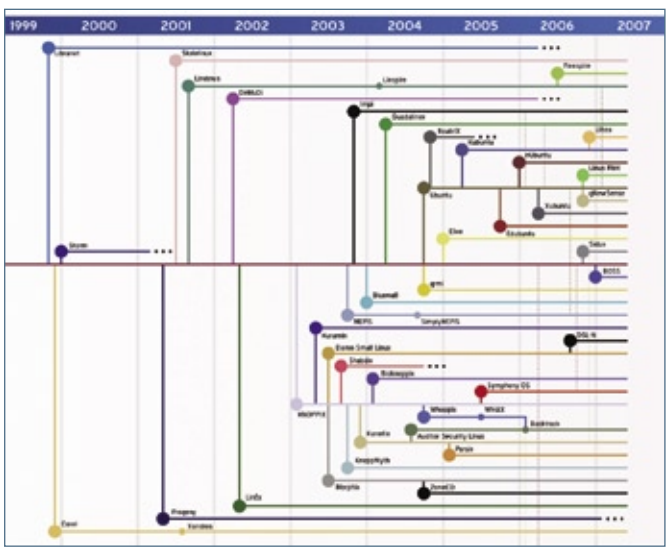


Figura 1 Não é difícil imaginar onde começa a linha que dá origem a todas essas distribuições.

Claro que as relações de ascendência e descendência, neste caso, podem não ser tão objetivas quanto a maternidade humana: uma nova distribuição de Linux pode obter componentes de diversas origens. Frequentemente, entretanto, elas surgem a partir de outras, pré-existentes, na forma de variações ou especializações, que depois acabam seguindo seu próprio caminho independente, como aconteceu com o Mandriva, a partir do Mandrake, que incorporou o Conectiva Linux – ambos, por sua vez, derivações do Red Hat Linux.

Boa parte das distribuições hoje populares pode traçar suas origens a partir de um pequeno número de distribuições iniciais e de poucas outras que surgiram de forma diferenciada ao longo do caminho – caso da criativa distribuição brasileira Gobolinux, por exemplo.

Acompanhar esta evolução é interessante. Você sabia que as primeiras distribuições Linux surgiram já em 1992 e que só uma delas (o SLS, há muito extinto) ainda tem descendentes em atividade? E sabe quais são as três principais origens das distribuições existentes?

A resposta pode ser encontrada no cladograma das distribuições Linux, um belo quadro (disponível sob uma licença livre em [1]) apresentando todo esse histórico. Eu recomendo a consulta, não apenas porque pode ser uma análise divertida, mas especialmente porque pode permitir entender melhor a dinâmica da evolução dos sistemas operacionais livres baseados no Linux. ■

Mais informações

[1] Cladograma das distribuições Linux: http://futurist.se/gldt/

Augusto César Campos é administrador de TI e, desde 1996 mantém o site BR-linux.org que cobre a cena do Software Livre no Brasil e no mundo.

Falhas de segurança graves no SSL e no SSH

Ao final do quarto dia da 25ª edição do Chaos Communication Congress (25c3) – ocorrido em Berlim, na Alemanha, de 27 a 30 de dezembro de 2008 – causaram sensação as apresentações de dois palestrantes a respeito de falhas de segurança em dois dos mais importantes protocolos utilizados na Internet quando a questão é segurança: SSH e SSL.

A palestra de Luciano Bello e Maximiliano Bertacchini discorreu especificamente sobre a falha de segurança no gerador de números pseudoaleatórios que se abateu sobre o projeto Debian em meados de 2008 (conforme publicamos no artigo "Insegurança – O desastre do SSL no Debian", na Linux Magazine 46). Os dois pesquisadores argentinos, membros do Laboratorio de Investigación y Desarrollo en Seguridad Informática – diretamente ligado ao Ministério da Defesa da Argentina –, analisaram a brecha de segurança mais a fundo e alertaram para o fato de que o problema não afeta somente servidores Debian, mas todos aqueles para os quais foram gerados certificados de acesso utilizando Debian. Além disso, o problema não afeta somente o SSH, mas também arquivos assinados via GPG ou certificados SSL gerados com a biblioteca *libssl*.

Para coroar o final do evento, entretanto, foi apresentada uma palestra cercada de uma aura de misticismo, sobre como seria possível – de modo bastante prático – se aproveitar de uma falha de segurança no mecanismo de geração de chaves MD5. Uma equipe internacional, encabeçada por Jacob Appelbaum, demonstrou como um certificado raiz completo que é aceito por qualquer navegador de Internet pode ser criado a partir dessa brecha de segurança.

O truque foi encontrar um provedor de certificados SSL listado por padrão nos navegadores de Internet que ainda usasse o algoritmo MD5 para geração de chaves – apesar das brechas de seguranças que foram reportadas para esse algoritmo em 2004 e 2007. A função de geração de chaves do algoritmo MD5 permite criar uma réplica de arquivo com a mesma identificação digital (*hash fingerprint*) do arquivo original. Essa réplica é conhecida por especialistas pelo nome de "colisão". Dispondo-se de potência computacional adequada, pode-se manipular uma quantidade limitada de bytes do arquivo de colisão. A equipe envolvida nesse projeto utilizou-se para essa finalidade de um cluster composto por 200 unidades da Sony Playstation 3, cada uma delas equipada com um processador do tipo Cell, que usaram o arquivo de colisão em seus cálculos por três dias. Alternativamente, o cálculo poderia ter sido realizado através do aluguel de um cluster como o EC2 da Amazon, pela bagatela de 2.000 dólares. ■

Linux Magazine agora oficialmente no UOL

O sucesso dos programas de código aberto não é mais novidade para ninguém. Tanto no mercado corporativo quanto nas residências de todo o país, softwares como Firefox, OpenOffice.org, Linux, Google Chrome, BrOffice e Thunderbird já estão presentes em boa parte dos PCs.

Confirmando a tendência ao crescimento das tecnologias abertas, a partir de hoje a Linux Magazine integra oficialmente o quadro de publicações de tecnologia do UOL, ao lado dos mais importantes periódicos desse segmento.

"A integração da Linux Magazine Online ao portal de tecnologia do UOL é uma consequência natural do aumento da importância do software livre e de código aberto no mercado nacional", afirmou Rafael Peregrino da Silva, diretor executivo da Linux New Media, editora que publica a Linux Magazine no Brasil. "A parceria com o UOL coroa também os esforços da Linux New Media do Brasil para trazer conteúdo sólido e de qualidade neste segmento ao mercado editorial brasileiro. O Brasil se destaca pela ampla adoção dessa tecnologia, com 73 por cento

OpenOffice.org vive, mas precisa de mais participantes

No final do ano passado, o colaborador do OpenOffice.org Michael Meeks escreveu em seu blog uma mensagem de preocupação com o projeto, pintando-a em termos não muito positivos. Ele citou diversas estatísticas para mostrar que em sua opinião o código tem pouca manutenção, o que teria feito com que muitos desenvolvedores deixassem o projeto em razão da frustração. Isso causou a reação de Thorsten Ziehm, *quality expert* da Sun Microsystems para o OpenOffice.org e sua variante comercial StarOffice. Em seu post "What was done in 2008" (O que foi feito em 2008), Ziehm seguiu o exemplo de Meeks e exibiu gráficos e estatísticas, porém, desta vez para refutar as afirmações de Meeks. Enquanto Meeks cita uma crescente saída de desenvolvedores do projeto, Ziehm mostra os "quase 900 projetos relacionados" que foram integrados. Enquanto Meeks está convencido da rendição da Sun, ZIehm cita uma opinião contrária do vice-presidente da empresa, Jim Parkinson, em seu blog, emitida em novembro último: "We are not about to walk away from OpenOffice.org" (Não estamos prestes a abandonar o OpenOffice.org).

Florian Effenberger, representante de marketing do projeto, apontou a atual polarização como prova da vitalidade do projeto: "Os pontos de vista diferentes provam apenas que o projeto continua ativo".

Amanda McPherson, vice-presidente da Linux Foundation, revelou que vê o OpenOffice.org como a grande alternativa à Microsoft e seus 28 milhões de downloads da última versão do pacote como prova indubitável de seu sucesso.

Todos pareceram concordar que o OpenOffice.org está vivo e continuará assim caso mais participantes se envolvam. Ziehm resume com "precisamos de mais pessoas em todas as áreas e projetos – experiência do usuário, QA, localização, desenvolvimento ou qualquer outro projeto do OOo – para realizar nossos sonhos". ■

de uso nas grandes empresas, número comparável ao de nações desenvolvidas na Europa, como Alemanha e Espanha", completou.

Com a parceria, a partir de hoje, todo o material publicado no site da Linux Magazine poderá ser disponibilizado também no portal UOL Tecnologia. Os visitantes do portal terão à disposição todas as notícias publicadas pela Linux Magazine Online, assim como matérias e artigos em formato digital das revistas impressas Linux Magazine e Easy Linux.

A Linux Magazine conta atualmente com 20 mil leitores mensais, oriundos de uma circulação paga média auditada de 5.500 exemplares por mês, que a posiciona entre as cinco revistas de TI mais vendidas no país. ■

Cisco entrando no mercado de servidores

A Cisco Systems, maior fabricante mundial de equipamentos de rede, anunciou recentemente que lançará uma nova linha de produtos na área de servidores. Com o anúncio, a empresa americana estreará no mercado de servidores já com grande penetração, em razão da forte presença que já tem nos departamentos de TI.

A oferta da Cisco será um servidor *blade x86* com forte foco em virtualização, o que sugere um confronto com os fabricantes já estabelecidos no setor, como IBM, HP e Dell. Além disso, somado às recentes aquisições da empresa – a Cisco adquiriu a especialista em mensagens instantâneas Jabber.com em setembro e US$ 150 milhões em ações da fabricante de softwares de virtualização VMware –, o novo lançamento parece indicar que a estratégia será duradoura e que já vinha sendo planejada há tempos.

Os maiores parceiros da Cisco, isto é, fornecedores de armazenamento, software e hardware para computadores, têm agora um concorrente poderoso que pretende deixar de ser o "encanador da Internet" para se tornar um fornecedor completo.

No entanto, segundo uma análise do New York Times, o movimento da Cisco não parece tão promissor para os negócios da empresa. Sua margem de lucro bruto de 65% no mercado de redes é muito superior aos 25% comuns no mercado de servidores. Ainda assim, como mostra a análise, trata-se de um mercado de US$ 50 bilhões.

A reação esperada dos novos concorrentes, segundo analistas, é o abandono das antigas parcerias com a fabricante em torno de seus equipamentos de rede. Resta a dúvida, porém, se IBM, HP e Dell vão preferir estabelecer parcerias com outras empresas (como a Juniper, por exemplo) ou passar a fabricar e promover seus próprios produtos de conectividade.

Para a Cisco, existe ainda a possibilidade de aquisição da própria EMC, importante fabricante de equipamentos de storage e proprietária da VMware. Com isso, a empresa afirmaria ainda mais sua nova estratégia de mercado ao ganhar independência de outros fornecedores.

Primeiros resultados

Como primeiras manifestações da incursão da empresa californiana no novo mercado, foram apresentados à imprensa dois servidores de armazenamento.

O equipamento mais potente (NSS3000) se parece mais com um PC de mesa, e pode ser equipado com no máximo quatro discos rígidos, atingindo uma capacidade de total de até 4 TB. Adicionalmente, o NSS3000 dispõe de suporte a RAID 10 e RAID 5. O servidor consome 19 W sem discos rígidos instalados e 56 W se equipado com quatro discos de 250 GB.

Os dois equipamentos dispõem de uma conexão Ethernet Gigabit. Ambos criptografam os dados armazenados utilizando o padrão AES (*Advanced Encryption Standard*) com chaves de 256 bits, e oferecem suporte para os protocolos SMB, CIFS, SFTP e NFSv3. O sistema de arquivos utilizado é o XFS, desenvolvido pela SGI. Nos dois modelos, o sistema operacional é o Linux, com um kernel da geração 2.6.

Os preços sugeridos nos Estados Unidos são de US$ 600 para o NSS2000 e US$ 1.100 para o NSS3000. Ambos os equipamentos são listados na linha de sistemas de armazenamento para pequenas e médias empresas. A principal novidade são os gabinetes, que não foram concebidos para ser montados em rack.

Durante a apresentação dos equipamentos também foram mostrados um servidor de segurança dedicado, equipado com uma aplicação antivírus, bem como um telefone IP especialmente desenvolvido para empresas de até 100 funcionários, que poderiam, assim, adquirir um ambiente operacional completo, totalmente entregue pela Cisco.

Entretanto, no que tange a assumir o uso de Linux e Software Livre e de código aberto em seus produtos, a empresa ainda adota uma posição no mínimo tímida: quando a redação da Linux Magazine solicitou informações a respeito do sistema operacional utilizado no servidor de segurança, a resposta foi: "Não é nem o Windows nem o IOS da própria Cisco", declarou o porta-voz da empresa, sem especificar que o equipamento é equipado com Linux. Segundo a empresa, essa postura seria por motivos de segurança. Da mesma forma, não há nenhuma informação sobre qual distribuição Linux equipa os dois novos dispositivos de armazenamento apresentados pela empresa. ■

Sun adquire empresa de cloud computing

A Sun Microsystems adquiriu a Q-layer, empresa belga especializada em computação em nuvem. Segundo o anúncio oficial, a aquisição teria por objetivo a melhoria do portifólio de soluções em *cloud computing* da Sun. A Q-layer, fundada em 2005, é fornecedora de duas soluções nesse segmento, com as quais serviços como administração de servidores e de memória, largura de banda e aplicações especiais podem ser administrados facilmente, de tal forma que o usuário pode configurar seu ambiente operacional de acordo com as suas necessidades.

As soluções da Q-layer vão se tornar parte da unidade de negócios de cloud computing da Sun, criada em agosto de 2008 a partir da unidade de *utility computing*. Informações sobre os valores desembolsados na aquisição, bem como um cronograma para integração das soluções da Q-layer ao portfólio da Sun, não foram divulgadas. ■

Linux para parlamentares europeus

A política europeia de combate às práticas de mercado deletérias da Microsoft está mostrando mais uma vez sua força. Começando pela França, os 1.100 PCs dos parlamentares e seus assessores tiveram o Microsoft Windows substituído pelo Ubuntu há um ano e meio, na esperança de economizar 500 mil euros ao longo dos cinco anos seguintes – em virtude não apenas do menor preço do Ubuntu, mas também dos menores custos de manutenção da solução livre.

Segundo artigo no blog do New York Times, os legisladores franceses apostam na capacidade do Linux e do Software Livre para alavancar a produção de empregos locais, justamente pela liberdade para a oferta de produtos e serviços de origens diversas, em oposição ao atrelamento à empresa norte-americana.

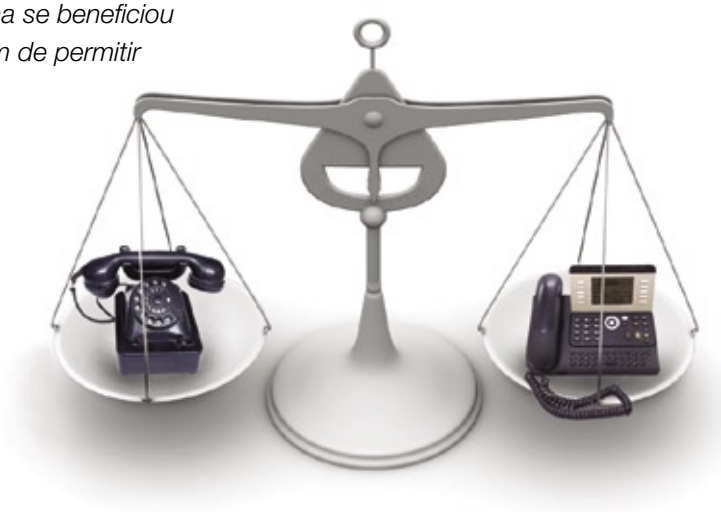
De acordo com Rudy Salles, vice-presidente da assembleia nacional francesa, uma pesquisa feita após 18 meses com os deputados constatou que 80% aprovaram o novo software, enquanto 14% preferem retornar ao Windows. ■

VoIP com Asterisk no Tribunal de Justiça de Santa Catarina

Telefonia justa

O Tribunal de Justiça do estado de Santa Catarina se beneficiou intensamente do amplo uso de voz sobre IP. Além de permitir custo zero nas ligações entre as várias unidades espalhadas pelo estado, a administração das centrais telefônicas ficou mais fácil e prática.

por Pablo Hess



O Tribunal de Justiça do estado de Santa Catarina [1] era uma das muitas instituições brasileiras que sofriam com serviços de telefonia insuficientes para sua operação. Não apenas o serviço de telefonia limitava as opções de comunicação como seu custo superava o desejável para uma solução de qualidade satisfatória.

Em 2006, o quadro começou a mudar. O TJ decidiu adequar a implantação, o uso e a gestão do VoIP no órgão para obter os resultados esperados de disponibilidade, qualidade de serviço, recursos e custo. Ao fim do segundo ano do processo de implantação, a **Linux Magazine** foi convidada a conhecer os ganhos obtidos pela instituição com a adoção progressiva da solução baseada no popular, poderoso e aberto PBX IP Asterisk [2].

Entenda como e por que o TJ/SC optou pela telefonia IP com este que já figura entre os softwares de código aberto mais utilizados pelas empresas mundo afora.

Antes

Até 2006, a gestão da telefonia era feita pela Diretoria de Infraestrutura do TJ, que não possuía o conhecimento técnico necessário. Com isso, a instituição se mantinha dependente das empresas de manutenção de telefonia e também das operadoras.

A solução de comunicação telefônica consistia em uma central com 600 ramais e uma placa E1, que se comunicava com um roteador com 30 canais de voz para servir 15 comarcas no estado via VoIP. O quadro era marcado por necessidade constante de ampliação. Além disso, a estrutura apresentava alto custo, fosse ela utilizada ou não.

Começam as mudanças

A primeira parte das alterações muito bem vindas à telefonia do TJ veio com a transferência da responsabilidade pelo serviço para a Diretoria de Informática. A empresa escolhida para a implantação da nova solução foi a também catarinense Khomp [3], fornecedora de hardware e prestadora de serviços em VoIP.

Tabela 1: Custos de ligação

Tipo de ligação	Custo anterior (R$)	Custo atual (R$)
Entre unidades (DDD)	0,37 a 0,41	0,00 (via Internet)
De unidade para outro telefone (DDD)	0,37 a 0,41	0,07 (ligação local)
Para celular no DDD 48	1,00 (via operadora fixa)	0,03 (via operadora móvel)

Os 200 ramais adicionais necessários à operação adequada do serviço de telefonia tiveram um grande peso na decisão da solução a ser adotada. Da mesma forma, era necessário haver autonomia perante a central telefônica, com menor custo de manutenção.

Os R$ 28 mil que seriam necessários para aquisição de mais 64 ramais mostraram-se gigantescos quando comparados ao custo aproximado de R$ 11 mil com a solução Asterisk para o mesmo número de ramais. Além disso, a nova solução oferecia um plano de discagem mais maleável, uma poderosa URA, bilhetagem com o software *A2Billing* e rotas de contingência para melhor disponibilidade do serviço.

Para os usuários, uma importante vantagem era a portabilidade do ramal, já que basta conectar-se a uma VPN para que qualquer funcionário tenha acesso à central do TJ/SC até a partir de outros estados.

Para a equipe técnica, a possibilidade de realizar manutenção e gerenciamento de todas as centrais de forma remota teve destaque. Usando VoIP com Asterisk em toda a estrutura do TJ, é mais fácil verificar onde se localizam eventuais problemas (na operadora ou na central, por exemplo). Ao expandir a solução para as demais comarcas do estado, essa vantagem será ainda mais explorada, gerenciando-se todo o sistema do estado a partir da sede do órgão, em Florianópolis.

Desafios

No início da implantação, a placa usada para comunicação com a central telefônica legada apresentou sérios problemas de eco nas ligações sob tráfego telefônico intenso. O fornecedor informava ser a operadora telefônica a responsável pelo problema. Após testes com outras placas compatíveis com a tecnologia R2 digital, foi encontrada uma que solucionava o problema.

Solução

A solução adotada oferece menor custo de manutenção do que a central telefônica legada. O custo total da implantação da central VoIP com Asterisk foi de R$ 172 mil, incluindo 200 aparelhos telefônicos IP, pois uma das exigências era a presença de um identificador de chamada nos aparelhos.

A central antiga, no entanto, não foi descartada. O plano inclui a migração gradativa da telefonia do TJ para o Asterisk ao longo de cinco anos, e no momento a central legada é perfeitamente adequada às necessidades atuais. Ela comporta 200 ramais que se somam aos 600 da central Asterisk, servidos por quatro links E1 ligados à rede pública e quatro outros links E1 comutados com a central legada. O gerenciamento, no entanto, é todo realizado pela nova central Asterisk.

Originalmente, o TJ/SC planejava trocar em seis anos todas as centrais ou inserir um gateway à frente das centrais de todo o estado, mas os resultados até o momento permitem reduzir essa expectativa para cinco anos.

Cada central Asterisk com até 70 ramais leva apenas uma semana para ser implantado sem qualquer trauma para o usuário final ou interrupção do serviço. Para centrais maiores, com 200 ramais, esse tempo continua baixo: duas semanas.

As ligações para celulares constituem um dos pontos marcantes da migração. A economia por minuto de conversação foi de 97% após a operação de duas placas GSM e um contrato com a operadora. Outras sete placas já foram adquiridas.

Panorama atual

No momento, há nove comarcas com um gateway Asterisk operando em conjunto com as antigas centrais. Cada gateway é equipado com 2 entradas E1 e atende 100 a 200 ramais, nesses casos. Existem também 13 comarcas sem qualquer central legada, funcionando com base em soluções Asterisk com uma entrada E1 para atender entre 30 e 80 ramais. Há ainda nove unidades dotadas de ramais remotos operando na sede principal, servidos por fibra óptica. Das 31 unidades, todas se comunicam via VoIP e fazem ligações locais por meio de qualquer uma das "pontas".

Na sede, a estrutura é mais poderosa: duas máquinas com quatro entradas E1 cada trabalhando em regime de redundância para garantir a alta disponibilidade. Além delas, há outras duas com a mesma configuração, também em redundância, para interligar os ramais legados. O gateway possui duas placas E1 ligadas ao roteador da operadora e uma terceira placa E1 conectada a uma central. A telefonia celular é atendida por uma máquina com duas placas GSM, e uma última se encarrega da funcionalidade de fax.

A **tabela 1** mostra a diferença dos custos gerais após a migração para o Asterisk.

Futuro

A solução atual de telefonia serve 850 ramais, com previsão para alcançar 1.100 até o final da migração. Além disso, as unidades nas regiões de prefixos 47 e 49 devem receber funcionalidade GSM ainda no primeiro semestre de 2009. ■

Mais informações

[1] Tribunal de Justiça de Santa Catarina: http://www.tj.sc.gov.br

[2] Asterisk: http://www.asterisk.org

[3] Khomp: http://www.khomp.com.br

A celebração da cultura digital

Campus Party Brasil 2009

A segunda Campus Party Brasil atraiu ainda mais participantes em relação à edição anterior, e cada vez mais se firma como um ambiente de construção de novas ideias e conceitos.

por Pablo Hess

Hoje em dia, falar em cultura digital de fato não é novidade. Todas as pessoas frequentam a Internet, seja para entrar no banco ou simplesmente para conferir os emails recebidos. Porém, há muitos que superam de longe esse uso da rede mundial de computadores. Na realidade, são eles quem propõem que a Internet não é uma rede de computadores, mas de pessoas – algo facilmente compreensível hoje, mas não tão claro assim há, digamos, dez anos.

São essas as pessoas que celebram a Campus Party Brasil: aquelas que consomem conteúdo online da mesma forma que o produzem, os *prossumidores* (produtores + consumidores), nas palavras de Don Tapscott e Anthony D. Williams, autores de *Wikinomics*[1]. São pessoas que de fato têm a Internet como habitat, ferramenta multifuncional e meio de comunicação.

Figura 1 O espaço Expo e Lazer contou com centenas de computadores para acesso pelos campuseiros.

História

Realizada pela primeira vez em 1997 na Espanha, a Campus Party descende de uma simples *LAN party*. Os membros da Asociación Juvenil EnRED realizavam, nessa época, frequentes LAN parties para jogar em rede usando as novas tecnologias que se estabeleciam nessa área. Ao decidirem abrir os eventos ao público, criou-se a Campus Party, à qual compareceram cada vez mais integrantes desde então.

Mudando de local a cada ano, a Campus Party foi realizada em várias cidades, como Málaga, Valencia e Palma de Mallorca. Em 2006, o evento espanhol já contava com aproximadamente 5,5 mil participantes e englobava áreas como astronomia, robótica, programação, jogos, *modding*, software livre e cinema.

Em 2008, o Brasil foi o segundo país a abrigar a Campus Party, seguido da Colômbia poucos meses depois. A primeira Campus Party Brasil, realizada no Parque do Ibirapuera, em São Paulo, SP, já começou com grande sucesso. Além da presença de 3,3 mil inscritos de 18 países, o evento contou com forte patrocínio de grandes empresas brasileiras e multinacionais.

Brasil, 2009

Na edição de 2009, realizada entre os dias 19 e 25 de janeiro no Centro de Exposições Imigrantes (**figura 1**), novamente em São Paulo, SP, foi marcante a diversidade do público participante.

Divididas nas áreas de desenvolvimento, design, fotografia, games, modding, música, robótica, simulação, software livre e vídeo, as 468 atividades do evento de fato abrangeram uma multiplicidade de conteúdos que dificilmente não impressiona. Os 6.655 campuseiros, como são chamados os participantes, utilizaram a banda de rede de 10 Gb para fornecer mais dados à Internet (62,7% do tráfego) do que para consumi-los (37,3%).

Variedade

Na área do software livre, uma das mais movimentadas do evento, foi intensa a mobilização dos participantes. Em um debate a respeito da legislação brasileira de controle da Internet, ficou marcada a desaprovação dos campuseiros com relação à proposta de lei do senador Eduardo Azeredo. Os participantes do debate, o desembargador Fernando Botelho,

o assessor de Azeredo, José Portugal, o sociólogo e professor universitário Sergio Amadeu e o também professor universitário Ronaldo Lemos, discutiram (**figura 2**) na presença de uma inflamada plateia (**figura 3**) que erguia laptops como cartazes para manifestar sua opinião.

Os defensores da chamada "lei Azeredo", Fernando Botelho e José Portugal, propuseram a idéia de que é necessária uma legislação adequada para tratar dos crimes na rede mundial, fornecendo as bases para que seja possível combater tais crimes, já que, de acordo com o código penal, "não há crime sem lei anterior que o defina".

Os protestos de Sergio Amadeu e Ronaldo Lemos, no entanto, apontaram que a lei não resolve os problemas que se propõe a solucionar. Lemos alegou que novos empreendimentos na Internet podem ser inibidos por temores de se cometer algum crime em consequência de má redação da lei. Amadeu, em específico, frisou as graves falhas de redação do projeto de lei que abrem brechas para se considerar ilegal, por exemplo, até mesmo a criação de uma rede *mesh* (veja o artigo à página 60 desta edição).

Figura 3 ...para uma plateia inflamada e favorável à liberdade.

Desafio multimídia

Jon 'maddog' Hall, presidente da Linux International e importante defensor e difusor dos princípios do Software Livre em todo o mundo, esteve presente a mais uma edição da Campus Party Brasil. Desta vez, maddog lançou o "Desafio Multimídia do Maddog", que contou com 16 inscrições. O objetivo: criar uma peça multimídia (em áudio, vídeo, música etc.) com o tema "Por que eu gosto de Software Livre". Naturalmente, os softwares usados para a tarefa deveriam obrigatoriamente ser livres.

A criação do vencedor Guilherme Guerra de Almeida retratou em uma animação bem humorada o episódio da professora americana que confiscou os CDs de distribuições Linux que seus alunos estavam trocando, alegando tratar-se de pirataria. A história ocorreu no segundo semestre de 2008 e foi difundida com extrema rapidez, integrando imediatamente a antologia de contos do Software Livre.

Maddog, na apresentação pública dos trabalhos, recomendou também que todos participassem do concurso "We're Linux" da Linux Foundation [2], que premiará com viagens e outros itens o criador do melhor vídeo de promoção do Linux no estilo da série "I'm a Mac", da Apple. O presidente da Linux International afirmou que iniciativas como essa podem facilitar imensamente a conquista de novos usuários pelo Software Livre.

Figura 2 Os participantes do debate sobre a liberdade na Internet expuseram suas idéias...

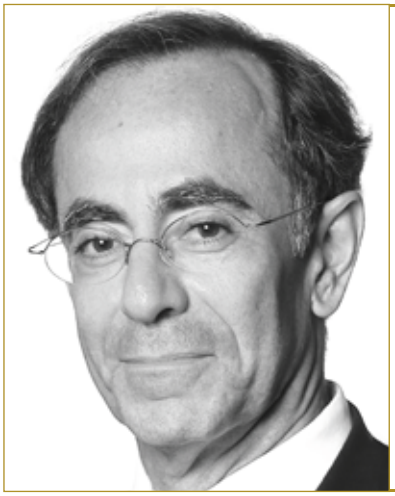
Ano que vem

Uma das felicidades ao fim da Campus Party é saber que no próximo ano teremos mais uma vez a realização de um dos maiores eventos de cultura digital do planeta. Para os que já se acostumaram a usar o evento também para fazer reuniões de grupos de usuários ou simplesmente reencontrar velhos amigos, essa certeza traz ao mesmo tempo paciência e ansiedade. ■

Mais informações

[1] on Tapscott e Anthony D. Williams, "Wikinomics": http://wikinomics.com

[2] Concurso "We're Linux": http://video.linuxfoundation.org/contest/we-are-linux



Coluna do Taurion

O movimento Open Source

Com a crise econômica que chega, o Open Source vai ganhar impulso. É hora de pensar no futuro do Open Source.

O movimento Open Source vai se acelerar com a crise econômica. Orçamentos mais apertados e ênfase na redução de custos tornam mais atraentes as alternativas de baixo custo.

Open Source não é mais novidade e vemos soluções de código aberto em todos os lugares. Na Internet, nos roteadores, nos data centers corporativos, em navegadores GPS etc. Estão na base da maioria das iniciativas Web 2.0 e redes sociais e nas linguagens de programação dinâmicas (PHP, Perl etc). Seu impacto na indústria de software, nas empresas e na carreira profissional não deve e nem pode ser ignorado.

E o que podemos esperar para os próximos anos? Embora futurologia seja uma área de alto risco, podemos certamente fazer algumas afirmações.

Open Source vai se tornar *mainstream* nas organizações. Será um padrão na prática em alguns segmentos como infraestrutura, ferramentas de desenvolvimento, computação de alto desempenho e na maioria dos softwares embarcados.

Os ecossistemas de software serão construídos em cima da sinergia entre modelos de negócio baseados em Open Source e proprietários. Praticamente todos os produtos de software proprietários vão incorporar, em maior ou menor grau, código aberto.

Parcela significativa das atividades profissionais de TI estarão diretamente relacionados ao Open Source. Não estou falando em ser simples usuário de Linux ou OpenOffice.org, mas em conhecer modelos de desenvolvimento colaborativos, criar e manter comunidades. Contribuir com a comunidade e obter certificações que comprovem conhecimentos em Open Source serão consideradas um "plus" pelos empregadores e profissionais.

Modelos de inovação abertos serão cada vez mais comuns e adotados pela maioria das empresas. Open Source é um exemplo prático e bem sucedido de Open Innovation. As empresas atuarão de forma mais pró-ativa com as comunidades, não apenas usando software Open Source, mas colaborando com código e capital intelectual. Desenvolver código aberto será visto como um investimento em P&D.

Open Source será visto como um campo específico da educação em ciência da computação e novas disciplinas serão adotadas pela academia. O diferencial do Open Source é o processo de desenvolvimento colaborativo de software, e não o uso dos sistemas Open Source, que é basicamente colaboração e inteligência coletiva. Precisamos estudar e entender os motivadores e inibidores para a criação de processos colaborativos, e para isso é necessário entender o que é uma cultura de participação (cultura = arquitetura + processos + governança).

Um ponto importante neste contexto é a criação de comunidades. É fundamental para o sucesso de qualquer projeto Open Source que uma comunidade ativa e atuante seja mobilizada. A comunidade é o coração dos projetos Open Source. O processo de desenvolvimento colaborativo do software Open Source também merece um estudo mais aprofundado. Apresenta características diferentes dos modelos de desenvolvimento comumente adotados nos softwares proprietários. Portanto, essa nova disciplina poderia ser algo como "Desenvolvimento em Open Source: práticas de engenharia de software". Outra disciplina essencial: modelos de negócio. Sem um ecossistema saudável, os projetos Open Source não serão sustentáveis. Deve existir receita em algum lugar da cadeia de valor, e a forma de obtê-la será o principal tópico dessa disciplina. ■

Cezar Taurion (ctaurion@br.ibm.com) é gerente de novas tecnologias aplicadas da IBM Brasil e editor do primeiro blog da América Latina do Portal de Tecnologia da IBM developerWorks. Seu blog está disponível em http://www-03.ibm.com/developerworks/blogs/page/ctaurion.

QUER VENDER PELA INTERNET E NÃO SABE POR ONDE COMEÇAR?

a partir de R$ 49,00 por mês

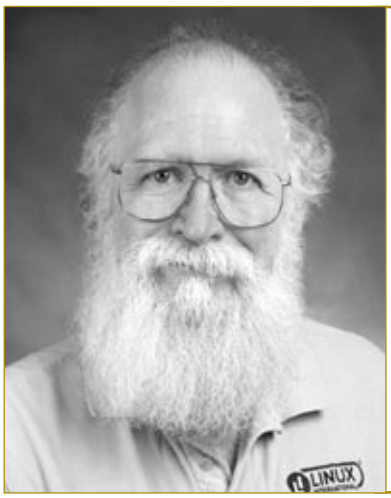
FALE COM O UOL.

0800 723 6000
www.uol.com.br/host

LOJA VIRTUAL UOL HOST

Solução fácil e descomplicada para empresas de qualquer tamanho colocarem seus produtos à venda na internet.

UOL HOST
QUALIDADE EM SERVIÇOS WEB



Coluna do maddog

A arte de vender software livre

A comunidade do Código Aberto não possui uma frota de advogados e consultores de marketing. Na hora de promover o Software Livre, talvez seja preciso fazer tudo.

Ontem à noite conversei com um difusor do Software Livre que abriu uma empresa. Perguntei o que sua empresa fazia, e ele respondeu "Software Livre". Mas isso me incomoda, porque sempre encontro pessoas que me dizem que gostariam de ganhar dinheiro "vendendo Software Livre". Às vezes essas pessoas de fato já pegaram empréstimos (ou usaram seu próprio dinheiro), abriram uma empresa, contrataram programadores, criaram um produto, publicaram o código-fonte e no final não chegaram a seu objetivo, pois não conseguiram gerar receita suficiente do software para obter lucro.

A melhor forma de contornar isso é não ter uma empresa que crie softwares livres, mas usar esses softwares para criar e fornecer as bases para uma empresa. Quando meu amigo respondeu que sua empresa fazia "software livre", pedi para ele definir o que seus clientes realmente poderiam comprar dele e como se beneficiariam do software. Ele não tinha uma resposta no momento. Ao abrir qualquer empresa é necessário estabelecer uma visão, uma missão e um conceito do que será vendido aos clientes. Software Livre não é suficiente.

Para ilustrar uma relação produtiva entre negócios e software livre, frequentemente uso o exemplo da indústria de turbinas a vapor. Há cinco centros de teste de turbinas a vapor no mundo para ajudar engenheiros a avaliar seus projetos de turbinas. Quatro desses centros usam softwares proprietários que costumam exigir um mês para uma alteração simples. O quinto teste usa softwares livres (MySQL, Linux, GNUplot, Tcl/TK e Python) para criar basicamente a mesma funcionalidade baseada na Web, mas o fornecedor do software frequentemente consegue realizar pequenas alterações de um dia para o outro. Todos os centros de teste oferecem serviços para engenheiros que projetam turbinas, mas o que usa Software Livre também oferece serviços de personalização rápida. Acho óbvio que o centro que oferece esse serviço a mais possui uma vantagem competitiva.

Aproveitando o tópico de abertura de empresas, na mesma viagem eu conversei com jovens programadores que queriam criar "software livre" e colocá-lo no "domínio público". Descobri que estavam um pouco confusos sobre o que é software no "domínio público" e o que a GPL oferece como licença. Quando eles diziam "domínio público", na verdade estavam pensando em código sob a GPL. E estavam ainda mais confusos quanto às questões de direitos autorais, licenças, marcas registradas e outros aspectos legais em torno da produção e publicação de software.

Em vez de definir todos esses conceitos aqui (há definições em vários locais), quero chamar a atenção dos programadores que estão entrando no negócio do Software Livre para que investiguem e realmente compreendam as questões de direitos autorais, as várias licenças livres e de código aberto e suas respectivas implicações. Esses aspectos podem ter um efeito dramático sobre a receita da sua empresa.

Por último, como dica para todos que estiverem pensando em abrir sua própria empresa de consultoria de Software Livre: em vez de começar sozinho, tente formar uma cooperativa com amigos do universo do Software Livre. As empresas frequentemente evitam trabalhar com autônomos. Elas questionam o que acontecerá se essa pessoa perder o interesse pela programação ou mudar de carreira. Uma cooperativa oferece uma presença permanente e asseguradora. Se o principal programador deixar a cooperativa, a organização pode recrutar um substituto. ■

Jon 'maddog' Hall é presidente da Linux International, instituição internacional dedicada a promover o Linux e o Software Livre e de Código Aberto. Maddog viaja o mundo ministrando palestras e debatendo com decisores sobre o uso do Software Livre em âmbito tanto corporativo quanto comunitário.

Como você se sente quando mais PRECISA do SUPORTE do fabricante do seu PABX ?

Só a Linha VOX - PABX IP da digivox, equipada com placas Sangoma, pode lhe proporcionar

A VERDADEIRA LIBERDADE EM TELEFONIA

Linha VOX - PABX IP

VOX mini + Placa A101 =

VOX MINI

Capacidade: VOX mini
Até 01 E1 (ISDN/R2)
50 ramais IP e/ou
12 ramais analógicos

VOX MAX

Capacidade: VOX max
Até 04 E1s (ISDN/R2)
200 ramais IP e/ou
96 ramais analógicos

Principais Vantagens da Linha VOX IP

- Interface amigável para gerenciamanto do PABX
- Instalação plug-n-play do Asterisk™ - rápida e fácil
- Integração com sistemas gerenciais da empresa
- Sistema de bilhetagem e controle de ligações - VOX audit
- Módulos adicionais de URA, gravação, fax e telemarketing
- Crescimento modular de troncos SIP/IAX e ramais IP
- Fácil integração com operadoras VoIP (Vono, Net fone, Hip, GT, etc)

Placas Sangoma

Placa A101 - 01 E1

A melhor qualidade em placas E1 para Asterisk™

- Produtos montados no Brasil
- Placas homologadas pela Anatel
- Suporte técnico no Brasil pela Digivox
- Entrega imediata e extensão de garantia nacional
- Descontos para revendas e integradoras de TI

Placa A104D - 04 E1s
Com cancelamento de eco

Diferenciais das Placas Sangoma

- Compatível com todos os links E1 - Digital
- Suporte a R2, ISDN, SS7, entre outros
- Funciona com Asterisk™, Yate™ e FreeSwitch™
- Integração com a maioria das centrais digitais E1
- Configuração automática das placas no Asterisk™
- SNMP para monitoramento de placas e portas
- Integração de dados e voz através da mesma E1
- Conexão de dados utilizando protocolos Frame Relay, PPP, HDLC e X.25
- Compatível com Linux, Windows™ e FreeBSD
- Hardware modular: PCI, PCI Express e PCI x

Placa A108D - 08 E1s
Com cancelamento de eco

digivox
www.digivox.com.br

empowered by SANGOMA

Compre agora na loja www.dvox.com.br e pague com: Cartão BNDES · VISA · MasterCard · Em até 36 meses Aymoré Financiamentos

Asterisk™ is a Trademark of Digium™ Inc.

Limite de 1 unidade por cliente. Máximo de uma unidade por CNPJ até 20/10/08 - US$ 1.165 x cotação do dólar no dia

Qualidade de serviço e legislação de VoIP no Brasil

O VoIP que não temos

Há inúmeros fatores que podem afetar a qualidade das comunicações por VoIP. No Brasil, a legislação é pouco exigente e o consumidor sai prejudicado.

por André Cabral Dutra

Os sistemas de comunicação baseados em VoIP têm se mostrado uma atraente solução na redução de custos de indivíduos e instituições no que diz respeito à economia proporcionada a partir de chamadas telefônicas locais e internacionais. A cada ano novos prestadores de serviço (operadoras e empresas especializadas em comunicação de dados) são atraídos por este promissor modelo de negócio. Observa-se uma grande expansão no provimento de ofertas e soluções VoIP, uma vez que novas empresas entram no mercado apostando na alta margem de lucratividade do negócio.

Entidades governamentais, empresas provedoras do serviço, clientes empresariais e os da última milha (usuário residencial, produto de massa) são grupos que têm, constantemente, demonstrado tendências para o uso da tecnologia, assim como para a substituição da telefonia convencional. Eles percebem na tecnologia um grande facilitador como meio de redução de custos ou aumento do lucro.

A implementação do VoIP por provedores de serviços tem inserido novas opções de serviço a preços reduzidos em comparação com a telefonia convencional. As novas tecnologias, que são inerentes ao VoIP, possibilitam a criação de novos produtos e serviços customizados para o usuário corporativo ou doméstico, como o caso das comunicações unificadas (UC – *Unified Communications*). A redução de custos operacionais, proporcionada pela desativação das antigas e grandes centrais telefônicas e consequente diminuição da necessidade de manutenção da infraestrutura necessária, anima todos os empreendedores que apostam no VoIP para a obtenção de rápido retorno financeiro.

Porém, aspectos de segurança, desempenho, qualidade da rede (QoS) e expectativa de qualidade (QoE) normalmente não são levados em consideração pelos provedores de serviço segundo a avaliação feita pela pesquisa da IN-STAT e que foi confirmada em algumas das empresas entrevistadas.

A pesquisa do IN-STAT em 2006 relata que 40% dos provedores de VoIP no mundo não possuem qualquer preocupação com segurança, apesar de terem recursos financeiros para implementá-la e terem algum conhecimento sobre os perigos existentes. Além disso, os dispositivos que provêem o serviço de VoIP estão compartilhando os mesmos recursos das redes de dados, absorvendo todas as eventuais vulnerabilidades existentes.

Regulamentação

Para o Brasil, ainda não há uma definição sobre a regulamentação do VoIP. Esperava-se que fosse divulgada uma nova SCM ainda em 2008, mas a expectativa é de que venha a ser regulamentado como telefonia convencional principalmente pelo fato de a portabilidade permitir que um número fixo STFC seja portado para uma estrutura VoIP.

O não posicionamento da agência reguladora, somado à falta de preparo empresarial e à facilidade de implementação de um pequeno ambiente VoIP, permitem a criação

de um ambiente propício para que o usuário compre um produto e não tenha a qualidade esperada (QoE) de um serviço de telefonia.

Recentemente, uma das maiores operadoras de VoIP no país foi proibida de comercializar seus produtos na cidade de Londrina (PR) por causa da má prestação dos serviços. A questão é que a decisão partiu do Procon, e não do órgão regulamentador.

Estudos do IETF (*Internet Engineering Task Force*) indicam que para o *codec* G.729 devem ser evitadas perdas superiores a 2% dos pacotes; já o G.711 aceita uma perda de até 5%. O ITU2 alega e exemplifica por meio das recomendações G.113 e G.114 que perdas de 1% já são suficientes para gerar insatisfação no uso do serviço, e que uma latência de 150 ms é suficiente para praticamente inviabilizar o serviço. Alguns provedores de serviço nos EUA seguem essas recomendações à risca, inclusive em seus SLAs. A **tabela 1** mostra alguns exemplos de SLAs praticados nos EUA.

A vantagem do uso do G.729 em vez do G.711 basicamente é a banda ocupada. O primeiro demanda, para cada chamada, pacotes de 8 KB, enquanto o segundo faz uso de pacotes com 64 KB, sendo este o padrão de compressão da telefonia convencional. A desvantagem é a qualidade do áudio decorrente da amostragem de áudio gerada. O G.729 é recomendado para ambientes de rede com baixa capacidade de rede.

As características destes codecs se devem ao fato de que o G.711 opera em modo PCM (modulação por pulso) e não faz distinção entre voz humana e ruído; já os outros codecs, que possuem uma alta taxa de compressão, são chamados de *vcoders*. Eles distinguem o que é uma voz humana de outros sons e realizam uma compressão muito maior em virtude disso.

Com esse benefício surge um problema. Em razão da existência de filtros especializados, demanda-se um poder computacional significativamente maior, e não é possível obter a mesma qualidade de voz (QoE) quando se compara um vcoder à tecnologia PCM.

Independente do codec, a percepção do usuário é a mesma ao se iniciar a perda de pacotes. Observa-se a voz metalizada e o efeito de eco seguido pelo corte da voz. Quanto maior a compressão, mais rápida é a percepção do problema.

Este é um grande problema ao se dimensionar um sistema com segurança em redes para VoIP. Quanto maior for a quantidade de elementos de segurança, maior o risco do atraso devido ao tratamento dos pa-

Tabela 1: SLA de redes nos EUA

Provedor	Latência	Jitter	Perda de pacotes
AT&T Managed Internet Service [1]	Máximo de 37 ms	Não fornecido.	Máxima perda de 0.05 %
Verizon Voice over IP [2]	Máximo de 45 ms	Máximo de 1 ms	Máxima perda de 0.05%
Qwest SLA [3]	Máximo de 42 ms	Máximo de 2 ms	Máxima perda de 0.1%
Verio SLA [4]	Máximo de 50 ms	Máximo de 10 ms	Máxima perda de 0.1%
Internap SLA [5]	Máximo de 45 ms	Menor do que 0.5 ms	Máxima perda de 0.3%

cotes. Existe ainda a possibilidade do uso de criptografia, mas ela deve ser muito bem estruturada em conjunto com a solução de rede. E, como explicado anteriormente, quanto maior a compressão, mais difícil é ter uma qualidade comparável ao PCM. Com isso, a tolerância à latência é muito baixa.

Estes são bons motivos para que não sejam utilizados equipamentos não especializados no ambiente VoIP. A escolha do codec correto, por exemplo, apresenta-se como fator de fundamental importância para viabilizar um processo de comunicação eficiente. A probabilidade do mau dimensionamento e de equipamentos inapropriados inserirem problemas dos mais diversos tipos aumentará exponencialmente com o aumento do risco de falhas na rede.

Conclusões

Um serviço telefônico não pode, de maneira alguma, ser tratado como se fosse mais um serviço de dados na rede IP, principalmente quando passa a assumir o papel da telefonia convencional. Mesmo em um ambiente controlado é necessário ter em mente que telefonia é um serviço crítico e não apenas mais uma ferramenta de trabalho como email, aplicativos de escritório ou Internet.

Em uma emergência, o contato mais rápido para ajuda é o telefone. É necessário garantir alta disponibilidade do serviço em qualquer situação crítica e de emergência.

A impressão atual é que muitas empresas que fornecem o serviço não possuem o conhecimento da criticidade da telefonia. Em parte, a agência reguladora tem sua parcela de responsabilidade no que se diz respeito à falta de regulamentação, deixando a responsabilidade para outros órgãos, como o Procon.

Muitas implementações são elaboradas com foco somente no custo final do serviço para o usuário, desprezando questões relacionadas à qualidade do serviço. Existem empresas que passam essa responsabilidade para o usuário final, instruindo-o a não utilizar nenhum outro serviço de dados enquanto estiver fazendo uso do VoIP.

Como uma implementação dessa natureza conseguirá se manter na nova cultura de troca de arquivos, uso de serviços de *multicast*, vídeo sob demanda e as futuras implementações de IPTV?

Um passo importante para começar uma mudança do cenário seria uma regulamentação adequada desse tipo de serviço com a conscientização das premissas necessárias, assim como maior controle sobre seus provedores e a qualidade dos serviços. Ou então, estabelecer métodos de divulgação do serviço para que ele só possa ser comercializado com explicações claras sobre suas limitações, índices de disponibilidade, dependência de acesso à rede de dados e outros que sejam particulares de cada provedor. Porém, na maioria dos casos, isso não ocorre para os usuários residenciais. ■

Mais informações

[1] VoIP da AT&T: http://businessesales.att.com/products/product_mis.jhtml

[2] VoIP da Verizon: http://www.verizonbusiness.com/terms/us/products/advantage/

[3] VoIP da Qwest: http://www.uswest.org/legal/docs/IA_SLA_V4_052308-.pdf

[4] VoIP da Verio: http://www.verio.com/global-ip-guarantee/

[5] VoIP da Internap: http://www.internap.com/product/technology/performanceip/files/DS_IS_sla_0208.pdf

Sobre o autor

André Cabral Dutra (andrec@andrecabral.eti.br) é tecnólogo em administração de redes e tem pós-graduação em gestão de segurança da informação.

Tecnologias e ferramentas para redes mais seguras

Estratégias de segurança

Seus dados estão seguros? O perímetro da sua rede está protegido? Nesta edição vamos examinar algumas técnicas desenvolvidas por especialistas para tornar suas redes mais seguras.

por Joe Casad e Rafael Peregrino da Silva

Administradores de verdade pensam em segurança o tempo todo. Mesmo que você seja apenas um usuário ocasional, vale a pena saber o que os agressores sabem. Serviços que costumavam ser seguros hoje estão abertos à visitação pública, a menos que você acompanhe as mudanças que estão acontecendo nos ambientes de TI. Nesta edição, vamos examinar algumas das estratégias de segurança utilizadas por especialistas nessa área.

Vamos começar com um artigo que descreve como usar senhas descartáveis (*one-time passwords*) para incrementar a segurança com a autenticação de dois fatores em sites. Você vai conhecer a biblioteca PHP *OTPauth*, e vamos fornecer um exemplo detalhado sobre como configurar o seu próprio sistema de senhas descartáveis.

Em seguida, abordaremos a proteção de redes de telefonia IP (VoIP) contra abelhudos e outros intrusos. Você vai conhecer algumas tecnologias de criptografia disponíveis para ambientes VoIP e pegar algumas dicas sobre como isolar e proteger sua rede de comunicação por voz.

O terceiro artigo examina o estado da arte em segurança para ambientes de redes sem fio. Vamos mostrar algumas das estratégias para você tirar o máximo de segurança de uma rede WEP – quando não houver outra alternativa mais segura – e como maximizar a segurança de protocolos mais recentes, como WPA e WPA2.

No que tange à segurança, é impossível ter a rede perfeita ou dispor das ferramentas perfeitas o tempo todo – afinal, o mundo da segurança de redes está em constante mutação – e, como não há como conhecer esse mundo com perfeição, talvez a melhor estratégia seja mesmo informar-se o tempo todo. Nesse sentido, a **Linux Magazine** deste mês lhe oferece um bom apanhado de informações em estratégias de segurança. Boa leitura! ■

Índice das matérias de capa

Implementação de um sistema one-time password para Web

Uma chave, dupla proteção

Acrescente segurança ao seu site com um sistema one-time password.

por James A. Barkley

O sistema de autenticação com dois fatores (*two-factor authentication*) utiliza uma combinação de dois fatores diferentes para autenticar um usuário. Em oposição a um único fator, os dois fatores fornecem uma melhor segurança na autenticação. Os fatores combinados podem consistir em:

- algo que o usuário sabe (senha ou PIN),
- algo que o usuário tem (smartcard, certificados PKI, RSA SecurID),
- algo que o usuário é ou faz (impressão digital, sequência de DNA).

A primeira opção é a escolha fácil. Senhas são usadas com vários objetivos. A terceira opção geralmente é biométrica – uma má escolha para o ambiente web. "Algo que o usuário tem" é o melhor segundo fator para autenticação. Quase todas as soluções de autenticação de dois fatores baseadas na web disponíveis hoje envolvem alguma forma de token em hardware, como o RSA SecurID. Distribuir esses tokens para os usuários não é nem barato nem escalável em preço. Uma empresa pode ser capaz de comprar tokens para 1.000 usuários, mas basta um bom post de blog para ela se ver com 30 mil usuários da noite para o dia. Exigir que os usuários obtenham um token pessoalmente é trabalho demais para a maioria dos usuários. Além disso, os tokens precisam ser sincronizados com softwares especiais no servidor, o que pode demandar uma licença proprietária.

Uma alternativa menos custosa e mais escalável para autenticação de dois fatores na Web é um sistema de *one-time password* (OTP). A edição de novembro de 2008 da **Linux Magazine** ofereceu uma introdução às OTPs [1] que se concentrava primariamente na autenticação em estações de trabalho; porém, tarefas como verificar uma conta de banco a partir de uma rede pouco confiável imploram por alguma forma de autenticação de dois fatores, e um sistema OTP costuma ser uma solução prática. Este artigo descreve como adicionar a segurança das OTPs a um site.

OTP na Web

A RFC 2289 [2] define um sistema OTP derivado da tecnologia Bellcore S/KEY (RFC 1760). Se implementada corretamente, ela oferece uma solução de autenticação de dois fatores barata para sites. Imagine que um técnico de help desk com privilégios administrativos para um site chegue a uma página administrativa que gera uma lista de 30 pares OTP de número e chave. Depois, a lista é entregue ao usuário em mãos. Essa senha então se torna algo que o usuário tem – o segundo fator – e ,como jamais foi transmitida eletronicamente, ela fornece um grau elevado de segurança.

Se o site não se importar com transmissões eletrônicas para seus domínios confiáveis, o administrador pode enviar a lista por fax ou até mesmo por email para o usuário. A

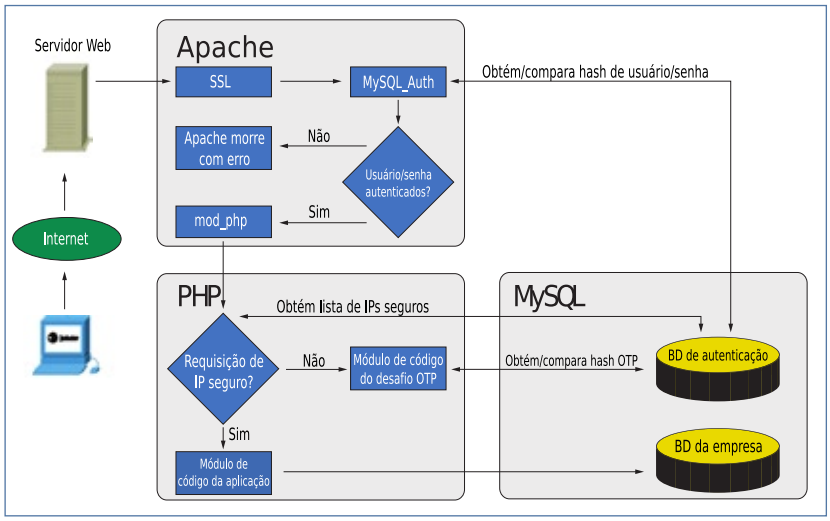


Figura 1 Cenário de login via Web com OTP.

partir de um café em Amsterdã, por exemplo, o usuário já pode digitar um nome de usuário e uma senha convencionais. Se essa autenticação inicial tiver sucesso, o servidor propõe um desafio que requer uma resposta com a OTP correspondente. Depois desse login, a OTP é imediatamente invalidada para uso futuro, o que significa que jamais será usada para um ataque. No próximo login, o usuário digitará a próxima OTP da lista.

Forçando o usuário a se autenticar por meio de dois mecanismos diferentes, a autenticação por dois fatores oferece uma alternativa muito mais segura para o login via Web. Esse cenário básico pode ter inúmeras variações. Por exemplo, um usuário poderia associar um número de celular à conta; depois, ao fazer login, o sistema enviaria a OTP por mensagem de texto. Ou, um usuário poderia gerar senhas OTP a partir de um programa executado em seu PDA. Uma outra vantagem desse cenário é que a implementação pode fornecer à OTP um componente temporal para que a senha expire após 60 segundos, exatamente como o RSA SecurID, mas isso exige que o PDA sincronize a aplicação com o servidor.

Ferramentas OTP

Há várias bibliotecas OTP para SSH, console e login de rede, além de diversas outras para o *SquirrelMail* e Palm Pilots, mas não é fácil encontrar alguma de código aberto para ambientes web. Um sistema OTP que segue a RFC 2289 foi testado e liberado sob a GPL: chama-se *OTPauth* [3] e é uma biblioteca em PHP.

A OTPauth usa o algoritmo SHA1 para *hashing* e já é usada com sucesso num site com várias centenas de usuários, há mais de dois anos. Existe também outra biblioteca em PHP, a *otp*, disponível no SourceForge [4].

Os desenvolvedores da otp pretendem ter em breve um demo da biblioteca.

Existem várias ferramentas em Java que ajudam na tarefa de construir e validar OTPs [5], mas não foi possível encontrar uma biblioteca web completa (por exemplo, algo que integre uma implementação

Exemplo 1: Banco de dados para autenticação

```
01 CREATE TABLE user (
02    user_id int(11) NOT NULL AUTO_INCREMENT,
03    user_name text NOT NULL,
04    user_pw varchar(32) NOT NULL DEFAULT '',
05    realname varchar(32) NOT NULL DEFAULT '',
06    STATUS char(1) NOT NULL DEFAULT 'A',
07     add_date int(11) NOT NULL DEFAULT '0',
08    confirm_hash varchar(32) DEFAULT NULL,
09    phone_number varchar(20) NOT NULL DEFAULT '',
10    last_pw_change int(11) NOT NULL DEFAULT '0',
11    otp_enabled tinyint(1) NOT NULL DEFAULT '0',
12    PRIMARY KEY (user_id),
13 ) TYPE=MyISAM;
14
15 CREATE TABLE session (
16    user_id int(11) NOT NULL DEFAULT '0',
17    session_hash char(32) NOT NULL DEFAULT '',
18    ip_addr char(15) NOT NULL DEFAULT '',
19    otp_auth tinyint(1) NOT NULL DEFAULT '0',
20    time int(11) NOT NULL DEFAULT '0',
21    locked tinyint(1) NOT NULL DEFAULT '0',
22    PRIMARY KEY (session_hash),
23 ) TYPE=MyISAM;
24
25 CREATE TABLE otp (
26    user_id int(11) NOT NULL DEFAULT '0',
27    sequence int(11) NOT NULL DEFAULT '0',
28    otp char(60) NOT NULL DEFAULT '',
29    PRIMARY KEY (session_hash),
30 ) TYPE=MyISAM;
```

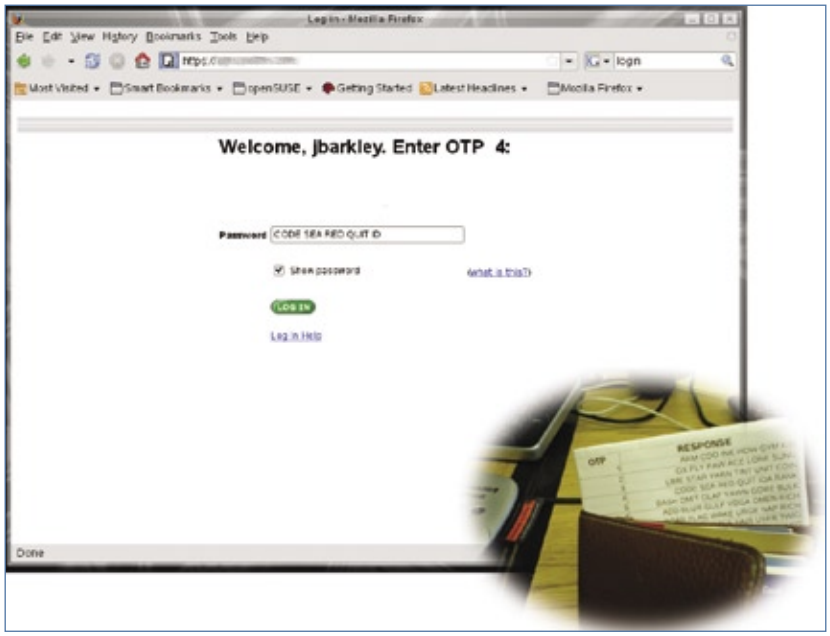


Figura 2 Login com uma one-time password.

completa de desafio/resposta numa aplicação J2EE).

A biblioteca *AuthSub* [6] do Google não segue estritamente a RFC 2289, mas permite a autenticação do tipo OTP segura nas aplicações do Google. Será interessante acompanhar se o Google vai continuar desenvolvendo essa solução ou migrar para o *OAuth*. Há um punhado de outros pacotes que oferecem uma solução OTP personalizada exclusivamente para seus softwares, como um plugin para o CMS *Joomla* [7].

Este artigo descreve como configurar um sistema OTP com a biblioteca de código aberto OTPauth. As outras ferramentas operam de forma semelhante e, para explorá-las, basta conferir a documentação em seus respectivos sites.

OTP self-service

Imagine um banco que queira incentivar boas práticas em segurança mas não possa insistir na autenticação de dois fatores sem assustar metade de seus correntistas. O banco deseja um sistema que suporte a opção OTP para os "descolados", mas sem arriscar o modelo de negócios forçando restrições a quem não as deseja.

A solução precisa fornecer um meio para o usuário visitar uma página de preferências e especificar que o programa exija a autenticação com dois fatores ao fazer login a partir de um computador diferente daquele sendo usado no momento. O usuário depois gera uma lista pessoal com 30 pares de número/chave OTP a partir da página de preferências. Na próxima vez que o usuário acessar a conta a partir de um local não confiável, o site pedirá que ele forneça uma OTP junto com o nome de usuário e a senha.

Exemplo 2: mod_auth_mysql no httpd.conf

```
01 #carrega a biblioteca do mod_auth_mysql
02 LoadModule mysql_auth_module modules/mod_auth_mysql.so
03
04 #faz o mod_auth_mysql conectar-se ao banco de autenticação
05 #Os parameters são:<br>#Auth_MySQL_Info hostname user password
06 Auth_MySQL_Info localhost usuário_bd_autenticação minhasenha<br>
07
08 #redireciona logins sem sucesso para a página de rejeição
09 ErrorDocument 401 /chapter14/rejection.html
10
11 #define as tabelas e colunas do MySQL para autenticação
12 AuthName "Meu Site"
13 AuthType Basic
14 Auth_MySQL_DB auth_db
15 Auth_MySQL_Encryption_Types MySQL
16 Auth_MySQL_Password_Table user
17 Auth_MySQL_Username_Field user_name
18 Auth_MySQL_Password_Field user_pw
19
20 require valid-user
```

Exemplo 3: Lógica de OTP com PHP

```
<?php

...
...

//obtém o ID do usuário do global definido pelo Apache ou método
//similar
$uid = user_getid();

//cria a sessão do usuário
$session = user_getsession($uid); //tenta obter sessão do BD
if (!$session) {
   $session = user_create_session($uid); //cria entrada na tabela de sessões
}

//verifica se o usuário já está autenticando
//isso impede uma condição de corrida especificada na RFC 2289
while ($session['locked']) {
   /* girar até o bloqueio ser liberado ou chegar um timeout */
   $session = user_getsession($uid);
   if (spinlock_timeout_reached()) {
      header("Location: http://www.example.com/retry.php");
      exit;
   }
}

//bloqueia a conta durante a autenticação
set_session_lock($uid); //define a flag "bloqueado" na tabela de sessões

//veririca se a autenticação otp está ativa na conta
//user_getotpauth() pconsulta o banco e retorna
//a flag otp_enabled da tabela de usuários
$otp_auth_enabled = user_getotpauth($uid);

if ($otp_auth_enabled) {
   if ($session['otp_auth']) {
   /* sucesso,o usuário já se autenticou com otp */
} else {
   /* usuário fez login, mas sem otp */

   //untrusted_host() compara o IP da sessão atual à
   //lista confiável especificada pelo usuário
   if (trusted_host($uid)) {
      /* usuário está em endereço que não requer a OTPauth */
   } else {
      /* usuário precisa da autenticação por OTP */
      header("Location: http://www.example.com/otp_auth.php");
      exit;
   }
}
}

//em todos os casos exceto o de OTP necessário,
//o usuário acaba passando neste ponto.
//liberar o bloqueio e proceder ao código específico da página.
unset_session_lock($uid);

...
...

?>
```

O primeiro passo é fornecer autenticação básica com um nome de usuário e uma senha. Grandes bibliotecas e metodologias padrão dão conta disso, seja com uso dos arquivos `.htacess` do Apache ou com a validação a partir de um banco de dados na camada da aplicação, ou com a validação pelo Apache com o *mod_auth_mysql*. Para usar controles de segurança em múltiplas camadas, é possível usar a arquitetura mostrada na **figura 1**.

O banco de dados de autenticação fica armazenado separadamente do banco principal da empresa, e guarda o nome de usuário, senha e informações da OTP. O **exemplo 1** mostra os comandos `CREATE` do MySQL que contêm todas as informações necessárias para o banco de dados inteiro, que o *mod_auth_mysql* verifica em busca do nome de usuário e senha.

Primeiramente é preciso instalar o *mod_auth_mysql* [8]. Com ele instalado, configure-o acrescentando as linhas do **exemplo 2** ao arquivo `httpd.conf`. Lembre-se de alterar as configurações para o site em questão.

Agora é possível adicionar o código na camada da aplicação para verificar a autenticação por OTP. O usuário jamais chegará à aplicação sem digitar nome e senha corretos, mas, depois que ele o fizer, é preciso garantir que a autenticação com dois fatores esteja disponível.

Primeiro, a aplicação precisa determinar se o usuário ativou a autenticação por OTP para a conta. Em caso positivo, a aplicação precisa comparar o IP ou hostname atual com aqueles listados na conta do usuário. Se a autenticação por OTP não estiver ativada pelo usuário ou se ele se encontrar num endereço confiável, a aplicação permite acesso à(s) página(s) web pedidas. O código do **exemplo 3** poderia ser incluído nas páginas da aplicação no início ou em uma função, dependendo do momento em que se deseja que ele seja executado.

Quando o usuário é redirecionado para o site *otp_auth*, a biblioteca OTP se encarrega do desafio/resposta. O **exemplo 4** mostra uma página básica que apresenta o desafio para o usuário e valida a resposta. Essa página é intencionalmente vazia, pois a apli-

Exemplo 4: Página de autenticação com OTP

```
01 <?php
02 /* LICENSED UNDER THE GPL */
03 # se já clicaram no botão de login
04 if ($login) {
05    $success = valid_otp($form_challenge_response, 06$user);
06    if ($success) {
07       /* atualiza estado da sessão/autenticação e redireciona
   para recursos do sistema e depois sai */
08       header("Location: http://www.example.com/".$page);
09       exit();
10    }
11 }
12
13 $sequence = get_otp_seq($uid);
14 if ($sequence == -1) {
15    /* imprime mensagem de erro e sai */
16 }
17
18 print "
19 <p>
20 <FORM ACTION=\"$PHP_SELF\" METHOD=\"POST\">
21 <p>
22 <INPUT TYPE=\"TEXT\" NAME=\"user\" VALUE=\"$user\">
23 <p>
24 Enter One-Time Password for Challenge number <B>$sequence</B>:
25 <br><INPUT TYPE=\"TEXT\" NAME=\"form_challenge_response\"
  VALUE=\"$form_challenge_response\" SIZE=\"31\">
26 <p>
27 <INPUT TYPE=\"SUBMIT\" NAME=\"LOGIN\" VALUE=\"Login\">
28 </FORM>
29 <P>
30 ";
31 ?>
```

Exemplo 5: Planilha para geração de lista de OTPs

```
01 $otp_list = generator($uid);
02 /*
03 criação de uma planilha para formatação dos dados
04 */
05 header("Content-Type: application/vnd.ms-excel");
06 header("Expires: 0");
07 header("Cache-Control: must-revalidate, post-check=0, pre-check=0");
08
09
10 print "<TABLE BORDER=1>";
11 print "<TH>Número de sequência</TH><TH>Senha</TH>";
12 while (list($key, $val) = each($otp_list)) {
13    print "<TR><TD>$key</TD><TD>$val</TD></TR>";
14 }
15 print "</TABLE>";
```

cação ainda não está convencida de que o usuário é autêntico. Ao evitar fornecer todas as bibliotecas normais ou código em JavaScript, ou até mesmo o visual do site final, podemos reduzir os vetores de ataque nessa página. Usar uma boa biblioteca OTP simplifica essa lógica da aplicação para uma quantidade trivial de código com chamadas de funções como `valid_otp()` e `get_otp_seq()`. O código do **exemplo 4** produz uma tela semelhante à da **figura 2**.

Por último, não se esqueça de fornecer a seus usuários as ferramentas para ativar o OTP em suas contas, gerar suas OTPs e gerenciar suas listas de confiança. O **exemplo 5** cria uma página bem leve que gera uma planilha de OTPs, mas certifique-se de que quando um usuário ativar OTP em sua conta ele não sofra logout antes de gerar a lista de OTPs.

Prepare-se para ter um mecanismo para zerar as listas de OTPs. Isso poderia ser feito por meio de um canal de suporte via telefone, email ou até IRC, ou uma página automatizada, mas de qualquer forma o usuário precisará fornecer provas de sua identidade ou algo mais como pergunta de segurança. Além disso, não se esqueça das outras verificações de segurança – nenhum dos exemplos deste artigo valida dados de entrada, por exemplo.

Sem token

A especificação RFC 2289 para solução de *one-time password* pode oferecer autenticação com dois fatores; porém, jamais será tão segura quanto uma alternativa baseada em token. Muitas soluções que usam tokens exigem que se concatene um PIN à OTP para criar o segundo fator, o que aumenta bastante a segurança. Além disso, as soluções com token são projetadas para ser à prova de bisbilhoteiros, caso alguém tente fazer a engenharia reversa do algoritmo de geração de senha. Por último, as ferramentas baseadas em tokens utilizam a hora como dado, então podem mudar a cada minuto (ou outra unidade de tempo), o que significa que é muito difícil um agressor obter uma OTP que o usuário ainda não tenha usado. Com soluções que requerem uma lista OTP, um agressor que conseguir uma foto da lista (ou encontrar uma lista usada na rua) tem acesso a respostas OTP futuras.

O sistema OTP definido pela RFC 2289 oferece uma solução aberta e escalável para a autenticação baseada na Web. É até possível integrar um sistema OTP ao telefone celular do usuário. A OTP baseada na Web possui seus próprios riscos e vetores de ataque, e provavelmente jamais será tão segura quanto soluções baseadas em hardware como o RSA SecurID. Apesar disso, a OTP, combinada com um esquema convencional de autenticação via Web, é um excelente candidato para a autenticação com dois fatores. ■

Mais informações

[1] Udo Seidel, "Chaves nunca repetidas": http://lnm.com.br/article/2415

[2] RFC 2289: http://www.apps.ietf.org/rfc/rfc2289.html

[3] OTPauth: http://code.google.com/p/otpauth/

[4] otp: http://sourceforge.net/projects/otp/

[5] OTPs em Java: http://tinyurl.com/ablhoh

[6] Biblioteca AuthSub: http://tinyurl.com/5c6kat

[7] Plugin OTP do Joomla: http://tinyurl.com/bgsj6k

[8] Módulo mod_auth_mysql para Apache: http://tinyurl.com/26rboa

[9] Ataque de phishing em OTPs: http://tinyurl.com/arypv

Tornando redes VoIP seguras

Segure a ligação!

Para instalar uma escuta clandestina e espionar conversas telefônicas na rede local, bastam ferramentas padrão do Linux. Saiba como se proteger.

por Christoph Egger e Michael Hirschbichler

Muitas empresas de pequeno e médio portes simplesmente conectam seus novos sistemas VoIP a redes locais já existentes. Funcionários externos dão telefonemas via Internet e filiais remotas utilizam uma conexão padrão para falar com a central. Infelizmente, esse tipo de instalação não fornece nem de longe a infraestrutura de segurança mínima para ambientes VoIP.

Este artigo vai abordar alguns dos principais perigos que afetam infraestruturas de telefonia IP, bem como descrever algumas estratégias e protocolos opcionais para proteger a comunicação por voz usando a Internet.

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com
   ;branch=z9hG4b42
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>
   ;tag=42
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Max-Forwards: 70
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
                                  SIP

v=0
o=alice 53655765 2353687637 IN IP4
   pc33.atlanta.com
s=-
t=0 0
c=IN IP4 pc33.atlanta.com
m=audio 3456 RTP/AVP 0 1 3 99
a=rtpmap:0 PCMU/8000
                                  SDP
                                SIP/SDP
```

Figura 1 Iniciação de uma conexão VoIP – uma solicitação Invite engloba os componentes SIP e SDP.

Conexão VoIP típica

O protocolo SIP (*Session Initiation Protocol* – Protocolo de Iniciação de Sessão, RFC 3261) [1] é o padrão aberto VoIP mais popular para iniciação, negociação e gerenciamento de conexões de voz via Internet. Em combinação com o protocolo SDP (*Session Description Protocol* – Protocolo de Descrição de Sessão, RFC 4566) [2], que lida com negociação de codificação e decodificação de áudio e vídeo, o protocolo SIP transmite informações sobre a conexão entre os participantes de uma ligação. Com a conexão estabelecida, os participantes enviam e recebem dados usando o protocolo RTP (*Realtime Transport Protocol* – Protocolo de Transporte em Tempo Real, RFC 3550) [3].

Suponha que um usuário *A* em um domínio *A* deseje iniciar uma conexão VoIP com o usuário *B* no domínio *B*. O usuário *A* precisa enviar uma solicitação *Invite* (ou seja, um "convite" – **figura 1**) ao servidor proxy SIP do seu próprio provedor. O servidor proxy – que é responsável por atender requisições, repassando os dados a outros servidores – realiza uma consulta à base DNS, procurando o servidor proxy SIP do domínio *B*, a quem envia a solicitação. O servidor proxy SIP do domínio *B* verifica o IP do usuário *B* em sua própria base DNS, bem como a porta registrada por esse usuário para a comunicação via VoIP, e enfim encaminha a solicitação ao sistema desse usuário.

Qualquer resposta do cliente do usuário *B*, que poderia ser *Ringing* ou *OK* quando o usuário aceita a chamada, é transportada de volta ao

usuário A usando o mesmo caminho de volta na rede.

Após a chamada ser ativada via SIP/SDP, o sistema estabelece uma conexão RTP direta entre o cliente do usuário A e o cliente do usuário B. Em algumas configurações de sistemas de telefonia IP, os provedores se utilizam de servidores proxy intermediários para dados de áudio e vídeo, com o objetivo de evitar problemas de NAT (*Network Address Translation* – Tradução de Endereços de Rede, **figura 2**). As vantagens desse proxy de mídia são:

- pelo menos um dos participantes não fica atrás de um gateway de NAT e
- os dois dispositivos ou clientes envolvidos na comunicação dispõem de um participante na outra ponta com um endereço IP público.

Para utilizar os proxies de mídia de forma transparente, os próprios proxies SIP trocam o endereço IP do outro participante na comunicação, contidos na área de dados do pacote SDP, pelo endereço IP e a porta do proxy de mídia. Essa técnica é muito conveniente para simplificar a passagem através de firewalls e de gateways NAT. Entretanto, ela também limita seriamente as alternativas para criptografar as transmissões de dados. Atravessar um gateway NAT pura e simplesmente, sem lançar mão da técnica supracitada, é uma tarefa extremamente complexa, principalmente nos contextos VoIP, SIP e RTP. Os interessados podem encontrar informações detalhadas (em inglês) a esse respeito em [4].

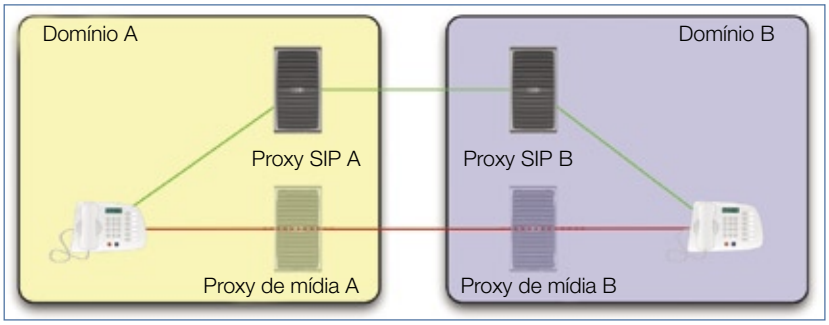


Figura 2 O trapézio SIP/RTP clássico: mensagens SIP (em verde) são transmitidas através de diversos saltos; no caso do RTP, elas geralmente são transmitidas de forma direta.

SIP

É muito difícil tornar o protocolo SIP seguro, graças a sua sinalização em texto puro e a sua arquitetura *hop-by-hop*, na qual a transmissão de pacotes se dá entre os diversos pontos de ligação (*hops*) entre computadores na rede; a informação enviada passa por cada hop, que indica qual o próximo ponto a recebê-la e assim sucessivamente. Assim, como os pacotes de dados atravessam múltiplos hops, os administradores de sistemas acham difícil conseguir um nível de segurança de ponta a ponta que garanta disponibilidade, confidencialidade e integridade. Cada elo da rota pode adicionar, remover ou manipular cabeçalhos de pacotes de dados, o que torna impossível a assinatura desses cabeçalhos.

A assinatura da área de dados (*payload*) dos pacotes SDP também não é uma alternativa satisfatória, pois o protocolo SIP inclui informação de sinalização confidencial no cabeçalho. Um agressor poderia, por exemplo, manipular o cabeçalho *From* de um pacote para alterar a identificação de quem estiver solicitando a chamada (o *caller ID*), o que afetaria a validade da assinatura da área de dados.

Os autores da RFC 3261, o atual padrão SIP, estavam bastante conscientes dos problemas de segurança que envolvem o protocolo, e resolveram a questão criando o que se convencionou chamar de SIPS (*SIP Secure Schema* – Esquema de Segurança SIP) [5]. O SIPS usa TLS (*Transport Layer Security* – Segurança da Camada de Transporte, RFC 4346) [6]

Quadro 1: TLS e IPSec

A *Transport Layer Security* se encontra acima da camada de transporte como versão melhorada do SSL (ver **figura 3**). Os participantes de uma comunicação são autenticados por meio do protocolo de comunicação TLS via criptografia assimétrica usando a abordagem das chaves públicas. No processo de autenticação recíproca, os parceiros de comunicação negociam uma chave simétrica especialmente gerada para a sessão. Para verificar a integridade, é acrescentado um código de autenticação da mensagem (MAC, *Message Authentication Code*), criptografado por um algoritmo SHA (preferível) ou MD5. Para maiores informações a respeito do TLS, refira-se à RFC 4346 [7].

Ao contrário do TLS, o *Internet Protocol Security* (IPSec) fica diretamente na camada IP. Desenvolvido durante a padronização do IPv6, o IPSec traz opções de segurança ao IPv4 e também trabalha de maneira transparente na camada de aplicação, de modo a fornecer proteção adicional a protocolos que não dispõem de mecanismos de segurança próprios. Como trabalha uma camada abaixo do TLS, o IPSec não precisa de um protocolo de transporte confiável para funcionar. Entretanto, ele também oferece suporte a protocolos baseados em pacotes (datagramas) como o UDP. O IPSec dispõe de suporte a dois modos diferentes: transporte e túnel.

Quadro 2: Skype e segurança

O provedor VoIP de tecnologia proprietária Skype já confirmou que não dispõe de uma política oficial de segurança. Uma pesquisa independente [12], para a qual foi dado acesso ao código-fonte do programa, confirma a existência de algoritmos de segurança atualizados e corretamente implementados no Skype. Entretanto, essa tecnologia não deixa de ser uma "caixa preta" e não está aberta a análise de especialistas no assunto. É impossível descartar a existência de falhas de segurança (intencionais, como *backdoors*, ou acidentais) ou aplicativos que permitam a agressores – ou mesmo "autoridades" – escutar clandestinamente e manipular a comunicação dos dados de voz e de vídeo, se presente.

como protocolo-base de sinalização entre dois hops e o último servidor proxy SIP (**quadro 1**).

A criptografia baseada em TLS acrescenta uma outra camada de segurança ao protocolo SIP. O SIPS, entretanto, não fornece uma solução completa. O problema dessa técnica é que o cliente *A* não tem como saber se cada hop que está no caminho de comunicação transmite a sua solicitação por meio de uma conexão TLS segura separadamente. O padrão define somente a segurança TLS para o servidor proxy no domínio *B*. A conexão entre o proxy e o cliente *B* não é criptografada.

Uma alternativa é fazer o cliente *B* funcionar como servidor TLS, mas somente poucos clientes realmente implementam esse recurso. Atualmente, os desenvolvedores estão trabalhando em um padrão que resolve esse problema, permitindo que o cliente *B* configure um túnel TLS até o proxy *B* e mantenha esse túnel aberto para todas as solicitações subsequentes.

Um outro problema inerente ao protocolo SIP é que se o protocolo UDP (*User Datagram Protocol* – Protocolo de Pacotes do Usuário, RFC 768) for usado na camada de transporte, é fácil para um agressor responder a uma solicitação manipulando cabeçalhos e, com isso, roubar a identidade do usuário e realizar chamadas em seu nome. Para evitar esse tipo de ataque, a maioria dos sistemas de proxy SIP oferecem suporte à autenticação de usuários. A autenticação no seu próprio servidor proxy SIP oferece proteção contra roubo de identidade e apropriação ilegítima de recursos. Quando o cliente se registra no servidor ou quando uma chamada é estabelecida, os sistemas geralmente usam um algoritmo de autenticação de acesso por resumo criptográfico (*digest*) semelhante ao utilizado pelo protocolo HTTP.

Tenha em mente, contudo, que somente a autenticação não é capaz de deter ataques *man-in-the-middle* (aquele em que o agressor se interpõe entre as duas partes que se comunicam, observando e interceptando suas mensagens), porque a integridade da solicitação não pode ser validada. Um agressor poderia "farejar" as informações de autenticação apenas vasculhando o tráfego da rede em texto puro, usando em sequência diferentes cabeçalhos para manipular a chamada.

A autenticação de cada uma das solicitações também não é uma opção viável em várias situações; por exemplo, requisições dos tipos *ACK* e *CANCEL* não esperam uma resposta e portanto não dispõem de suporte a *handshake*. Os padrões oferecem suporte a outras formas de autenticação SIP, mas o excesso de zelo com autenticação sobrecarrega o proxy SIP do provedor, aumentando atrasos e afetando a qualidade da conexão.

RTP

RTP não usa criptografia e depende do UDP – um serviço sem conexão – como protocolo de transmissão. Essa combinação simplifica para um agressor a manipulação do roteamento e dos dados sendo transmitidos, bem como sua "análise silenciosa". A RFC 3550 [8], que define o RTP, leva isso em consideração e fornece uma opção de criptografia de modo a garantir a confidencialidade dos dados.

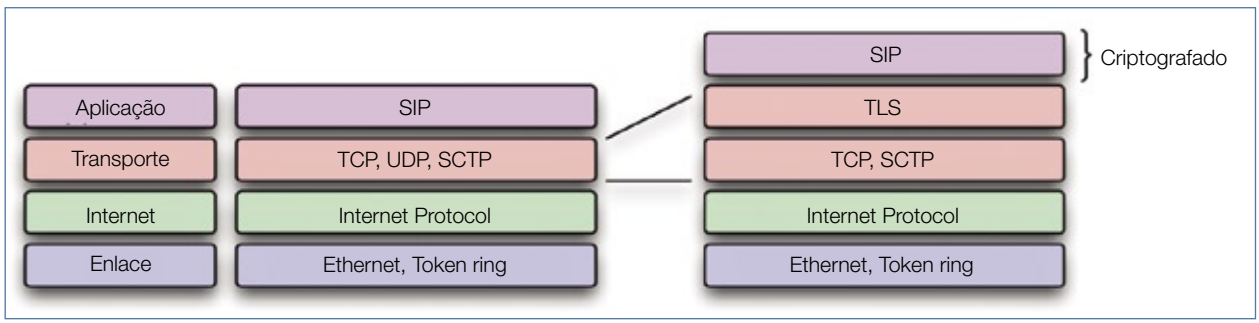


Figura 3 Como o TLS é transparente do ponto de vista da aplicação, ele oferece aos protocolos um mecanismo de segurança universal para aplicações que não dispõem de segurança própria.

A RFC 3550 indica o SRTP (*Secure Real-Time Transport Protocol* – Protocolo Seguro de Transporte em Tempo Real, **figura 3**) [9] como metodologia preferida para aumentar a segurança do protocolo RTP, criptografando o conteúdo da área de dados desse tipo de pacote. Como o SRTP não dispõe de mecanismos para criar e trocar chaves ele mesmo, ele depende de métodos externos para esse fim, tais como o *MIKEY* (*Multimedia Internet Keying* – Gerenciamento de Chaves via Internet para Multimídia, RFC 3830) [10].

A troca de chaves acontece durante a troca de mensagens que se inicia com o *Invite*. O método MIKEY dispõe de várias abordagens, incluindo o uso de chaves compartilhadas (*preshared keys*), em que o cliente *A* codifica a chave SRTP a ser trocada por uma senha secreta pré-estabelecida (MIKEY-PSK). Essa troca de chaves requer somente uma única mensagem; as duas chaves SRTP para essa sessão podem ser transmitidas com a solicitação *Invite*. O método MIKEY também oferece suporte para a troca de chaves baseada em uma infraestrutura de chaves públicas (MIKEY-RSA), em que aquele que inicia a comunicação envia ao cliente *B* uma chave pública e também a chave da sessão no pacote SDP.

Uma outra metodologia disponível no MIKEY (MIKEY-RSAR) funciona sem qualquer troca prévia de chaves ou do uso de uma infraestrutura de chaves públicas. O cliente *A* envia sua própria chave pública e o cliente *B* responde gerando a chave da sessão SRTP e a enviando para o usuário *A*.

Phil Zimmermann, inventor do PGP, desenvolveu também o ZRTP (Zimmermann Real-Time Protocol – Protocolo em Tempo Real de Zimmermann) [11], uma alternativa ao método de troca de chaves para iniciar uma chamada SRTP. O ZRTP não substitui o SRTP, mas estende os seus recursos.

Ao contrário do MIKEY, o ZRTP não utiliza a via de sinalização para transmitir a informação da chave SRTP, mas depende totalmente da via de dados multimídia. Uma conexão RTP é criada inicialmente e usada para trocar as chaves da sessão. Após a autenticação dos dados de voz, os clientes mudam para o protocolo SRTP, que é criptografado.

Muito embora criptografia de ponta a ponta seja possível e os dados estejam protegidos na rota entre o cliente do usuário e o provedor, a comunicação segura acaba no próximo proxy de mídia e não no destinatário da ligação. Isso confere aos provedor de rede um acesso completo as dados de áudio (e vídeo) do usuário. Da mesma forma

Quadro 3: S/MIME

O padrão S/MIME (*Secure/Multipurpose Internet Mail Extensions* – Extensões Multifunção Seguras para Mensagens de Internet) [5] fornece uma alternativa de segurança apenas para a área de dados (*payload*) do protocolo. O cliente envia uma solicitação *Invite* junto com um bloco SDP criptografado, de modo a assegurar a confidencialidade e a integridade dos dados SDP. Com, isso, ele também garante que os sockets que recebem e enviam os dados RTP do outro lado realmente pertencem ao parceiro autenticado. Mas isso só faz sentido se os dados de áudio (e vídeo, se existentes) forem criptografados usando RTP seguro [9].

Como muitos servidores proxy SIP tentam reescrever o cabeçalho SDP, por causa dos proxies de mídia, a criptografia do protocolo SDP pode criar problemas inesperados, e clientes em funcionamento que sejam capazes de lidar com S/MIME são desconhecidos.

De acordo com uma técnica definida na RFC 3893 [16], partes do cabeçalho SIP são assinadas juntamente com o SDP. O certificado usado para a assinatura S/MIME pode assinar também os cabeçalhos originais `From` e `To`. A solicitação SIP é dividida em três seções para isso:

- a requisição SIP propriamente dita;
- uma parte do tipo *message/sipfrag* que contém uma cópia de uma parte da solicitação SIP (`To`, `From`, informações de data e hora e outros detalhes);
- a assinatura da segunda parte.

Caso o destinatário conheça a chave pública do remetente, ele pode validar os dados relevantes. Implementações dessa técnica, conhecida como *Authenticated Identity Body* (AIB), ainda não são muito difundidas.

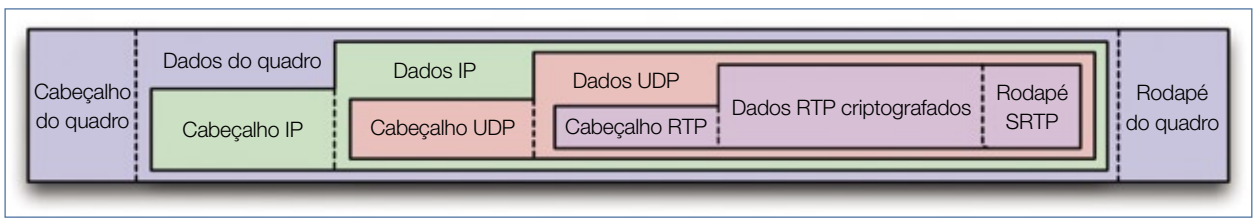


Figura 4 O protocolo SRTP fornece uma camada transparente entre o RTP e as camadas de transporte, oferecendo assim proteção contra o acesso aos dados de voz e garantindo a privacidade, a integridade e a autenticidade desses dados.

que acontece com tecnologias proprietárias como o Skype, o que está em questão é se o fabricante é ou não confiável (veja o **quadro 2**).

A rede

Os especialistas concordam que um passo importante na conquista da segurança em ambientes VoIP é a separação do tráfego da rede de telefonia IP dos dados circulando na rede local. A complexidade e a falta de segurança relativa naturais a redes de computadores oferecem muitas oportunidades para a criação de escutas clandestinas, entre outros tipos de ataque, e administradores de sistemas experientes estão bastante conscientes de que o acesso físico é praticamente sinônimo de segurança do sistema comprometida.

Para isolar a rede VoIP, prefira separá-la fisicamente da rede local ou configurar uma rede local virtual (VLAN). É óbvio que apenas separar o tráfego de dados de voz não vai protegê-lo em caso de acesso físico de um agressor a uma porta que estiver acessível na rede VoIP. Se uma porta estiver aberta, o agressor pode simplesmente se conectar a essa rede com um laptop e descobrir o endereço MAC de um telefone analisando os dados que circulam na rede. A melhor maneira de se combater esse tipo de invasão da rede local é usar autenticação adicional 802.1x [13][14] no switch da rede.

Para garantir comunicação VoIP segura para os funcionários da empresa que estiverem telefonando de fora, o mais sensato é rotear suas conexões por meio de um túnel VPN. Essa configuração deverá oferecer suporte para modalidades de codificação e decodificação cuja largura de banda não ultrapasse a de conexões discadas, tais como GSM [17]. Como alternativa, é possível usar SRTP e SIPS-S/MIME (**quadro 3**), se os clientes e servidores oferecerem suporte a essa opção, em razão do melhor desempenho desse protocolo.

A **figura 5** ilustra a combinação das técnicas abordadas neste artigo, na qual a infraestrutura VoIP fica isolada do restante do sistema. A conexão entre as filiais e a matriz usa múltiplos túneis VPN. Aliás, ao dimensionar a banda, é importante levar em consideração o consumo de banda extra do protocolo VPN. A conexão à rede de telefonia pública (PSTN, *Public Switched Telephone Network* – Rede Pública de Telefonia Comutada) pode ser configurada separadamente em cada filial ou roteada via matriz. Uma conexão adicional de becape para cada uma das filiais também é uma boa idéia. Isso mantém os escritórios das filiais disponíveis

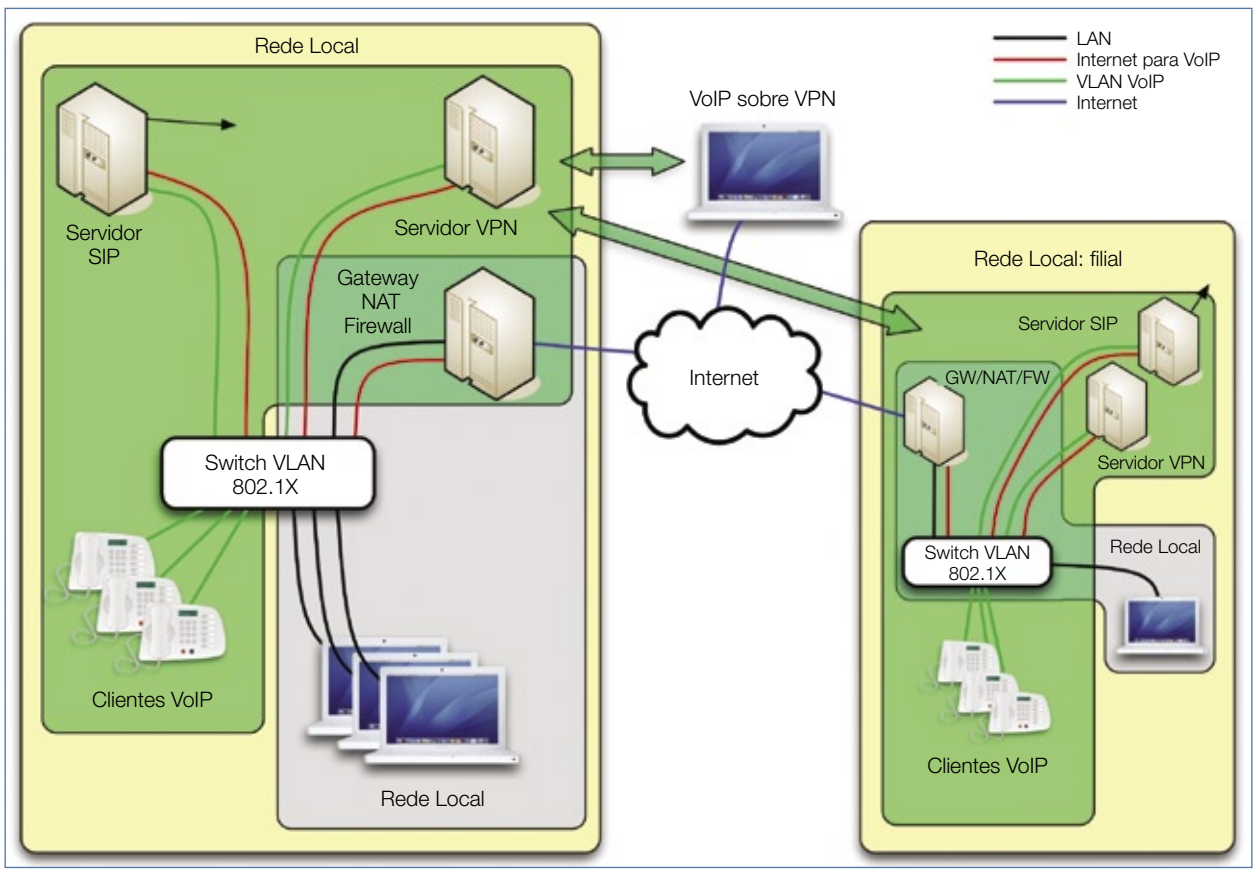


Figura 5 Uma rede VoIP com um esquema de segurança implementado.

Tabela 1: Clientes e servidores VoIP

Fabricante	TLS	IPSec	SRTP	Comentários
Grandstream	✔	✘	✔	Nem todos os equipamentos, situação “pendente”
Snom	✔	✘	✔	Nem todos os equipamentos, algumas incompatibilidades
Zultys	✘	✘	✔	AES
CrypTone	✘	✔	✘	
Servidores				
Asterisk	✔ (com patch)	✔ (SO)	✔ (com patch)	
Kamailio (OpenSER)	✔	✔ (OS)	✘	
OpenSIPS (OpenSER)	✔	✔ (OS)	✘	
SER	✔	✔ (OS)	✘	

mesmo que uma conexão IP falhe ou esteja sobrecarregada.

Conclusão

Se há um projeto de telefonia IP em andamento na sua empresa, faz sentido planejar a segurança desse projeto desde o início. As ferramentas de hardware e software de um ambiente VoIP fornecem uma grande quantidade de opções de segurança interessantes. Determine primeiro quais protocolos e quais componentes serão necessários na rede VoIP, e então adquira as ferramentas que disponham do suporte necessário. A **tabela 1** mostra os resultados de nossa pequisa sobre compatibilidade de telefones e sistemas VoIP comercializados por diversos fabricantes.

Se já houver uma rede VoIP em operação, técnicas simples como isolamento de redes pelo uso de tecnologia VLAN e o uso estratégico de alternativas de criptografia disponíveis serão de grande auxílio na configuração de um ambiente melhor e mais seguro para a comunicação por VoIP. ■

Mais informações

[1] Padrão SIP, RFC 3261: http://www.ietf.org/rfc/rfc3261.txt

[2] Padrão SDP, RFC 4566: http://www.rfc-editor.org/rfc/rfc4566.txt

[3] Referência para RTP: http://www.voip-info.org/wiki-RTP

[4] SIP, SDP, RTP e NAT: http://www.voipuser.org/forum_topic_7295.html

[5] Segurança SIP: http://tinyurl.com/bf5avg

[6] TLS: http://en.wikipedia.org/wiki/Transport_Layer_Security

[7] Protocolo TLS, RFC 4346: http://www.ietf.org/rfc/rfc4346.txt

[8] RTP, RFC 3550: http://www.ietf.org/rfc/rfc3550.txt

[9] RTP seguro: http://tinyurl.com/arxq3m

[10] MIKEY, RFC 3830: http://www.ietf.org/rfc/rfc3830.txt

[11] ZRTP: http://zfoneproject.com/zrtp_ietf.html

[12] Pesquisa de segurança sobre o Skype: http://www.anagram.com/berson/skyeval.pdf

[13] Padrão de autenticação 802.1x: http://en.wikipedia.org/wiki/802.1x

[14] 802.1x e agressores na mesma porta: http://tinyurl.com/d8wdpt

[15] S/ MIME: http://en.wikipedia.org/wiki/S/MIME

[16] SIP, RFC 3893: http://www.ietf.org/rfc/rfc3893.txt

[17] Comunicação GSM: http://en.wikipedia.org/wiki/Global_System_for_Mobile_Communications

Estratégias de segurança para redes sem fio

Segurança no ar

Redes sem fio oferecem acesso à Internet sem o inconveniente do emaranhado de cabos. Mas se você não tomar cuidado com a segurança, convidados indesejáveis podem aparecer.

por Erik Bärwaldt

As redes sem fio conquistaram espaço definitivo em muitos lares e redes de pequenos escritórios. Dispositivos como roteadores sem fio e modems DSL ou a cabo podem ser encontrados por preços baixos e até de graça, fornecidos pelo provedor de acesso à Internet. A maioria dos computadores atuais já é equipada com tudo o que se precisa para redes sem fio, e mesmo que seja necessário equipamento adicional para conseguir acesso sem fio, placas Wi-Fi PCI são baratas.

No entanto, a diversão acaba quando você descobre que alguém no escritório vizinho está usando sua rede sem fio para navegar na Internet. Mesmo que o clandestino ocasional não provoque um prejuízo na sua empresa, a navegação não autorizada pode ter consequências desagradáveis. Se o gentil vizinho eventualmente usar sua conexão para praticar atos ilegais, é você quem poderá ser visitado pela polícia. Mesmo que você não esteja no centro de uma organização criminosa da Internet, a simples presença de um usuário de fora na sua rede ocasiona diversos problemas eventuais. Por isso, é muito importante – principalmente se você usa equipamento defasado – empregar o máximo de recursos de segurança oferecidos pelos dispositivos sem fio. Este artigo dará algumas dicas para você obter melhor segurança em redes sem fio.

Figura 1 O logo Wi-Fi atesta a compatibilidade com os padrões.

Os dispositivos Wi-Fi 802.11b ainda usados em muitas redes sem fio comerciais pertencem a uma geração de hardware que data da década de 1990. Eles permitem uma velocidade de transferência de no máximo 11 Mbps, e a banda é compartilhada entre os clientes. Isso significa que, em circunstâncias técnicas ideais, pode-se esperar taxas de transferência de apenas 5 Mbps.

Para aumentar a velocidade, muitos fabricantes desenvolveram extensões proprietárias que prometiam melhores taxas de transmissão. Contudo, a maioria desses componentes proprietários funcionam apenas com dispositivos equivalentes e do mesmo fabricante. É praticamente impossível conseguir uma rede Wi-Fi segura onde há diversos dispositivos feitos por diferentes fabricantes, o que explica por que a Wi-Fi Alliance mantém sua própria certificação em paralelo ao padrão WPA. Os dispositivos precisam ser 100% compatíveis e possuir o selo de aprovação da Wi-Fi Alliance (**figura 1**).

Na época em que o padrão 802.11b foi desenvolvido, ninguém se preocupava de verdade com a segurança de redes sem fio. Além disso, muitos fabricantes de roteadores Wi-Fi desativam o mecanismo de segurança por padrão – uma má conduta que deixa a rede totalmente desprotegida, a menos que o usuário altere intencionalmente as configurações de segurança.

WEP

Configurações como essa que, acredite ou não, ainda existem, dão a qualquer um no alcance da rede

a possibilidade de se associar ao ponto de acesso e usar a rede. Para agravar ainda mais o problema, mesmo que o usuário habilite as opções de rede disponíveis no dispositivo, frequentemente essa segurança é ineficiente. O sistema de segurança *Wired Equivalent Privacy* (WEP) usado no padrão 802.11b rapidamente tornou-se inútil. Já em 2001, especialistas demonstraram que a criptografia por WEP possui sérias vulnerabilidades.

O método WEP usa chaves com comprimento de 40 ou 104 bits (232 bits em casos excepcionais). Todos os dispositivos na rede usam essa chave. O padrão permite utilizar um máximo de quatro chaves diferentes, mas não permite alterações dinâmicas. Além disso, cada pacote de dados inclui um vetor de inicialização (VI) com tamanho fixo de 24 bits.

Fabricantes de componentes Wi-Fi anunciam criptografia de 64 ou 128 bits no padrão 802.11b; contudo, o vetor de inicialização é transmitido sem proteção. O número máximo de valores possíveis para o vetor de inicialização é 17 milhões. Se ele for repetido diversas vezes numa sessão e a chave não for alterada, agressores podem calcular a chave e decifrar a transmissão. O agressor precisa apenas interceptar pacotes suficientes e realizar um ataque de força bruta para comprometer a chave.

Para uma rede com grande volume de tráfego, não demora para que o agressor intercepte pacotes suficientes para quebrar a chave. Já em redes doméstica pequenas, o invasor precisará capturar pacotes por mais tempo, mas há ferramentas que podem gerar tráfego no ponto de acesso e acelerar o processo de quebra da chave.

Outro meio de quebrar a chave de uma rede sem fio protegida por WEP é realizar um ataque de dicionário. Um ataque de dicionário consiste em usar diversas chaves (geralmente milhões delas) até que a correta seja aceita. É um método que obtém sucesso, mas exige mais tempo e capacidade computacional.

Veja as dicas do **quadro 1** para evitar ataques em redes WEP. A maioria dos defasados dispositivos WEP 802.11b não é compatível com padrões recentes, portanto, melhorar a segurança na rede sem fio significa adquirir novo hardware.

Sucessor: WPA

As muitas deficiências do WEP impulsionaram a Wi-Fi Alliance a desenvolver uma alternativa, o *Wi-Fi Protected Access* (WPA), para cobrir o buraco até que o novo padrão 802.11i pudesse oferecer mecanismos de segurança mais robustos. O WPA é um paliativo entre o WEP e o mais recente WPA2. Por um lado, ele utiliza um novo método de autenticação base-

Quadro 1: Dicas para redes WEP

Se você ainda usa uma rede sem fio compatível com o padrão WEP, pode melhorar a segurança com as seguintes medidas:

- Escolher uma chave compartilhada o mais longa possível e ter certeza de que é uma combinação arbitrária de letras e números. Isso reduz o risco de ataques de dicionário;
- Definir todas as quatro chaves e trocá-las em intervalos regulares para evitar ataques de força bruta;
- Se possível, desativar o servidor DHCP do roteador sem fio. Em seu lugar, definir endereços IP estáticos e o menor espaço de endereços possível;
- Alterar a senha de configuração do roteador. As senhas padrão dos roteadores comuns são amplamente conhecidas na Internet. A rede estará completamente aberta a ataques do agressor que ganhar acesso ao roteador;
- Desligar a visibilidade do SSID e *beacons* (*broadcast*) do ponto de acesso;
- Posicionar o roteador Wi-Fi de modo que a recepção seja adequada para você, mas não cubra a propriedade do vizinho. Lembre-se que as ondas também se propagam verticalmente;
- Se o dispositivo permitir, configure a potência de transmissão do roteador para não superar o mínimo necessário para suas próprias necessidades.

ado em chaves pré-compartilhadas, com senhas entre oito e 63 dígitos. Por outro lado, os desenvolvedores do WPA mantiveram o algoritmo RC4, que é sabidamente inseguro.

De acordo com a Wi-Fi Alliance, o uso do RC4 era necessário em função dos aspectos técnicos envolvendo os pontos de acesso da época. Esses dispositivos não tinham capacidade computacional interna suficiente para operar com um algorítimo como o AES por meio apenas de uma atualização do firmware.

Ao introduzir o WPA, os desenvolvedores modificaram os métodos de autenticação e criptografia para proporcionar mais segurança: os clientes usam chaves pré-compartilhadas ou (em redes sem fio de grande porte) um servidor *Radius* para se associar ao ponto de acesso. Depois da autenticação, o cliente e o ponto de acesso negociam uma chave individual de 128 bits para impedir que outra estação na Wi-Fi intercepte o tráfego de dados. Além desses melhoramentos, o WPA usa um vetor de inicialização de 48 bits. Renegociações periódicas da chave entre o ponto de acesso e o cliente acrescentam segurança ao padrão WPA, eliminando a possibilidade de um agressor iniciar um ataque de força bruta contra grandes volumes de pacotes de dados interceptados.

Veja no **quadro 2** como tornar o WPA ainda mais seguro.

Estado da arte

O WPA2, que foi introduzido em 2004, torna a rede sem fio ainda mais segura. Os desenvolvedores finalmente retiraram os recursos defasados da infraestrutura de segurança sem fio ao substituir o inseguro algoritmo RC4 pelo padrão AES superior, por exemplo. A essa base melhor, o novo padrão incorporou ainda os métodos de autenticação e criptografia do WPA. Graças a essas melhorias, agressores não podem mais se beneficiar da captura de tráfego de uma rede sem fio por horas ou dias e executar um ataque de força bruta sobre os resultados.

O WPA2 introduz um padrão em duas partes: o subgrupo WPA2 *personal* especifica um padrão com menos recursos, destinado a redes domésticas e de pequeno porte. Apesar de essa variação oferecer todos os recursos básicos e populares de segurança, não é capaz de trabalhar com um

Quadro 2: Dicas para redes WPA/WPA2

É melhor prevenir do que remediar. Como em todos sistemas de autenticação por senha, escolha senhas o mais longas possível e tenha certeza de que todas contêm combinações arbitrárias de números e letras.

- Configure o roteador sem fio para negociar automaticamente novas chaves com os clientes em intervalos regulares. Isso dificulta ataques de força bruta;
- Desligue a configuração padrão de DHCP e atribua endereços IP estáticos;
- Utilize nomes individuais para o SSID e ESSID;
- Desative os beacons no roteador;
- Se disponível no hardware, defina listas de controle de acesso (ACLs) para controlar o acesso à Internet pelo endereço MAC das placas de rede;
- Mude a localização e a potência de transmissão do roteador sem fio de modo que a recepção seja boa para sua rede, mas não cubra a propriedade alheia;
- No caso de dispositivos MIMO com três antenas, é recomendável afastar uma antena da outra para melhorar a capacidade de transmissão e recepção;
- Sempre utilize o meio cabeado para configurar o roteador. Isso dificulta a interceptação dessa comunicação.

servidor Radius. A versão *Enterprise* do WPA2 cobre o padrão 902.11i por completo e por isso suporta a autenticação por servidor Radius.

Segurança com WPA2

Atualmente, redes sem fio baseadas no padrão WPA2 são consideradas as mais seguras. Ataques de dicionário contra a chave pré-compartilhada são a maior ameaça – supondo que o agressor possua tempo e capacidade computacional suficientes. Teoricamente, as chaves *broadcast* e *multicast* representam outra vulnerabilidade. Todos os pontos na rede precisam conhecê-las, e um agressor que descobrir uma das chaves poderá ao menos interceptar a troca de chaves entre o ponto de acesso e a estação.

Graças ao modelo de segurança do padrão WPA2, redes sem fio modernas contam com segurança eficaz. O maior fator de insegurança é o próprio usuário. Hoje, quando um agressor persistente consegue acesso a uma infraestrutura Wi-Fi moderna e dedica energia suficiente para entrar na rede e causar danos, a causa mais provável é a configuração descuidada do ponto de acesso. Portanto, é necessário verificar cuidadosamente cada configuração do roteador sem fio (**figura 2**).

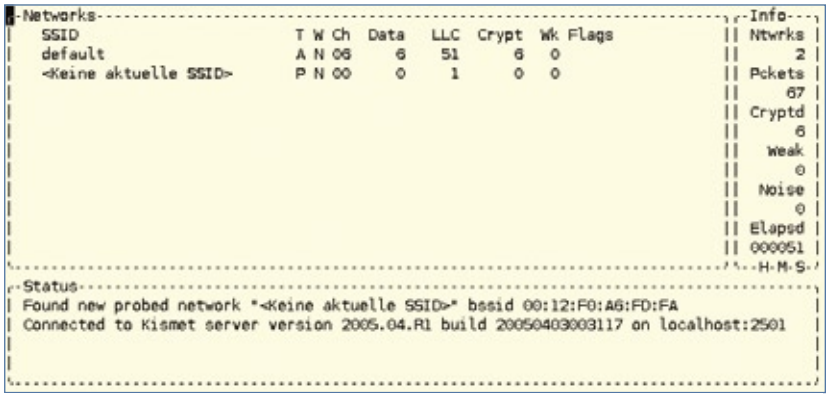


Figura 2 O interceptador WiFi Kismet pode verificar as vulnerabilidades de uma rede e descobrir informações sobre os protocolos utilizados.

Para reduzir ainda mais o risco residual, é possível acrescentar à rede alguma proteção baseada em software. Se você utilizar um túnel, como uma VPN com IPSec, por exemplo, a barreira ficará difícil até mesmo para invasores experientes. Como de costume, o Linux, com suas diversas ferramentas de segurança, é uma escolha perfeita para eliminar o risco residual. ■

TUTORIAL

O Adriane permite o uso do desktop por deficientes visuais

Desktop auditivo

O sistema desktop auditivo Adriane fornece um Linux para usuários com deficiência visual.

por Klaus Knopper

O Linux tem vários belos desktops, incluindo cubos giratórios, janelas moles e inflamáveis e gestos do mouse para funções semiautomáticas terrivelmente inteligentes. Infelizmente, desde o início da computação orientada ao mouse, o desktop gráfico foi projetado para usuários que trabalham dentro de um contexto visual.

Considere, por exemplo, um desktop KDE bem configurado com ícones cuidadosamente organizados. Se você desligar o monitor, você ainda conseguirá abrir o cliente de email e ler suas mensagens? A maioria dos usuários sequer conseguiria encontrar o ícone do programa sem informações visuais, mas é assim que um desktop gráfico se apresenta para deficientes visuais.

Alguns fabricantes de software vendem "acessórios de acessibilidade" para desktops gráficos, que criam a impressão de que todo o problema é subitamente resolvido com a compra de mais softwares – e mais caros. Acrescentar retornos auditivos aos menus, elementos interativos e teclados de fato possibilitam a cegos operarem um programa que não conseguiam usar antes, mas como as interfaces interativas dos programas ainda são projetadas para um panorama "visual", ainda é difícil e cansativo procurar botões ou funções específicos.

Já existem muitas ferramentas e extensões de código aberto necessárias para trabalhar com deficiências visuais (como cegueira ou baixa acuidade visual), motoras e mentais – e elas são parte de quase todos os repositórios de distribuições GNU/Linux. O projeto *Adriane* [1] reúne essas tecnologias num sistema auditivo único para deficientes visuais.

Acessibilidade vs. equalização

Muitos usuários cegos nem têm a opção de escolher qual software usar no ambiente de trabalho. Essa decisão muitas vezes é tomada por um empregador que precisa se adequar legalmente às exigências de acessibilidade do local de trabalho. A decisão mais comum é simplesmente que "todos na empresa precisam usar o mesmo software" ou alguma extensão disso. Essa escolha costuma ser feita sem considerar a verdadeira opinião ou as necessidades dos funcionários deficientes visuais. Um sistema desktop proprietário para cegos custa muitos milhares de reais, e a pessoa que instala o sistema parte do princípio de que qualquer coisa tão cara *precisa ser boa*. Mas será que é verdade?

Empregadores que tomam decisões apressadas sobre softwares sem barreiras não apenas criam espaços de trabalho ineficientes e frustrantes como também perdem muita produtividade por subestimarem as capacidades dos funcionários. Um cego que use hardware e software apropriados consegue trabalhar pelo menos tão rápido quanto (e às vezes até mais rápido do que) alguém sem deficiência visual. Ele consegue ler, entender e lembrar-se de uma página inteira de texto em apenas alguns segundos – enquanto usuários comuns ainda estão definido quais partes da página são anúncios. Ele consegue encontrar ofertas perfeitas e vencer leilões no eBay antes mesmo de você encontrar o caminho de cliques para fazer uma oferta – isto é, se o ambiente de trabalho for projetado para as capacidades do usuário em vez de uma "ponte de acessibilidade" padrão para layouts orientados à visão.

Adriane

A linha de comando é a interface mais eficaz para trabalhar com computadores, pois oferece uma forma direta de introduzir comandos que fazem

o computador realizar exatamente o que se deseja. Uma interface de texto direta se concentra no conteúdo, não no layout ou intuição visual.

Até nos antigos sistemas DOS era possível exibir o texto da tela para periféricos conectados por meio de portas paralelas e seriais, como impressoras, sintetizadores de voz por hardware e dispositivos Braille. Infelizmente, os desktops gráficos complicaram um tanto a vida dos usuários deficientes.

Enquanto a equipe do Adriane pesquisava o assunto, trabalhando de perto com usuários cegos (até mesmo iniciantes em computação) – e também desenvolvedores cegos – chegamos à conclusão de que as interfaces dos softwares deveriam adaptar-se às capacidades do usuário, em vez de forçá-lo a se adaptar a uma interface que jamais teve como objetivo oferecer suporte a cegos.

O nome ADRIANE (*Audio Desktop Reference Implementation And Networking Environment* – Ambiente de Rede e Referência para Implementação de Desktop Auditivo) descreve uma interface de usuário que não requer sequer um monitor ou saída visual, mas ainda oferece uma interface de usuário passo-a-passo linear e fácil de usar, organizada em menus de acordo com as preferências do usuário. Em vez de reinventar a roda, integramos técnicas já existentes no GNU/Linux, tais como leitores de tela, sintetização de voz, drivers Braille, navegação pelo teclado e programas que podem ser inteiramente controlados em modo não gráfico (**figura 1**).

Inteiramente por acaso, Adriane também é o nome da minha esposa, que foi a primeira *beta tester* e co-desenvolvedora do projeto pela perspectiva do usuário e que, naturalmente, tinha grande ceticismo quanto à usabilidade para iniciantes, dadas as experiências passadas desconfortáveis com sistemas para

Figura 1 O Adriane suporta um ambiente desktop completo para deficientes visuais.

cegos. Quando ela aprendeu Braille, era necessária uma pessoa com visão por perto para instalar e trabalhar com dispositivos Braille, e usar a Internet com as ferramentas disponíveis até então estava fora de questão. O máximo que um usuário cego conseguia fazer era processar texto com atalhos complicados.

O menu do Adriane ordena as tarefas mais comuns numa estrutura plana, sem comandos a serem lembrados ou digitados. A primeira linha diz "Enter para ajuda, seta para baixo o próximo menu", que é um bom ponto de partida quando se interage com uma interface não visual pela primeira vez. A pedido especial de usuários e programadores cegos mais experientes, depois acrescentamos um item *Shell* ao primeiro menu. Em geral, o menu é fácil de estender e funciona em modo texto assim como em modo gráfico, graças ao *dialog* e ao *Xdialog*, como mostra a **figura 2**.

Os utilitários que acompanham o Adriane, incluindo ferramentas como *Elinks*, *Mutt*, *Irssi*, *MPlayer* e *Sane/OCRopus*, cobrem as atividades mais populares de usuários finais e rodam com facilidade em consoles de texto juntamente com o leitor de

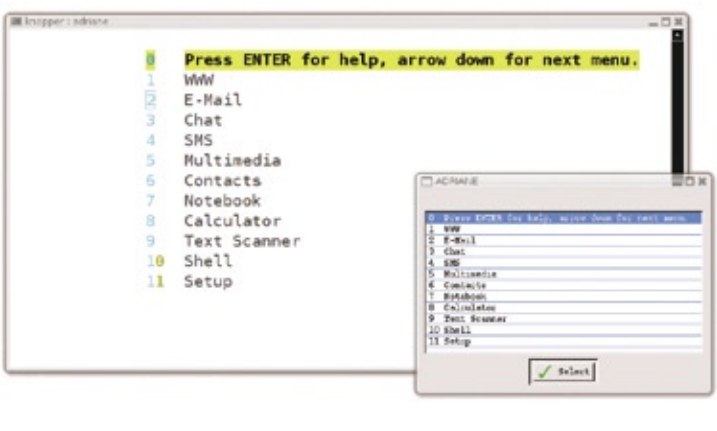


Figura 2 Os menus simples do Adriane são fáceis de navegar e personalizar.

tela. Uma especialidade que ainda não apareceu em sistemas proprietários sem barreiras é a opção de ler e enviar mensagens de texto SMS com um celular – sem qualquer software no próprio telefone. Como se pode imaginar, é impossível ler ou responder uma mensagem de texto tendo apenas a pequena tela do aparelho como fonte de informação quando se é cego. Com o GSM, o usuário do Adriane consegue baixar mensagens SMS para o computador e respondê-las com uso de um editor e um teclado normal, em vez das teclinhas do telefone.

As seções a seguir descrevem alguns dos programas e componentes usados no Adriane. O foco são os recursos para usuários cegos e com visão prejudicada. Não serão cobertas outras ferramentas para suportar deficiências motoras ou mentais, como reconhecimento de fala, teclas *sticky* etc.

Dispositivos Braille e leitores de tela

Fazer o computador pronunciar uma linha de texto e exibir o texto num dispositivo Braille são as formas mais comuns para cegos aprenderem o que está escrito na tela do computador. Um dispositivo Braille, comumente chamando simplesmente de "linha" devido a suas dimensões verticais restritas, é uma tela tátil que consiste em seis ou oito pontos por letra, que podem ser lidos pelo toque por quem sabe Braille. A **figura 3** mostra uma parte da tradução do Braille para o alfabeto brasileiro. Cada idioma usa uma tabela diferente para a tradução e, como não há símbolos especiais para números, as letras *a* a *j* são usadas para representar os algarismos de um a zero, às vezes com um "símbolo de número" antes para esclarecer que tratam-se de dígitos.

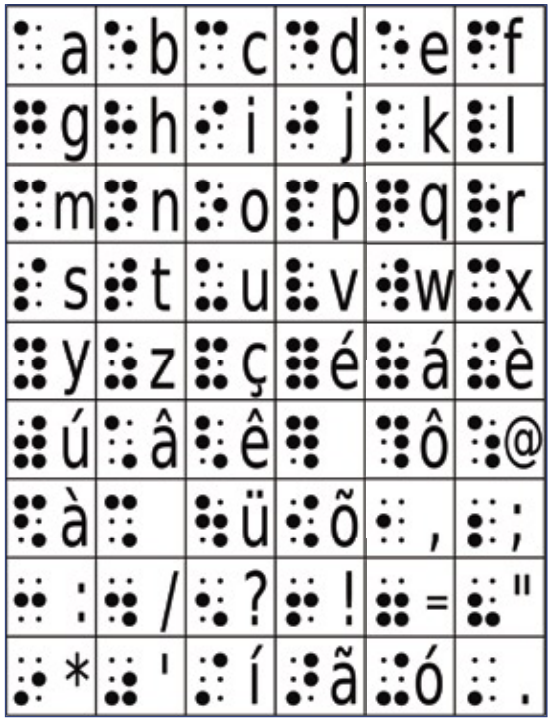


Figura 3 Alfabeto Braille português.

Juntamente com outras opções especializadas, há duas opções principais de leitores de telas e drivers para dispositivos Braille e de fala no Linux: *brltty* [2] e *SUSE Blinux* (SBL) [3].

O Brltty provavelmente é o driver de interface Braille mais conhecido, enquanto a força do SBL está na extensão do suporte a Braille e *text-to-speech* para aplicativos individuais, o que possibilita a personalização individual de quais partes da tela e do texto serão exibidas. O SBL também permite que os usuários naveguem na tela com teclas de dispositivos Braille, assim como apenas com o teclado, que é o motivo pelo qual o SBL é o leitor primário de tela no sistema Adriane. O Adriane emprega a tecla raramente usada `Caps` para navegação e funções do teclado no SBL, como mostra a **tabela 1**.

Tabela 1: Navegação com a tecla Caps Lock

Caps Lock + para cima	Lê a linha anterior
Caps Lock + para baixo	Lê a próxima linha
Caps Lock + page up	Lê desde o início da página
Caps Lock + page down	Lê até o fim da página
Caps Lock + f/s	Fala mais rápido (*f*) ou devagar (*s*)
Caps Lock + l/p	Altera o idioma

Usando essas poucas combinações de teclas é possível descobrir, linha por linha, o que o computador está exibindo na tela, mesmo com o monitor desligado. Embora não seja possível visualizar um panorama dos programas, menus e botões todos de uma só vez, o usuário ainda consegue acessar qualquer linha de informação e pular para partes específicas de uma página usando as teclas.

O *Orca* [4] é um leitor de tela escrito em *Python* para aplicativos gráficos e usa a biblioteca GTK2 e a Interface de Provedor de Serviço de Tecnologia Assistiva (AT-SPI, na abreviação em inglês). O Orca envia rótulos de texto de menus, botões e áreas de texto misto (como o painel principal do navegador web) para um dispositivo Braille e sintetizador de voz. O Orca ainda possui um recurso de magnificação, porém este não funcionou de forma confiável em nossos testes.

O Orca possibilita o trabalho com o *OpenOffice.org* 2.3 e versões posteriores com áudio e Braille, contanto que o usuário conheça todas as

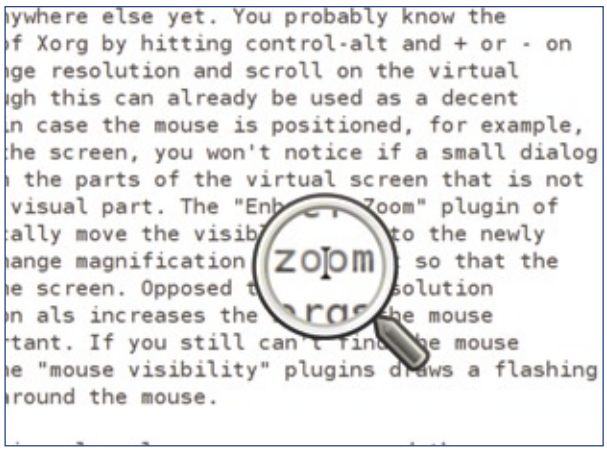


Figura 4 Há um plugin para o *Compiz* que permite a magnificação do texto ao redor do cursor do mouse.

abreviações de teclado necessárias para ativar funções normalmente selecionadas com o mouse. O Orca não apenas lê o texto "visual" aparente, mas também oferece dicas e metainformações como família e renderização da fonte, tipos de elementos de formulário e assim por diante. Apesar de estar sendo desenvolvido primariamente para o ambiente Gnome, ele funciona bem com todos os gerenciadores de janela contanto que o aplicativo individual suporte o AT-SPI. Estes incluem Firefox, OpenOffice.org, Pidgin e até (parcialmente) Gimp. Se o Orca não for iniciado a partir do Gnome, é preciso definir algumas variáveis de sistema para ativá-lo com aplicativos GTK2:

```
export SAL_USE_VCLPLUGIN="gtk"
export GTK_MODULES="gail:atk-
➥bridge"
orca &
soffice documento.odt
```

O Orca tem plugins tanto para o brltty quanto para o SBL, então, o mesmo driver Braille que funciona no modo texto também pode ser usado no modo gráfico.

Apesar de o Orca possibilitar que cegos utilizem programas orientados ao mouse, uma interface gráfica com vários botões ou menus numa única janela não é ótima ou eficiente para uso não gráfico. Trabalhar com interfaces gráficas é ainda mais lento e complicado para usuários com deficiência visual do que para aqueles com visão. O verdadeiro desastre ocorre quando o programa é minimizado ou sua janela perde foco por causa de outro aplicativo. Com isso, a janela se torna inacessível pelo leitor de tela até receber novamente o foco, e para o usuário fica ainda mais "invisível". A menos que se saiba como restaurar janelas minimizadas (**[Alt]+[Tab]** em alguns casos), não fica claro para o usuário sem visão se o programa simplesmente perdeu o foco e desapareceu ou se o próprio leitor de tela travou por erro de software. Portanto, a interface de escolha para deficientes visuais iniciantes na computação ainda é o console de texto, que nunca perde foco e sempre fornece um modo "tela cheia" para cada programa.

O leitor de tela envia o texto para um dispositivo Braille ou um sintetizador de voz, mas a entrada ainda precisa ser digitada num teclado comum. Embora uma pessoa cega não consiga ver o que está escrito em cada tecla, pode-se perceber que todo teclado – até o seu – possui pequenas marcações táteis nas teclas **[F]** e **[J]** que fornecem certa orientação para deficientes visuais. Nos telefones, geralmente é o número 5 que possui a marca. Ao preparar um teclado para deficientes visuais, geralmente os técnicos usam um ferro de solda para acrescentar outras marcações com pontos táteis para uma melhor orientação.

Texto para voz

Para permitir que o computador leia o texto exibido na tela, é preciso um sintetizador de texto para voz (ou *text-to-speech*, como também é conhecido). A teoria linguística por trás de como gerar texto falado de alta qualidade a partir de algo escri-

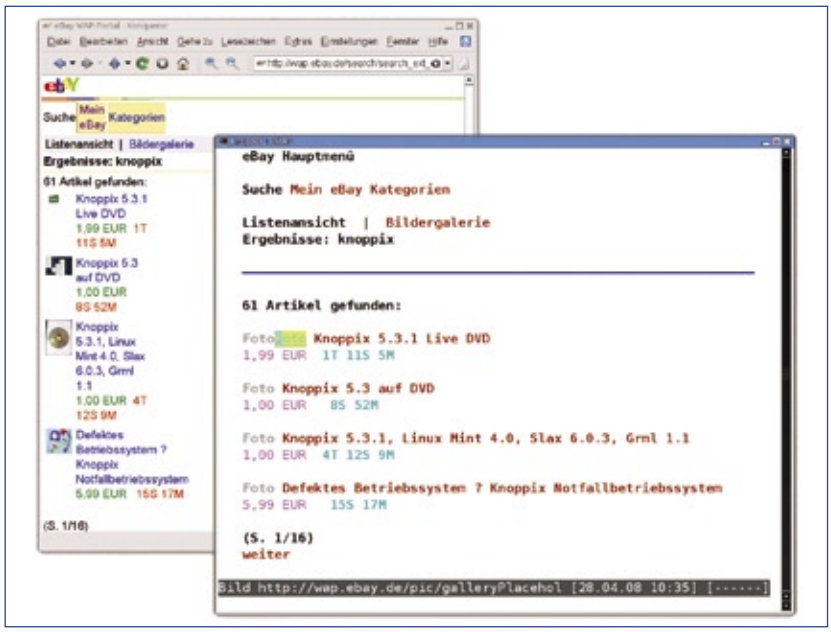


Figura 5 Portal WAP do eBay no Konqueror e no Elinks.

to preenche, por si só, alguns livros. Algumas regras fonéticas informam como o texto escrito (combinações de letras e sílabas) são pronunciados corretamente, e há programas que rastreiam vários milhares de exceções frequentes. Apenas concatenar sons leva a resultados incompreensíveis; portanto, ferramentas de boa qualidade escolhem partes maiores (*seleção de unidades*) ou menores (*difonos* ou *síntese de meias-sílabas*) de sons pré-gravados a partir de um enorme banco de dados. Gravar e rotular textos pré-gravados exige muito trabalho; infelizmente, o resultado raramente é coberto por uma licença que permita distribuição irrestrita.

Figura 6 O Adriane leva a computação sem barreiras ao barato netbook EeePC.

O *Festival* [5] é um sofisticado analisador e sintetizador de fala, mas criar um banco de dados e um conjunto de regras de fala para ele não é fácil, pois ele usa uma sintaxe semelhante à de *lisp* e requer um banco de dados de difonos com aproximadamente 3 mil trechos de áudio, cortados e estendidos por pontos de entonação. Há somente algumas poucas vozes gratuitas gravadas para o Festival no momento. O *Mbrola*, um sintetizador de fala "grátis para uso não comercial", conseguiu muitas contribuições em seu banco de dados de fala, mas sua licença não permite a distribuição livre para nenhum propósito e ele não é compatível com licenças de código aberto. Há alguns sistemas proprietários de text-to-speech disponíveis, mas o foco deste artigo são as opções livres e de código aberto.

A melhor escolha para o Adriane é o *eSpeak* [6], que consome poucos recursos da CPU, fala mais de 30 idiomas e é facilmente extensível. O eSpeak possui uma abordagem inteiramente sintética, sem vozes gravadas que soem reais, então ele parece "robótico"; por outro lado, é livre de alegações proprietárias.

Para coordenar os recursos de text-to-speech, o *Speech Dispatcher* [7] já faz parte de vários programas de acessibilidade. Ele é capaz de interromper a saída de um texto longo com uma mensagem mais curta com maior prioridade, depois retornar ao texto inicial em seguida, opcionalmente com vozes diferentes ou sons entre os discursos. Tanto o SBL quanto o Orca (em suas versões atuais) empregam o Speech Dispatcher para fornecer recursos de voz a diferentes programas.

Todos os *back-ends* de acessibilidade, como o Speech Dispatcher, o *kbdsniffd* (driver de navegação pelo teclado) e o SBL (leitor de tela) são iniciados pelo *adriane-screenreader* na ordem correta.

Magnificação e cor

Apesar de você talvez imaginar que gerenciadores de janela 3D como o *Compiz Fusion* não sejam adequados para deficientes visuais, na verdade o Compiz contém algumas extensões úteis para usuários com baixa visão. A lente de aumento que acompanha o foco é uma ferramenta prática que não vi em nenhum outro projeto além do Compiz. Basta pressionar **[Ctrl]+[Alt]+[+]** para aumentar a tela ou **[Ctrl]+[Alt]+[-]** para diminuí-la. Apesar de isso já poder ser usado como uma lente de aumento decente, se o mouse for posicionado, por exemplo, no canto superior esquerdo da tela, não será possível perceber caso um pequeno diálogo se abra em outra parte da tela que não esteja magnificada. O plugin *Enhanced Zoom* do Compiz Fusion move a janela visível para a nova janela com foco e altera o aumento para que se adapte à nova janela. Diferentemente do alternador de resolução do Xorg, essa ferramenta também aumenta o tamanho do ponteiro do mouse. Se ainda não for possível encontrar a localização do cursor na tela, o plugin *mouse visibility* exibe um círculo de fogo bastante chamativo ao seu redor.

Um segundo plugin de lente de aumento apenas aumenta uma área em torno do ponteiro do mouse, o que talvez seja preferível caso se deseje visualizar a tela inteira com apenas os detalhes aumentados (**figura 4**).

No caso de certos tipos de daltonismo, é possível trocar certas cores por outras, com escolhas a partir de várias tabelas, ou simplesmente inverter toda a tabela de cores (que também é um recurso prático para apresentações com contraste insuficiente).

Navegador acessível

O *Elinks*, um *fork* "experimental" do navegador *Links*, oferece certos recursos práticos para interfaces de texto [8]. O Elinks suporta CSS e JavaScript, o que permite o uso de algumas páginas que se recusam a funcionar em navegadores sem suporte a JavaScript. O SBL é configurado por padrão para que o Elinks leia somente textos marcados com tags `<a href>`, e sem texto puro entre

elas. Isso permite a navegação rápida apenas conferindo e seguindo links a princípio; ao final, chegando-se à página desejada, o usuário pode obter uma leitura da página inteira.

Botões gráficos e símbolos não textuais simplesmente não podem ser exibidos como uma "letra", portanto criam barreiras para usuários deficientes visuais. Os famosos "captchas", imagens com texto difícil de reconhecer para fins de validação, são um exemplo de criação involuntária de barreiras artificiais. Embora o Elinks não tenha grande chance de contornar essas barreiras, ele oferece certas ferramentas para acesso a elementos "invisíveis" de formulários, tais como um formulário de envio sem botão de "enviar".

A **figura 5** mostra uma comparação entre o *Konqueror* e o Elinks, ambos usando a versão WAP menos carregada do eBay [9].

Imagens e elementos gráficos numa página web ainda podem ser "vistos" num ambiente somente-texto caso haja uma descrição textual por meio de metainformações (*textarea*, botão *enviar* etc.) ou títulos e rótulos, para os quais o parâmetro `alt` no interior das tafs `<img>` é usado. Se não houver nenhuma descrição, as imagens ficam "invisíveis" a menos que algum tipo de software de reconhecimento de escrita (ou OCR, como costumam ser chamados) descubra textos desenhados na figura.

O Elinks pode invocar um visualizador de imagens pelo *framebuffer* como o *fbi* para exibir imagens no console de texto. Da mesma forma, vídeos podem ser reproduzidos no framebuffer com o MPlayer, então o suporte completo a multimídia não depende totalmente do Xorg.

Reconhecimento de texto

Além disso, pode-se usar softwares de OCR para converter textos escritos em algo legível digitalmente. Durante certo tempo, o *GOCR* foi o único software livre capaz de converter imagens escaneadas para texto.

Agora o Google iniciou um novo projeto de código aberto chamado *OCRopus* [10], que consiste em uma análise de layout em áreas ou colunas consecutivas, assim como um mecanismo OCR baseado no *Tesseract* que faz reconhecimento e ajustes no texto baseados em probabilidade. A versão de desenvolvimento do OCRopus já produz resultados muito bons na maioria dos casos, então é possível usar a combinação do OCRopus com a ferramenta de escâner *Sane* [11] para escanear e ler cartas e livros.

Obtendo o Adriane

O sistema Adriane está disponível no Live CD ou DVD do Knoppix desde a versão 5.3 por meio da opção de inicialização *adriane*. Também é possível remasterizar o CD ou DVD para usar o Adriane como opção padrão alterando-se o arquivo `boot/isolinux/isolinux.cfg`.

Com o SBL 3.2.1, seu autor Marco Skambraks acrescentou um outro *daemon* de teclado para navegação pela tela, então o patch do kernel para o *keyboard sniffer* deixa de ser necessário; o módulo `uinput` já é suficiente. Portanto, o Adriane agora também já deve ser instalável na forma de acessório do Debian, simplesmente instalando os pacotes conforme descrito na página do Adriane [1].

Depois de corrigir a configuração do Xorg pré-instalada no Asus EeePC, que não veio configurado com o *composite* ativado, experimentamos (**figura 6**) e criamos um cartão flash SD inicializável com o Adriane, Orca e Compiz Fusion.

Juntamente com um teclado USB marcado e (para usuários com baixa visão) uma tela TFT suficientemente grande, incluir o Adriane num pen drive ou cartão de memória inicializáveis consiste em uma forma muito barata e acessível para se criar um ambiente de trabalho portátil.

Conclusão

Apesar da comunidade Linux oferecer diversos ótimos auxílios e ferramentas para suporte a usuários com deficiências, simplesmente fazer todos esses programas funcionarem em harmonia geralmente exige muito trabalho, a menos que venham pré-instalados e pré-configurados. O Adriane une todas as ferramentas necessárias num único desktop auditivo fácil de usar. ■

Mais informações

[1] Projeto Adriane: http://knopper.net/knoppix-adriane/index-en.html

[2] brltty: http://mielke.cc/brltty/

[3] SUSE Blinux: http://en.opensuse.org/SUSE_Blinux

[4] Orca: http://live.gnome.org/Orca

[5] Festival: http://www.cstr.ed.ac.uk/projects/festival/

[6] eSpeak: http://espeak.sourceforge.net/

[7] Speech Dispatcher: http://www.freebsoft.org/speechd

[8] Elinks: http://elinks.or.cz/

[9] Versão WAP do eBay: http://wap.ebay.com

[10] OCRopus: http://code.google.com/p/ocropus/

[11] Sane: http://www.sane-project.org/

Minix 3 e a experiência do microkernel

Kernel inteligente

O Minix costuma ser considerado o antecessor do Linux, mas seus kernels são bem diferentes. O novo Minix 3, agora com uma licença no estilo BSD, está em busca de usuários.

por Rüdiger Weis

O Linux possui uma relação longa e tempestuosa [1] com um outro sistema tipo Unix conhecido como Minix [2]. O famoso autor, professor e cientista da computação Andrew S. Tanenbaum lançou a primeira versão do Minix em 1987 como uma ferramenta para ensinar sistemas operacionais a seus alunos. Rapidamente, esse pequeno e bem documentado sistema ganhou popularidade com os entusiastas de sistemas operacionais. Numa mensagem enviada ao grupo de discussão do Minix, o jovem universitário finlandês Linus Torvalds anunciou em 1991 seu próprio sistema experimental chamado Linux. Portanto, muitos dos primeiros colaboradores do Linux vieram da comunidade do Minix.

Porém, Tanenbaum e Torvalds desde cedo começaram a se digladiar quanto a aspectos de projeto. Por um lado, Tanenbaum sempre favoreceu a arquitetura de microkernel, um recurso particular do Minix nos dias atuais (veja o **quadro 1**). Linus, por outro lado, criou o Linux como um kernel monolítico, com sistemas de arquivos, drivers e outros componentes incorporados ao kernel. Numa mensagem famosa enviada ao grupo do Minix, o criador Tanenbaum se referiu ao Linux como "um grande passo de volta ao anos 1970", e a confiante resposta do jovem Torvalds ao professor é uma das primeiras evidências da objetividade que agora já sabemos ser sua característica. Ainda assim, Linus reconheceu a importância do trabalho de Tanenbaum na formação de suas próprias ideias. Em sua autobiografia *Just for Fun* (infelizmente ainda não traduzida para português), Linus se refere ao livro de Tanenbaum *Operating Systems: Design and Implementation* como aquele que mudou sua vida.

O debate sobre microkernels versus kernels monolíticos continua até os dias atuais, e, da mesma forma que o Linux não desapareceu, o Minix também permanece presente. A versão 3 do sistema operacional de Tanenbaum foi projetada com o objetivo de criar um sistema mais seguro e confiável do que outros sistemas POSIX comparáveis, e a licença de código aberto no estilo BSD o torna um forte candidato à produção, assim como ao uso educacional.

O Minix está conseguindo atrair até a atenção de importantes patrocinadores. A União Europeia está patrocinando o projeto com vários milhões de euros, e o Google possui vários projetos do Minix em seu programa "Summer of Code".

O Minix 3 é compatível com processadores *x86* de 32 bits e também com várias máquinas virtuais, incluindo *Qemu*, *Xen* e *VMware*. O sistema operacional inclui um sistema *X Window*, vários editores (*Emacs* e *Vi* incluídos), shells (incluindo *Bash* e *Zsh*), *GCC*, linguagens de script como *Python* e *Perl* e ferramentas de rede como SSH. Sua arquitetura resistente a falhas e com baixo consumo de recursos pode tornar o Minix um bom candidato para sistemas embarcados, e sua estabilidade superior lhe confere um papel promissor como firewall.

Inseguro desde o projeto

Os problemas de segurança enfrentados pela safra atual de sistemas operacionais, incluindo o Windows e o Linux, são resultado de erros de projeto. Os erros foram herdados, em sua maior parte, de seus ancestrais da década de 1960. A maioria desses problemas pode ser atribuída ao fato de que desenvolvedores não são perfeitos. Humanos cometem erros. Obviamente, seria bom reduzir os números e abrandar seus efeitos; entretanto, os arquitetos frequentemente se mostram dispostos a favorecer a velocidade em detrimento da segurança e da eficiência do projeto. Tanenbaum chama esse fenômeno de "Pacto Faustiano".

Além das questões relacionadas ao tamanho, projetos monolíticos também são suscetíveis a problemas estruturais: qualquer erro é capaz de pôr em perigo o sistema inteiro. Um erro de projeto fundamental é que os sistemas opera-

Quadro 1: Por que os computadores não funcionam sem parar?

Os usuários de computadores estão mudando. Há dez anos, a maioria dos usuários de computadores era pessoas ou profissionais jovens com amplo conhecimento técnico. Quando algo saía errado – o que ocorria com frequência – eles sabiam consertá-las. A maioria deles consegue consertar computadores tão bem quanto um nerd de computador padrão sabe consertar seu carro. O que eles querem mais do que qualquer outra coisa é que o computador funcione o tempo todo, sem interrupções ou falhas.

Muitos usuários comparam automaticamente seus computadores a suas televisões. Ambos estão repletos de componentes eletrônicos mágicos e possuem telas grandes. A maioria dos usuários tem um modelo implícito de uma televisão: (1) você compra a TV; (2) você a liga na tomada; (3) ela funciona perfeitamente sem qualquer falha durante os próximos dez anos. Eles esperam isso do computador e, quando não é o que obtêm, ficam frustrados. Quando os especialistas em computadores lhes dizem: "Se Deus quisesse que os computadores funcionassem o tempo todo, Ele não teria inventado o botão de RESET", eles não se convencem.

Por falta de uma melhor definição de disponibilidade, adotemos a seguinte: um dispositivo é dito disponível (isto é, podemos dispor dele) se 99% dos usuários jamais experimenta qualquer falha durante todo o período em que o possuem. Por essa definição, virtualmente nenhum computador é disponível, enquanto a maioria das TVs, iPods, câmeras digitais etc. são. Usuários técnicos de computador estão dispostos a perdoar um computador que trave uma ou duas vezes por ano; usuários comuns, não.

Usuários domésticos não são os únicos incomodados com a baixa disponibilidade dos computadores. Até mesmo em ambientes altamente técnicos, a baixa disponibilidade dos computadores é um problema. Empresas como Google e Amazon, com centenas de milhares de servidores, experimentam várias falhas todo dia. Elas aprenderam a conviver com isso, mas prefeririam sistemas que simplesmente funcionassem sem parar. Infelizmente, os softwares atuais falham nesse aspecto.

O problema básico é que softwares contêm bugs, e quanto mais software, mais bugs. Vários estudos já mostraram que o número de bugs por mil linhas de código (KLoC) varia de um a dez em grandes sistemas de produção. Um software muito bem escrito talvez tenha dois bugs por KLoC ao longo do tempo, mas não menos. Um sistema operacional com, digamos, 4 milhões de linhas de código, portanto, deve ter pelo menos 8 mil bugs. Nem todos são fatais, mas alguns serão. Um estudo da Universidade Stanford mostrou que drivers de dispositivos – que compõem até 70% da base de código de um sistema operacional típico – possuem taxas de bugs 3x a 7x mais altas que o resto do sistema. Drivers de dispositivos têm taxas mais altas porque (1) são mais complicados e (2) são menos inspecionados. Enquanto muitas pessoas estudam o escalonador, poucas verificam os drivers de impressoras.

A solução: kernels menores

A solução para esse problema é retirar código do kernel, onde o dano pode ser máximo, e colocá-lo em processos do espaço do usuário, nos quais bugs não conseguem causar falhas de sistema. É assim que o Minix 3 é projetado. O sistema Minix atual é o (segundo) sucessor do Minix original, que foi lançado originalmente em 1987 como sistema operacional educativo, mas desde então foi radicalmente revisado para se tornar um sistema altamente disponível e autorrecuperável. Segue uma breve descrição da arquitetura do Minix; há mais informações em www.minix3.org.

O Minix 3 é projetado para rodar o mínimo de código possível no modo do kernel, onde bugs podem facilmente ser fatais. Em vez de 3-4 milhões de linhas de código no kernel, o Minix 3 tem aproximadamente 5.000 linhas de código no kernel. Às vezes, kernels desse tamanho são chamados de microkernels. Eles lidam com gerenciamento de processos no baixo nível, escalonamento, interrupções e o relógio, além de fornecerem alguns serviços de baixo nível para componentes do espaço do usuário.

A maior parte do sistema operacional roda como uma coleção de drivers de dispositivos e servidores, cada um rodando como processo comum do espaço do usuário com privilégios restritos. Nenhum desses drives e servidores roda como superusuário ou equivalente. Eles não conseguem nem acessar dispositivos de I/O ou o hardware MMU diretamente. Precisam usar serviços do kernel para ler e escrever no hardware. A camada de processos rodando diretamente no modo de usuário acima do kernel consiste em drivers de dispositivos, com o driver de disco, o de Ethernet e de todos os outros rodando como processos separados protegidos pelo hardware MMU, para não conseguirem executar qualquer instrução privilegiada e nem lerem ou escreverem em locais de memória além dos seus próprios.

Acima da camada de drivers vem a de servidores, com um servidor de arquivos, um servidor de processos e outros. Os servidores fazem uso dos drivers assim como de serviço do kernel. Por exemplo, para ler um arquivo, um processo do usuário envia uma mensagem ao servidor de arquivos, que então envia uma mensagem para o driver de disco para buscar os blocos necessários. Quando o sistema de arquivos os tem em seu cache, ele chama o kernel para movê-los para o espaço de endereços do usuário.

Além desses servidores, há um outro servidor chamado "servidor de reincarnação". Ele é o pai de todos os processos de drivers e servidores e monitora seu comportamento. Se ele descobrir um processo que não esteja respondendo a pings, ele inicia uma nova cópia a partir do disco (exceto pelo driver do disco, que fica oculto na RAM). O sistema foi projetado para que muitos (mas não todos) os drivers e servidores críticos sejam automaticamente substituídos enquanto o sistema funciona, sem perturbar os processos de usuário em execução e sem nem notificar o usuário. Dessa forma, o sistema é autorrecuperável.

Quadro 1: Por que os computadores não funcionam sem parar? (continuação)

Para testar se essas ideias funcionam na prática, conduzimos os seguintes experimentos: iniciamos um processo de injeção de falhas que sobrescreveu 100 instruções de máquina no binário do driver Ethernet em execução para ver o que ocorreria caso um deles fosse executado. Se nada acontecesse em poucos segundos, outras 100 eram injetadas e assim por diante. No total, injetamos 800.000 falhas em cada um dos três diferentes drivers Ethernet e causamos 18.000 travamentos do driver. Em todos os casos, o driver foi automaticamente substituído pelo servidor de reencarnação. Apesar de injetar 2,4 milhões de falhas no sistema, o servidor não parou uma vez sequer. Nem é preciso dizer que se ocorrer um erro fatal num driver do Windows ou do Linux rodando no kernel, todo o sistema operacional travará imediatamente.

Existe alguma desvantagem nessa técnica? Sim. Há uma redução de desempenho. Não a medimos extensivamente, mas o grupo de pesquisa em Karlsruhe, Alemanha, que desenvolveu seu próprio microkernel (o *L4*) e depois rodou o Linux como um de seus processos de usuário, conseguiu uma perda de desempenho de apenas 5%. Acreditamos que se dedicarmos um pouco de atenção a esse típico, também conseguiremos reduzir a perda para a faixa entre 5 e 10%. Desempenho não é uma prioridade para nós, já que a maioria dos usuários que leem email ou navegam pelo Facebook não são limitados pelo desempenho da CPU. O que eles querem, no entanto, é um sistema que simplesmente funcione o tempo todo.

Se microkernels são tão disponíveis, por que ninguém os usa?

Na verdade, usam, sim. Provavelmente você roda vários deles. Seu telefone celular, por exemplo, é um computador pequeno, mas comum em todos os outros aspectos, e há uma boa chance de ele rodar o L4 ou o Symbian, outro microkernel. O roteador de alta performance da Cisco também usa um microkernel. Nos mercados militar e aeroespacial, onde disponibilidade é fundamental, o Green Hills Integrity, outro microkernel, é amplamente usado. O PikeOS e o QNX também são microkernels amplamente usados em sistemas industriais e embarcados. Em outras palavras, quando é realmente importante que o sistema "simplesmente funcione o tempo todo", as pessoas usam microkernels. Para mais informações sobre esse tópico, veja www.cs.vu.nl/~ast/reliable-os/.

Concluindo, é nossa crença, baseada em várias conversas com usuários não técnicos, que o que eles mais desejam é um sistema que funcione perfeitamente todo o tempo. Eles têm uma baixa tolerância a sistemas pouco confiáveis, mas atualmente não têm escolha. Acreditamos que sistemas baseados em microkernels podem nos levar a sistemas mais disponíveis.

cionais atuais não seguem o princípio da menor autoridade (POLA, na sigla em inglês). Em resumo, o POLA dita que os desenvolvedores devem distribuir os sistemas por diversos módulos para que um erro em um dos módulos não comprometa a segurança e a estabilidade de outros módulos. Eles também devem certificar-se de que cada módulo tenha apenas os direitos de que precisa para realizar suas respectivas tarefas.

Quadro 2: A questão da extensão

Muitos desenvolvedores e usuários discordam da doutrina de Tanenbaum, mantida há mais de uma década, de ser muito cauteloso quanto à introdução de extensões no kernel. O conceito de Tanenbaum a respeito da complexidade aceitável para o sistema operacional é um sistema que possa ser ensinado num único semestre. A modularidade permite que se complete o desenvolvimento de uma solução praticamente utilizável dentro do escopo de uma tese. Exemplos disso são os portes para várias arquiteturas de processador, modificações do Minix para a virtualização com Xen e aplicações de segurança.

Em sua autobiografia, Linus Torvalds revela seus motivos para rejeitar a arquitetura de microkernel para o Linux: "A teoria por trás de um microkernel sempre foi a separação do kernel em 50 partes independentes, e cada uma delas possui 1/50 da complexidade. Mas todos ignoram o fato de que a comunicação entre as partes na verdade é mais complicada do que o sistema original – sem contar que as partes ainda são não triviais". Um sistema monolítico desorganizado, portanto, pode oferecer benefícios de desempenho e escalabilidade, mesmo que lhe falte a estabilidade de um microkernel.

O crescimento continuado dos sistemas operacionais traz a integração de novos drivers. Sistemas monolíticos incluem os drivers de dispositivos no kernel, o que significa que um erro no driver pode comprometer a estabilidade de todo o sistema. Drivers de código fechado, em particular, ameaçam a segurança do sistema. Segundo Tanenbaum, incluir no kernel um driver fechado é como aceitar um pacote lacrado de um estranho e levá-lo à cabine de comando de um avião.

Arquitetura transparente

O Minix é provavelmente o sistema operacional mais documentado da atualidade. O livro de Tanenbaum e Woodhull *The Minix Book* é a referência principal. Há várias publicações a respeito de novos recursos e pesquisas atuais na página do Minix 3

[2]. O Minix segue o padrão POSIX IEEE 1003.2-1996, e os desenvolvedores já portaram vários programas do Unix para ele.

Diferença

O Minix 3 está na ementa de várias universidades, e diversas gerações de estudantes já analisaram seus poucos milhares de linhas de código e consertaram a maioria dos erros. A arquitetura de microkernel implementa drivers como processos separados no espaço do usuário que não têm permissão de executar comandos privilegiados ou realizar operações de I/O, ou ainda de gravar diretamente na memória. Em vez disso, essas operações são realizadas por chamadas de sistema auditáveis (**figura 1**).

O sistema usa mensagens de tamanho fixo para a comunicação entre processos. Essa arquitetura simplifica a estrutura do código e ajuda a reduzir o risco de estouros de *buffer*. O sistema de arquivos do Minix é excecutado como um simples processo de usuário. Como ele é composto por aproximadamente 8.200 linhas de código de espaço do usuário, mas nenhum código do kernel, depurá-lo é fácil.

Um componente inovador, o servidor de reencarnação, aumenta a confiança do Minix por agir como pai de todos os servidores e drivers. Ele detecta travamentos rapidamente e monitora continuamente a função de processos críticos, reiniciando processos travados conforme necessário, para manter o sistema em funcionamento.

Firewall Minix

Os filtros de pacotes constituem um componente do sistema em risco. Apesar da excelente qualidade da implementação do *Netfilter* no Linux, várias falhas de segurança já surgiram. Se um subsistema desse tipo estiver em execução no kernel Linux, ele colocará em risco a segurança do sistema.

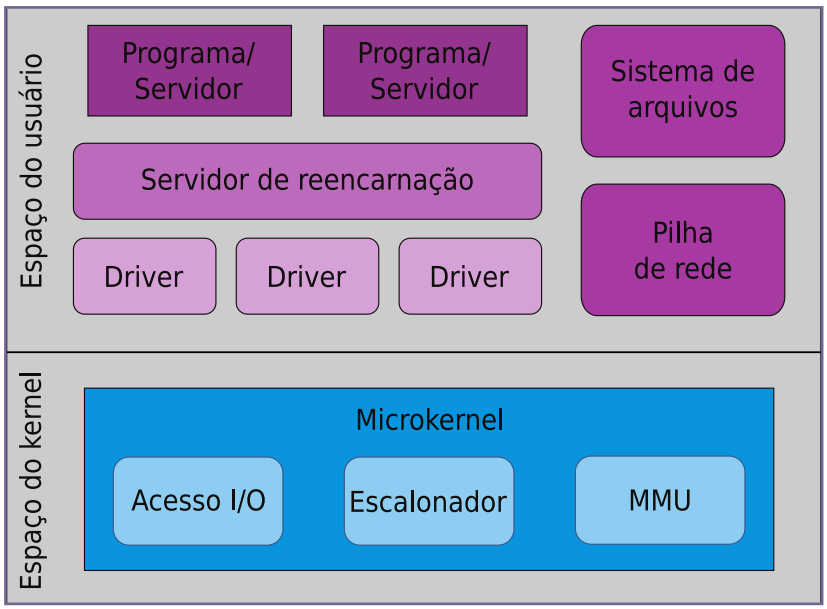


Figura 1 O microkernel Minix encapsula vários subsistemas no espaço do usuário, incluindo drivers, o sistema de arquivos e a pilha de rede. O kernel executa apenas funções críticas, como I/O, escalonadores e gerenciamento de memória.

Em colaboração com o grupo de Tanenbaum, a Universidade Técnica de Ciências Aplicadas de Berlim, Alemanha, portou a plataforma Netfilter para o Minix 3 [3].

Novamente, a estabilidade da arquitetura de microkernel promove mais vantagens. No Linux, um agressor que conseguir provocar um travamento – por exemplo, explorando um estouro de buffer na função `do_replace()` – pode destruir o firewall Linux. No Minix 3, um único processo de usuário travaria sem comprometer a segurança do sistema. O servidor de reencarnação então simplesmente reiniciaria o processo.

As diferenças se tornam ainda mais aparentes se o agressor conseguir executar algum código. No Minix, o sequestro de um processo de usuário ainda é problemático, mas seu efeito sobre o sistema é menos sério graças ao isolamento.

Até a Microsoft está explorando seu próprio sistema de microkernel, *Singularity* [4]. Apesar de o Minix já representar o time do microkernel há vários anos, seu maior obstáculo à ampla difusão sempre foi sua licença não livre. Agora que ele foi liberado sob a licença de código aberto BSD e as extensões de firewall estão disponíveis sob a GPL [3], os pesquisadores da TFH Berlin também estão explorando a potencialidade do Minix como firewall virtualizado. Estabilidade, baixo consumo e um novo modelo de licenciamento oferecem ao Minix 3 um grande potencial de crescimento, principalmente em sistemas embarcados. ■

Mais informações

[1] Andrew S. Tanenbaum, "Some Notes on the 'Who wrote Linux' Kerfuffle, Release 1.5" (em inglês): http://www.cs.vu.nl/~ast/brown/

[2] Minix: http://www.minix3.org

[3] Minixwall: http://wiki.tfh-berlin.de/~minixwall

[4] Microsoft Singularity: http://www.codeplex.com/singularity

Explorando as redes mesh IEEE 802.11s

A rede que se Mesh

As redes Mesh finalmente chegaram com o rascunho do padrão IEEE 802.11s. Veja como usá-las.

por Uwe Schwartz e Nils Magnus

Martin Simonis – www.sxc.hu

Redes sem fio comuns costumam depender da presença de um dispositivo de ponto de acesso, que serve como centro da conexão entre os clientes e o link externo. Essa tecnologia é tão difundida nas áreas urbanas de países desenvolvidos – e até em certas cidades brasileiras – que raramente o usuário se encontra longe de um ponto de conexão. Elas estão em aeroportos, cafés, escritórios e residências.

Apesar de sua onipresença, o ponto de acesso na realidade não é um elemento essencial das redes sem fio. As que o utilizam operam no modo chamado de *infraestrutura* (*infrastructure mode*). A forma alternativa de operação , o modo *ad hoc*, permite que os computadores estabeleçam conexões diretas entre si. O modo ad hoc costuma ser usado em conexões entre, digamos, um laptop e um desktop quando não há um ponto de acesso disponível. Computadores que se conectam dessa forma não têm os benefícios do DHCP ou outros serviços comuns no ponto de acesso, então precisam negociar todas as configurações da rede por si sós.

Em regiões remotas e países em desenvolvimento sem uma infraestrutura ampla de rede, alternativas descentralizadas como o modo ad hoc se tornam bem atraentes. Entretanto, os usuários que tentam operar numa configuração convencional de rede ad hoc esbarram em restrições rapidamente. Apesar de ser possível comunicar-se com vizinhos imediatos, a rede ad hoc convencional não permite que se use os vizinhos como intermediários para acesso a outros sistemas, então o alcance da rede fica restrito ao de um único dispositivo sem fio. Alguns sistemas operacionais, na verdade, nem mesmo suportam mais de dois pares numa rede ad hoc.

Uma extensão posterior do conceito ad hoc é a rede mesh, definida pelo rascunho do padrão IEEE 802.11s. A rede mesh expande o tamanho e o alcance das redes ad hoc ao permitir que os pares repassem mensagens e transmitam informações de rede para outros pares. Com isso, eles conseguem criar uma imagem coletiva da topologia da rede, e uma mensagem pode passar por uma corrente de pares até um nó além do alcance do sinal do dispositivo central.

O padrão IEEE 802.11s recebeu mais atenção por meio dos trabalhos do projeto One Laptop Per Child (OLPC, também conhecido originalmente como "Laptop de 100 dólares"). O OLPC foi criado com o objetivo de fornecer laptops com conectividade para milhões de alunos de países em desenvolvimento. Seus desenvolvedores precisavam de uma forma de interconectar esses pequenos laptops de forma a prover Internet a vilas inteiras que talvez só dispusessem de um único link. A rede mesh oferece essa solução [1]; mesmo que um dispositivo só consiga enxergar um par, ele ainda consegue acessar outros dispositivos (e, de forma ideal, também um ponto de acesso à Internet) por meio da transmissão de pacotes por uma corrente de conexões ao longo do *mesh*. A rede mesh também é útil no mundo desenvolvido, embora seja improvável que ela venha a substituir a conveniência e simplicidade das redes de pontos de acesso. Em áreas rurais ou em situações nas quais uma rede precise ser montada e auto-configurada temporariamente, as redes mesh podem se tornar importantes facilitadores.

Quadro 1: Questão de localização

Num cenário real de rede móvel, como o mundo dos laptops OLPC, o posicionamento dos nós na rede mesh é arbitrário, sendo impossível definir um local fixo para os pares. Entretanto, em outros cenários o administrador talvez tenha controle sobre a localização e a topologia geral da rede. Por exemplo, uma rede mesh montada temporariamente numa conferência. Nesses casos, é uma boa idéia escolher com cuidado a localização dos nós.

Para otimizar a velocidade, os dispositivos WLAN devem sempre ter visada direta, se possível. Se for possível um nó permanentemente se comunicar com um outro específico, pode ser usada uma antena direcional para melhorar o desempenho da rede.

O espaço no qual as ondas de rádio se propagam entre um transmissor e um receptor é chamado de zona de Fresnel. É um elipsóide no qual o transmissor e o receptor ocupam os pontos focais (veja a **figura 2**). Como o elipsóide fica mais alto no meio, os participantes da rede mesh devem posicionar seus nós o mais alto possível.

O IEEE 802.11s ainda é um rascunho [2], e dificilmente será aprovado oficialmente antes de 2010. Em agosto de 2008, o kernel Linux recebeu um *patch* para suporte a esse padrão no módulo *ath5k*, com suporte nativo disponível para os drivers *b43*, *libertas_tf* e *zd1211rw* de rede sem fio. Na versão 2.6.26, o *zd1211rw* já possuía suporte rudimentar ao padrão 802.11s com base no subsistema *mac80211* [3]. O código do kernel 2.6.26 não permite que os usuários criem redes mesh verdadeiras, mas desde a versão 2.6.27 esse suporte já é verdadeiro.

A menos que você possua um laptop XO ou algum dispositivo igualmente pré-configurado, usar uma rede mesh no Linux não é exatamente fácil para iniciantes, mas se você tem vontade de experimentar, as ferramentas necessárias certamente já estão disponíveis – mas certifique-se de que seu hardware suporta IEEE 802.11s.

Como funciona

Numa rede 802.11s, cada nó fica em contato com seus vizinhos diretos. Os nós podem ser desktops, laptops ou dispositivos como smartphones com recursos IP. O protocolo mesh tenta descobrir automaticamente a melhor rota para os pacotes (veja o **quadro 1** para mais informações sobre roteamento). Não há conexões fixas entre nós individuais. Cada nó transmite pacotes para um nó de destino específico que considera mais adequado para levar os pacotes para perto de seu destino final. O próximo pulo usa a mesma técnica, e assim por diante, até o pacote alcançar seu alvo.

Nesse fluxo, os dados são sincronizados com a maior frequência possível – normalmente em intervalos de poucos segundos. Essa técnica mantém a rede dinâmica e permite que ela reaja a alterações causadas pelo movimento dos nós, obstáculos ou conexões diretas. A rede avalia cada um dos caminhos ativos dos links e seleciona o melhor deles. Portanto, o roteamento é uma das tarefas mais críticas da rede, e também a mais "cara".

Em detalhes

Redes mesh com acesso à Internet geralmente possuem um nó principal que fornece o acesso à Internet por banda larga via DSL ou UMTS. Esse nó principal frequentemente oferece outros serviços de rede, como DHCP. O administrador da rede configura recursos de *gateway* nesse nó, digamos, com o comando `echo 1 > /proc/sys/net/ipv4/ip_forward`, e possivelmente cria regras de NAT para fazer o popular *masquerading*.

Os nós individuais da rede devem ter pelo menos dois módulos WLAN. Três módulos melhoram a disponibilidade e aumentam o número máximo de clientes. Normalmente, um módulo é usado

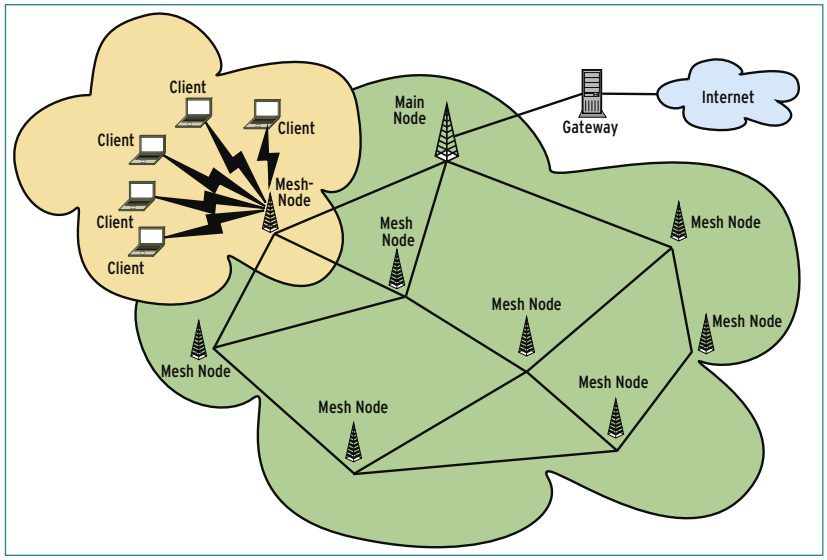


Figura 1 Uma rede típica de vila inclui dois dipos de participantes: clientes em modo de infraestrutura ligados a nós que se interligam por uma rede IEEE 802.11s, na qual um dos nós possui acesso à Internet.

Quadro 2: OLSR e Batman

Os desenvolvedores utilizam diferentes técnicas para implementar meshes. Duas opções populares são o roteamento otimizado de estado dos links (OLSR na sigla em inglês) [7] e a técnica melhor para redes *ad hoc* móveis (*Better Approach to Mobile Ad Hoc Networking*, ou simplesmente *Batman*) [8]. As duas opções usam o roteamento na camada 3 do modelo OSI para descobrir caminhos através da rede. O recente *Batman Advanced*, um *fork* do Batman sob desenvolvimento independente, usa o chaveamento na camada 2.

No OLSR, cada nó envia mensagens de *Hello* em intervalos fixos. Cada nó que recebe a mensagem avalia-a. Após criar um mapa de seu ambiente, o nó relata suas descobertas para seus vizinhos sob a forma de uma mensagem TC (mudança de topologia). Quando uma mensagem TC alcança um nó, o nó recalcula a topologia da rede. Isso significa que cada nó sabe, a todo instante, a melhor forma de rotear um pacote. Alterações à topologia são recalculadas pelo algoritmo de Dijkstra [9]. Nós individuais propagam seus conhecimentos sobre a estrutura da rede para seus vizinhos, melhorando de forma incremental o conhecimento de características individuais das rotas. O algoritmo é considerado estável, mas lento para convergir em alguns casos.

O OLSR não recalcula a topologia para cada pacote, pois precisa economizar ciclos de processamento. Ele usa a rede já existente enquanto recalcula a topologia. O OLSR avalia o número de saltos de rede até o destino através de uma certa rota, e assim calcula a melhor rota. Essa técnica garante uma rota curta para cada destino alcançável na rede após uma fase de inicialização. Como a estrutura da rede sem fio é suscetível a mudanças, é importante trocar mensagens *Hello* em intervalos relativamente curtos. O tempo de atualização típico é de cinco segundos. Além disso, chegam pacotes com mensagens vindos de outros nós, fazendo o nó recalcular a topologia. Isso pode ser problemático, especialmente para dispositivos pequenos com CPUs menos poderosas: esses dispositivos poderiam acabar gastando ciclos computacionais demais apenas para o cálculo da topologia.

Existem plugins e algoritmos alternativos para o OLSR reduzir o efeito de baixa escalabilidade do hardware. O Batman levou essa questão em consideração durante seu desenvolvimento. Os dois protocolos permitem que os usuários especifiquem se um nó tem acesso à Internet e portanto é capaz de agir como *gateway* para outros nós. Além disso, o Batman permite a configuração da banda usada pelo gateway para se conectar à Internet. Esse recurso permite que os clientes descubram o gateway mais favorável e elimina a dependência de uma entidade central para gerenciar essa informação.

A alternativa Batman tenta resolver o problema do OLSR com o desperdício de CPU para cálculo da topologia. O Batman usa uma técnica diferente para descobrir a melhor rota entre as disponíveis: exatamente como o OLSR, ele primeiro inunda a rede com mensagens de origem.

Cada nó gera uma mensagem e permite que seus vizinhos a distribuam. Cada vizinho então duplica cada mensagem recebida e a retransmite. Os nós não armazenam a topologia da rede completa, em contraste com o OLSR, mas apenas na parte necessária para alcançar seus vizinhos imediatos. Portanto, cada nó sabe apenas qual nó usar para transmitir um pacote, mas não como a transmissão se efetuará dali em diante. Essa técnica reduz o cálculo de rotas e permite que o protocolo escale mais facilmente. Embora o limite físico do OLSR seja de aproximadamente 100 nós – principalmente se os nós forem sistemas embarcados –, o Batman suporta redes com até 500 nós.

Como o Batman Advanced, mais recente, opera na camada de enlace, para o usuário a rede se comporta como um segmento gerenciado por um switch Ethernet. Do ponto de vista do usuário, cada participante tem conexão direta; o protocolo esconde os detalhes subjacentes. O Batman Advanced é o mais compatível com o padrão 802.11s, pois depende inteiramente da camada 2 (enlace). Infelizmente, esse esquema elimina a possibilidade de se definir um nó como gateway. Essa tarefa administrativa fica a cargo do administrador ou de um protocolo de camada superior.

no modo de ponto de acesso, para prover acesso a outros clientes; caso contrário os outros módulos usarão o modo ad hoc. Num mundo perfeito, todos esses módulos possuem antenas onidirecionais que cobrem uma grande área (**figura 1**).

Como se pode imaginar, a eficácia das redes mesh depende bastante do alcance do sinal dos nós. Com sistemas feitos especificamente para redes mesh, a configuração da antena recebe muito mais atenção. Uma rede com múltiplos módulos por nó pode usar uma configuração mais flexível para suas antenas; por exemplo, um módulo pode usar uma antena de setor para alcançar pares mais distantes enquanto outra trata dos nós próximos. As seções a seguir descrevem como configurar redes mesh no Linux.

Configuração

As versões mais recentes do kernel 2.6.26 e todas as versões do kernel 2.6.27 e 2.6.28 oferecem suporte a redes mesh. Se sua distribuição já fornece um kernel com uma dessas versões, você já tem todo o suporte de que necessita.

No espaço do usuário, é necessário usar a ferramenta *iw* para configurar a placa de rede [4].

Essa ferramenta tem como objetivo substituir, a longo prazo, o utilitário `iwconfig` e seus irmãos, assim como o `ip` se tornou uma alternativa ao `ifconfig`. O pacote do iw está ausente em várias distribuições; talvez seja necessário obter o código-fonte via *git*:

```
git clone http://git.sipsolutions/
➥net/iw.git
```

O pacote depende da *Libnl*, e é interessante usar a versão 1.0-pre8 ou outra mais recente [5]. Usuários de Debian e Ubuntu precisarão do pacote *libnl-dev*.

Para configurar a rede com a nova ferramenta, torne-se root e selecione uma ID para a rede. Essa ID precisa ser idêntica em todos os nós, além de precisar ter no máximo 32 caracteres; em nossos exemplos, usaremos a ID *MinhaMesh*. Em seguida, prossiga à configuração da interface – `wmaster0`, nesse caso:

```
# iw dev wmaster0 interface add \
mesh0 type mp mesh_id MinhaMesh
```

Isso criará uma nova interface, que ainda precisa ser configurada para operações com o gateway *10.0.0.1*. Para definir um IP e uma máscara de rede, pode-se usar o comando `ifconfig` comum, e depois o `iwconfig` para selecionar, por exemplo, o canal 7 para a rede mesh:

```
# iwconfig mesh0 channel 7
# ifconfig mesh0 100.0.0.1 \
netmask 255.255.255.0 up
```

Depois desses comandos para todos os dispositivos da rede mesh, a rede começará a convergir. Para verificar o progresso disso, use o seguinte comando:

```
iw dev mesh station dump
```

Os nós que forem alcançados pelo comando serão adicionados à lista. Isso permite que se verifique se o protocolo criou conexões entre os nós.

Para finalizar a configuração, é preciso configurar o acesso à Internet com DHCP e NAT no nó principal, que atuará como gateway para a rede mesh.

No Debian, bastaria configurar o pacote *dhcp3-server* em `/etc/dhcp/dhcpd.conf` conforme descrito no **exemplo 1**. Essa etapa configura uma subrede privada de endereço 10.0.0.0/24 para a mesh *minha-mesh.org*, define o gateway como *10.0.0.1* e informa os clientes para pedirem um servidor de

Exemplo 1: Configuração simples do servidor DHCP

```
01 subnet 10.0.0.0 netmask 255.255.255.0 {
02 range 10.0.0.100 10.0.0.199;
03 option routers 10.0.0.1;
04 option domain-name “my-mesh.org”;
05 option domain-name-servers 10.0.0.1;
06 default-lease-time 600;
07 }
```

nomes – o *Bind* 9, no Debian. No caso mais simples, seria preciso informar o IP de um servidor DNS da Internet na seção `forwarders` do arquivo `/etc/bind/named.conf.options`. A última etapa é ativar a configuração

```
net.ipv4.ip_forward=1
```

no arquivo `/etc/sysctl.conf` para rotear pacotes para fora da rede mesh (ou seja, para a Internet). Ao mesmo tempo, execute o seguinte comando para ativar o mascaramento (*masquerading*) dos endereços privados da rede sem fio no IP público do roteador:

```
# iptables -A
➥POSTROUTING \
-t nat -s
➥10.0.0.0/24 \
-j MASQUERADE
```

Com esses comandos está finalizada a configuração do nó principal.

Ainda é preciso executar o `iw` em cada nó para configurar a rede mesh. Os IPs e detalhes do DNS são fornecidos por DHCP. Se o objetivo for prover acesso também a clientes sem suporte a mesh, é preciso configurar outros módulos de rede sem fio com suporte ao funcionamento como ponto de acesso [6].

O suporte ao IEEE 802.11s no Linux ainda está no início. Precisaremos esperar um pouco até surgirem mais placas e chips de rede (com seus respectivos drivers) com suporte a redes mesh. Os usuários podem esperar que o IEEE 802.11s se torne bem mais simples a cada nova versão do kernel.

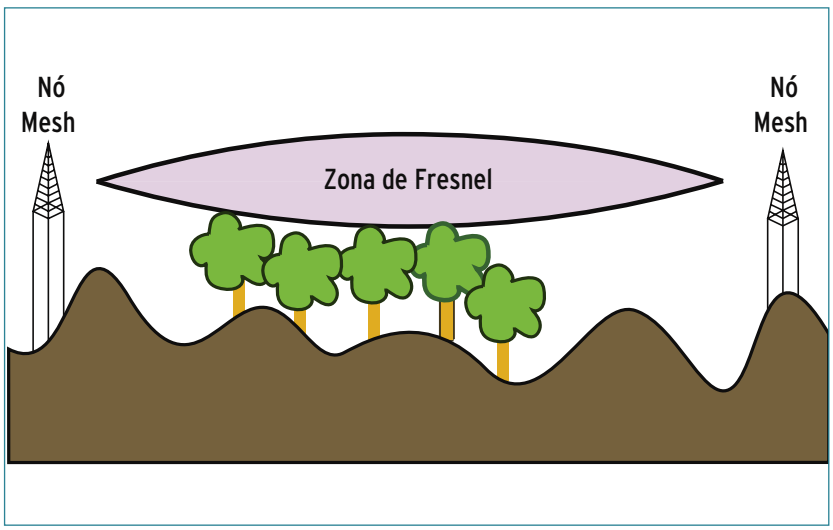


Figura 2 Os nós mesh devem ficar o mais alto possível na zona de Fresnel e sem obstáculos à visada direta entre eles.

Conclusões

Até o padrão ser amplamente adotado pelos fabricantes, não se surpreenda se houver alterações nos detalhes das ferramentas de configuração. Enquanto isso, com os dispositivos e ferramentas necessários, não há nada que impeça qualquer usuário de criar e configurar sua própria rede mesh imediatamente. ■

Mais informações

[1] Redes Mesh no OLPC: http://wiki.laptop.org/go/Mesh_Network_Details

[2] Grupo de trabalho IEEE 802.11s: http://grouper.ieee.org/groups/802/11/Reports/tgs_update.htm

[3] Projeto Open80211s: http://www.open80211s.org/trac/

[4] Howto iw: http://linuxwireless.org/en/users/Documentation/iw/

[5] Biblioteca Netlink: http://people.suug.ch/~tgr/libnl/

[6] Drivers para Host AP: http://hostap.epitest.fi

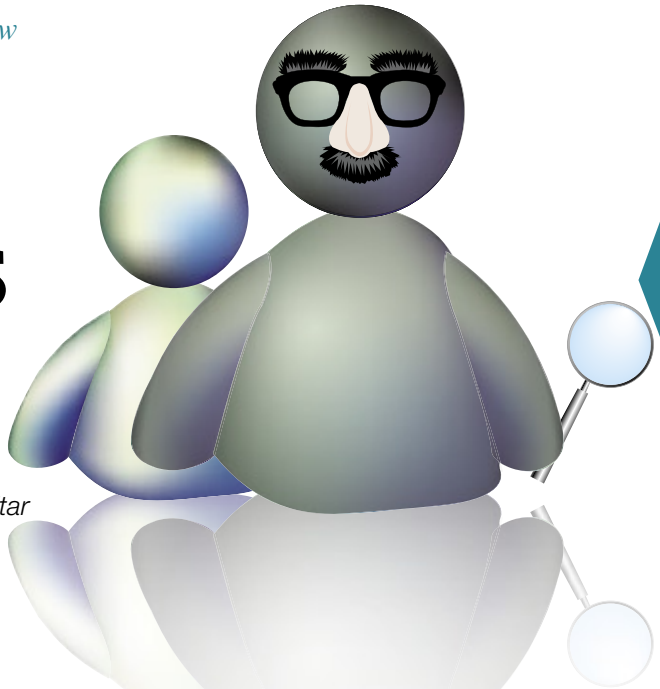
[7] Optimized Link State Routing (OLSR): http://www.olsr.org

[8] Páginas do Batman no Open-mesh.net: http://www.open-mesh.net/batman

[9] Algoritmo de Dijkstra: http://pt.wikipedia.org/wiki/Algoritmo_de_Dijkstra

Análise forense de mensagens instantâneas com o MSN Shadow

Mensagens instantâneas forenses



Escutas telefônicas já existem há tempos. Para evitar ou analisar crimes modernos via MSN, conheça o poderoso e completo MSN Shadow.

por Gabriel Menezes Nunes

Hoje em dia, com cada vez mais usuários migrando suas comunicações para a Internet por meio do uso de voz sobre IP, mensagens instantâneas e videoconferência, dentre outros, é necessário também que haja métodos de investigação dessas tecnologias. Da mesma forma que um usuário pode se comunicar com seus amigos ou colegas de trabalho, um pedófilo pode abordar uma criança ou traficantes podem se comunicar por meio desses novos protocolos.

O campo chamado de *Instant Messaging Forensics* (análise forense de mensagens instantâneas) é a área da computação forense que visa a estudar e localizar evidências em comunicações por mensagens instantâneas como MSN, Yahoo Messenger ou Jabber. Listas de contatos, conversações em texto, vídeo e arquivos transferidos são apenas algumas das evidências que podem ser coletadas nesse tipo de comunicação. Um exemplo prático é o uso de uma ferramenta de captura de tráfego MSN para a decodificação da lista de contatos de um determinado suspeito de tráfico de drogas. Com essa informação em mãos, um policial poderia ter uma lista de possíveis comparsas do meliante. Obviamente, para comprovar que tais contatos também são criminosos é necessária uma investigação mais detalhada de cada um deles.

Para suprir a falta de uma ferramenta mais completa que executasse uma análise do protocolo usado pelo MSN foi criado o software livre *MSN Shadow*, que captura e decodifica diversas informações pertinentes para investigações com foco em mensagens instantâneas. A escolha do MSN foi feita em razão de sua popularidade, mas o mesmo conceito pode ser aplicado a qualquer outro protocolo.

Esse projeto foi apresentado na última Conferência Internacional de Perícias em Crimes Cibernéticos (ICCyber 2008), organizada pela Polícia Federal do Brasil e com presença de diversos órgãos internacionais, como FBI, Serviço Secreto americano, Polícia Nacional da Espanha e outros.

Instalação

A instalação do MSN Shadow é simples e segue o clássico conjunto:

```
# ./configure
# make
# make install
```

No site oficial [1] há também pacotes binários em formato `.deb`.

Preparação

O MSN Shadow tem a capacidade de abrir arquivos no formato `.PCAP`, que é utilizado por praticamente todos os analisadores de tráfego (*sniffers*) disponíveis. No entanto, para uma investigação em tempo real, são necessários alguns passos antes da execução da ferramenta.

Para quem não tem familiaridade com equipamentos de rede, é importante relembrar. Hubs enviam todos os quadros para todos os hosts da rede, enquanto os switches enviam apenas para a interface de destino do quadro. Existe um domínio de *broadcast* para cada porta do switch e, em teoria, não é possível uma máquina ter acesso aos quadros de outra. Portanto, parece impraticável realizar uma análise de todo o tráfego da rede na presença de um switch, já que não é permitido o acesso a esse tráfego.

Uma opção é o espelhamento de tráfego, no qual uma determinada porta do switch é escolhida para a instalação do MSN Shadow e todo o tráfego do aparelho (ou seja, de todas as máquinas ligadas ao switch) será es-

pelhado para essa porta. Outra opção é a realização de ataques na camada de enlace como o *ARP Spoofing*. Com essa técnica, o *cache* ARP de todas as máquinas será modificado e, como consequência, elas começarão a enviar os quadros para uma determinada máquina. Isso é possível caso o investigador tenha instalado em sua máquina o pacote *dsniff* [2], que traz o binário `arpspoof`.

Atacar!

Para executar o ataque, primeiro é necessário ativar o encaminhamento de pacotes IP para que o tráfego da rede chegue aos destinos pretendidos. Em seguida, basta executar o ataque.

```
# echo 1 > /proc/sys/net/ipv4/
➥ip_forward
# arpspoof 10.0.0.1
# arpspoof 10.0.0.2
```

Nesses comandos, o endereço *10.0.0.1* pertence ao *gateway* da rede, enquanto *10.0.0.2* se refere à máquina que será analisada. Com isso, o `arpspoof` enviará pacotes de broadcast avisando a toda a rede que os IPs 10.0.0.1 e 10.0.0.2 possuem o endereço MAC da máquina atacante (aquela que está executando o `arpspoof`). Dessa forma, todos os pacotes com destino a esses IPs passarão primeiro pela máquina do investigador.

Agora o ambiente já está pronto para a investigação, já que os pacotes do MSN devem estar passando pela máquina do analista e já podem ser decodificados com o MSN Shadow.

Figura 1 Janela de configuração do *MSN Shadow*.

Uso do MSN Shadow

O uso do MSN Shadow é muito simples e sua interface é autoexplicativa (**figura 1**). Antes de iniciar qualquer tipo de análise em tempo real, é necessário configurar algumas opções, como interface de rede e regra de captura. Isso pode ser feito na janela *Options* do menu *File*. A interface escolhida é aquela na qual todo o tráfego da rede vai passar. No campo *Rule for Sniffing* há alguns valores padrão, como `port 1863`, que é a porta padrão utilizada no protocolo do MSN. Não há necessidade de mudar esses valores, mas existe a possibilidade, embora rara, de um cliente MSN usar a porta 80. As opções relativas ao vídeo seguem o mesmo pensamento, mas a regra para captura pode ser ignorada caso seja escolhida a autodetecção, que fará o software detectar automaticamente o tráfego relacionado a videoconferências. Na opção *mencoder path* deve estar o caminho para o binário do `mencoder`, o software de conversão de vídeos que acompanha o famoso reprodutor multimídia *Mplayer*. Essa opção é apenas necessária caso o investigador queira salvar o fluxo de vídeo para posterior análise. Esse fluxo será enviado ao `mencoder`, que o transformará num arquivo `.avi`.

Uma vez feita a configuração, basta pressionar *OK* e, na janela seguinte (**figura 2**), acionar o botão *Start Sniffing* tanto na aba *Text* quanto em *Video*, para o software começar a captura e a decodificação das informações. É necessário executar o MSN Shadow como root, já que a captura de pacotes só pode ser feita com permissões de administrador.

Com os dados coletados, é possível a visualização das listas de contatos, geração de relatórios HTML e diversas informações de conversas, como endereços IP, nomes de usuários, mensagens, horários etc.

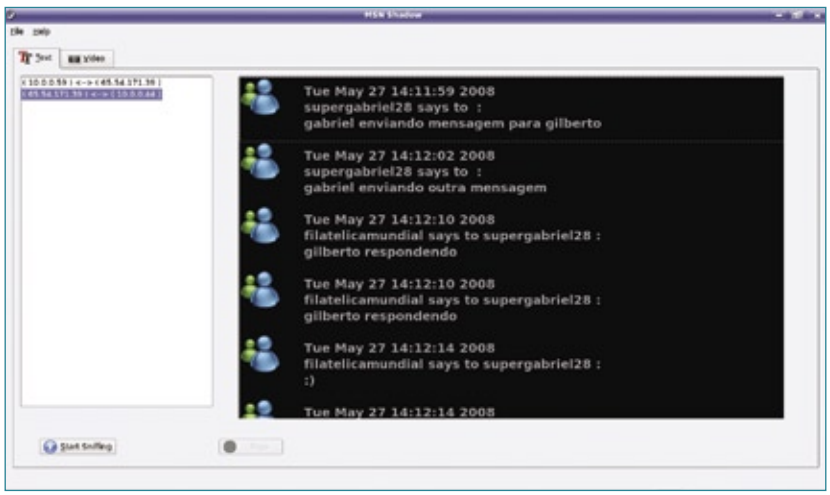


Figura 2 Janela de captura das conversas.

Investigação Ativa

Além das diversas informações que podem ser coletadas de forma passiva, o software MSN Shadow implementa algumas técnicas de investi-

gação ativa, como forjar mensagens, sequestrar totalmente a sessão MSN e derrubar de conexão.

Clicando com o botão direito do mouse sobre alguma das conversas decodificadas, é possível acessar o menu com essas opções. No forjamento simples de mensagens, é possível o investigador escolher um dos lados da comunicação e enviar uma mensagem falsa (*spoofing*). Isso é possível pois o software mantém em sua memória diversas informações, como endereços IP, portas, números de sequência e *ACK* (reconhecimento) e, quando requisitado pelo investigador, ele monta toda a pilha de protocolos desde a camada de rede, passando pela camada de transporte até a camada de aplicação. Para o investigador, é necessária apenas a digitação da mensagem. O MSN Shadow se encarrega de montar todos os cabeçalhos de baixo nível e enviá-los para a rede.

Na técnica de sequestro de sessão (*TCP Hijacking*), o investigador é capaz de se tornar qualquer um dos lados da conexão e se comunicar com o outro usuário sem que este perceba a mudança. Dessa forma, é possível coletar evidências e informações de forma mais ativa durante uma investigação. Um exemplo prático da necessidade dessas técnicas é a comunicação entre um pedófilo e uma criança, na qual um policial é capaz de sequestrar a conexão da criança e conversar com o pedófilo sem que este tome conhecimento da mudança. Isso é interessante pois o policial pode tentar uma abordagem direta ao pedófilo e este, desconfiado, pode não conversar com o policial. Caso sejam utilizadas essas técnicas ativas, o policial pode esperar pela abordagem do próprio pedófilo e, assim, sequestrar a conexão da crianca, sem gerar qualquer dúvida do lado do criminoso.

Na técnica de queda de conexão, é possível derrubar a conexão de um determinado usuário, mas apenas se este se encontrar na mesma rede local que o investigador, já que é necessário enviar um pacote *TCP RST* para a conexão desse usuário com o servidor MSN. Essa técnica é interessante para reduzir o vazamento de informações, por exemplo: um atacante interno pode estar enviando informações sigilosas da empresa por meio de tráfego MSN, e um investigador interno é capaz de derrubar essa conexão, evitando que mais dados críticos cheguem ao criminoso externo.

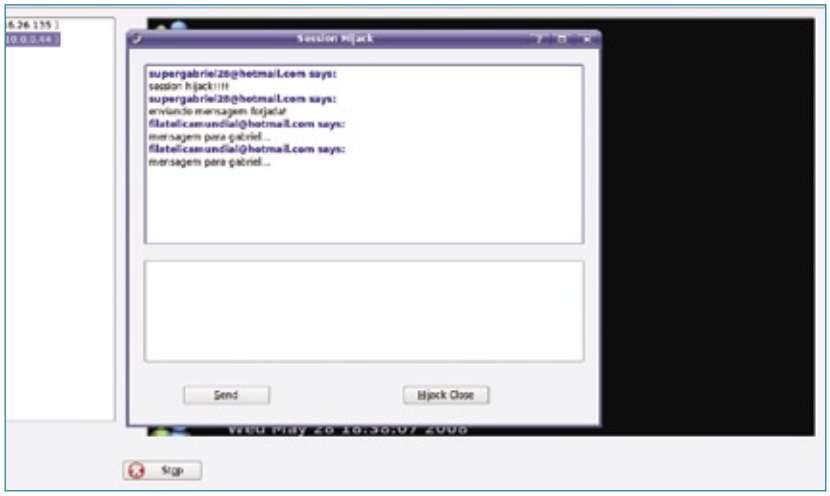


Figura 3 Sequestro de sessão.

Legalidade

Como este artigo demonstra, certamente é possível usar o MSN Shadow com objetivos maliciosos. No entanto, a questão não é a ferramenta, mas a conduta pessoal de cada um. Da mesma forma ,um carro pode ser utilizado para levar seus filhos à escola, mas também para matar uma pessoa. Um analisador de tráfego genérico como o *Wireshark* [3] pode ser utilizado para descobrir falhas na rede, assim como para capturar senhas em protocolos inseguros.

O uso dessa ferramenta depende da permissão do administrador da rede que será analisada, assim como a permissão daqueles que estão sendo analisados. Para autoridades como a polícia, é necessário mandado judicial para realizar grampos telefônicos num determinado suspeito, portanto é necessário também autorização jurídica para análise de tráfego de qualquer pessoa.

Como sempre, a única solução contra o abuso no uso dos recursos de uma infraestrutura de rede é a segurança. Controle do cache ARP, proteção de redes sem fio e métodos seguros de autenticação são algumas medidas importantes para impedir o uso de qualquer tipo de ferramenta de monitoramento sem a devida autorização. ■

Mais informações

[1] MSN Shadow: http://msnshadow.blogspot.com/

[2] Dsniff: http://monkey.org/~dugsong/dsniff/

[3] Wireshark: http://www.wireshark.org/

Sobre o autor

Gabriel Menezes Nunes é formado em Ciência da Computação pela Unesp/São José do Rio Preto e seu projeto final foi o software MSN Shadow. Gabriel também trabalha com outras áreas da segurança da informação, como testes de penetração e arquiteturas de rede (gab.mnunes@gmail.com) .

Análise de segurança com o hping

Nmap para quê?

Veja como testar as configurações do firewall e os sistemas de detecção de intrusões usando o hping.

por James Stanger

Para as tarefas de verificação de intrusões e da auditoria de segurança, a ferramenta *hping* é uma das melhores disponíveis na rede. Atualmente em sua terceira geração, o hping tornou-se o programa preferido para criar pacotes IP, normalmente utilizados com o propósito de testar firewalls e sistemas de detecção de intrusão.

Como o hping pode ser usado para manipular todos os campos, atributos e tipos de protocolo existentes na conjunto de protocolos baseados em TCP/IP, alguns usuários o apelidaram de "modelador de pacotes".

Com a manipulação dos pacotes, é possível realizar varreduras de sistemas (*portscans*) secretamente, gerar intenso tráfego de dados e, de modo geral, criar pacotes arbitrariamente conforme as necessidades de análise da rede. Através dos anos, o hping acabou se tornando uma referência quando a questão é gerar pacotes.

Aliás, a criação de pacotes customizados não é novidade. Outras ferramentas mais antigas de nomes tanto bombásticos quanto críticos, como *targa*, *synful*, *papa smurf* e *netdude*, eram capazes de auxiliar na tarefa de criação de pacotes sob medida. Muitas delas, entretanto, tinham problemas e limitações. Algumas, por exemplo, só funcionavam em redes IPv4 de classe C.

O que ele faz?

O hping fornece uma solução específica e universal que auxilia na prevenção de muitos dos problemas que acometiam as ferramentas da geração anterior. O programa é projetado para:

- realizar varreduras em computadores locais e remotos;
- auxiliar em análises de invasão;
- testar sistemas de detecção de intrusão e
- enviar arquivos de e para diversos computadores.

Este artigo explicará como gerar pacotes de teste usando essa ferramenta.

O *hping3* permite criar scripts razoavelmente sofisticados que auxiliam a simular o tráfego de pacotes para firewalls e sistemas de detecção de intrusão. Uma vantagem menos óbvia é o fato de que Salvatore Sanfilippo, criador e arquiteto do programa, reescreveu grande parte do código que lhe serve de base atualmente.

Instalação do hping

O código fonte do hping3 está disponível para download no site do projeto [1]. Porém, já existem pacotes binários disponíveis nos repositórios da maioria das principais distribuições. Pacotes para *Red Hat* ou *CentOS*, por exemplo, também estão disponíveis para download em [2].

Varreduras remotas

Após instalar o hping, pode-se começar a utilizá-lo diretamente. Não é necessária nenhuma configuração especial. Suponha que o objetivo seja enviar dois pacotes TCP para um sistema remoto cujo nome seja *james*, mais especificamente para a porta 80 desse sistema. Para fazer isso, o comando mostrado no **exemplo 1**

Exemplo 1: Uma varredura simples

```
01 pink@floyd:~/Desktop$ sudo hping3 -S james -c 2 -p 80
02 HPING james (eth0 192.168.15.134): S set, 40 headers + 0 data bytes
03 len=46 ip=192.168.15.134 ttl=64 DF id=0 sport=80 flags=SA seq=0 win=5840
➥rtt=0.3 ms
04 len=46 ip=192.168.15.134 ttl=64 DF id=0 sport=80 flags=SA seq=1 win=5840
➥rtt=0.3 ms
05
06 - james hping statistic -
07 2 packets transmitted, 2 packets received, 0% packet loss
08 round-trip min/avg/max = 0.3/0.3/0.3 ms
09 pink@floyd:~/Desktop$
```

Exemplo 2: Investigando UDP

```
01 sudo hping3 -2 192.168.44.45 -p ++44444 -T -n
02
03 HPING 192.168.44.45 (eth0 192.168.44.45): udp mode set, 28
➥headers + 0 data bytes
04 hop=1 TTL 0 during transit from ip=172.16.8.1
05 hop=1 hoprtt=1.7 ms
06 hop=2 TTL 0 during transit from ip=12.155.83.1
07 hop=2 hoprtt=2.7 ms
08 hop=3 TTL 0 during transit from ip=12.119.43.49
09 hop=3 hoprtt=10.0 ms
10 hop=4 TTL 0 during transit from ip=12.123.21.30
11 hop=4 hoprtt=13.6 ms
12 hop=5 TTL 0 during transit from ip=12.122.12.21
13 hop=5 hoprtt=13.3 ms
14 hop=6 TTL 0 during transit from ip=12.122.17.42
15 hop=6 hoprtt=11.9 ms
16 hop=7 TTL 0 during transit from ip=12.122.96.9
17 hop=7 hoprtt=36.6 ms
18 hop=8 TTL 0 during transit from ip=192.205.34.62
19 hop=8 hoprtt=13.6 ms
20 hop=9 TTL 0 during transit from ip=4.68.103.46
```

– que ilustra também o resultado do comando – deve ser executado.

No **exemplo 1**, perceba que o campo `flags =` está indicando o estado `SA`, que é a maneira de o hping indicar que a porta 80 está aberta na máquina *james*. Se ela estivesse fechada, o campo `flags =` indicaria o estado `RA`.

A opção `-S` envia um pacote *SYN*, que é frequentemente utilizado para criar varreduras difíceis de ser percebidas por sistemas de detecção de intrusões ou de ser marcadas como ameaças.

Depois que um sistema responde a um pacote *SYN*, fica-se automaticamente sabendo que há uma porta aberta (ou seja, um serviço "escutando", respondendo a solicitações) naquele sistema; o sistema de detecção de intrusões vai tratar o pacote *SYN* como tráfego comum, em vez de marcá-lo como ameaça.

A **figura 1** mostra como especificar uma varredura mais sofisticada que forneça um pequeno relatório – de boa qualidade – em texto puro.

E o Nmap?

É válido questionar por que usar o hping para procurar por portas abertas uma vez que existe o famoso *Nmap* para essa finalidade. É importante dizer que em algumas situações o hping oferece vantagens quando comparado ao Nmap.

Primeiramente, o hping é um aplicativo leve. Após sua instalação, ele já está pronto para o uso, o que de certo modo desestimula a instalação de outro aplicativo de mesma finalidade.

Em segundo lugar, é sempre bom saber fazer alguma coisa com mais de uma ferramenta. O criador do hping, por exemplo, continua desenvolvendo o aplicativo e fornecendo manutenção para ele, apesar de colaborar há vários anos com o Fyodor, criador do Nmap.

Terceiro: com o hping, é possível realizar varreduras incrementais, o que significa que cada varredura será feita isoladamente para cada porta do sistema:

```
sudo hping -S alvo -p ++0
```

Esse comando gera um relatório indicando quais portas estão abertas em um sistema.

Traceroute melhor?

Um recurso interessante do hping3 é a sua capacidade de gerar um relatório mais detalhado da rota de um pacote do que aquele produzido pelo *traceroute*, usando qualquer protocolo. Por exemplo, suponha que se deseje determinar exatamente o que acontece em cada um dos saltos (*hops*) da rota de um pacote. Para fazer isso, pode-se especificar o uso de um pacote *TCP SYN*. A opção `-T` permite que o recurso de traceroute do hping3 seja ativado. No comando mostrado acima, a opção `--ttl` permite que se especifique o número de roteadores (esto é, hops) que receberão o pacote.

Se o objetivo for executar um comando traceroute usando UDP, o comando mostrado no **exemplo 2** deve bastar. Sua saída ilustra como cada roteador processa o pacote UDP.

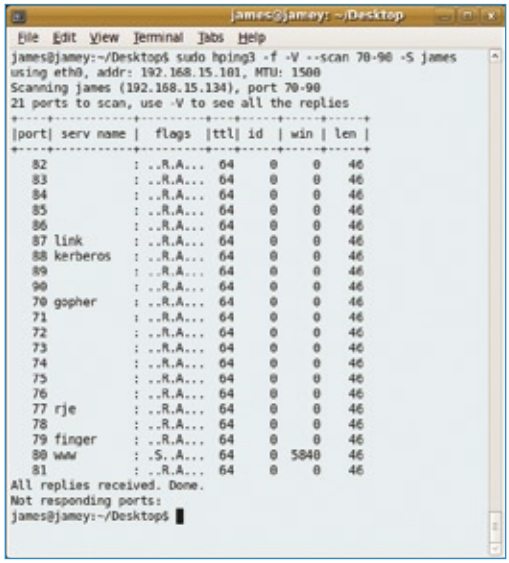


Figura 1 Um pacote gerado pelo hping observado no *Wireshark*.

Qual seria o propósito disso? O fato de que muitos roteadores bloqueiam tradicionalmente pacotes ICMP, mesmo que o último sistema utilizado use UDP.

Para analisar um hop em particular de um pacote de traceroute, pode-se usar a opção `--tr-keep-ttl`:

```
sudo hping3 -S 12.119.80.1 -p 80
➥-T --ttl 3 --tr-keep-ttl -n
```

A opção `-n` faz com que o endereço IP não seja resolvido.

O comando acima envia pacotes baseados em TCP para a máquina-alvo, mas gera um relatório apenas para o terceiro hop. A saída do programa é mostrada no **exemplo 3**, cujas informações podem ajudar a determinar exatamente como um dos roteadores está alterando os pacotes em trânsito.

Descobrindo o MTU

Para descobrir o MTU (*Maximum Transmission Unit* – o maior pacote permitido na rede), basta executar o seguinte comando:

```
hping3 -D -V -I
➥em1 --icmp alvo
```

Troque `alvo` pelo nome da máquina ou pelo endereço IP do sistema cujo MTU deve ser testado.

Por que é importante descobrir o MTU? Primeiro: conexões VPN e outras transmissões via rede às vezes tornam-se problemáticas se o MTU de um sistema ou da rede estiver mal configurado.

Em redes convergentes (por exemplo, nas quais se esteja configurando um sistema VoIP com os protocolos SIP ou H.323), pode ser necessário determinar o MTU para evitar problemas com latência e tráfego congestionado. Determinando o MTU e ajustando-o adequadamente no roteador ou em máquinas isoladamente, é possível reduzir atrasos e resolver problemas de qualidade em chamadas telefônicas VoIP que não poderiam ser solucionados de outra forma.

Exemplo 3: Análise de um hop

```
01 hop=3 TTL 0 during transit from ip=12.119.43.61
02 hop=3 hoprtt=31.7 ms
03 hop=3 TTL 0 during transit from ip=12.119.43.61
04 hop=3 hoprtt=6.9 ms
05 hop=3 TTL 0 during transit from ip=12.119.43.61
06 hop=3 hoprtt=5.0 ms
07 hop=3 TTL 0 during transit from ip=12.119.43.49
08 hop=3 hoprtt=5.2 ms
09 hop=3 TTL 0 during transit from ip=12.119.43.49
10 hop=3 hoprtt=5.2 ms
11 hop=3 TTL 0 during transit from ip=12.119.43.49
12 hop=3 hoprtt=4.9 ms
13 hop=3 TTL 0 during transit from ip=12.119.43.61
14 hop=3 hoprtt=5.4 ms
15 hop=3 TTL 0 during transit from ip=12.119.43.61
```

Exemplo 4: Envio de arquivos

```
01 Warning: Unable to guess the output interface
02 hping3 listen mode
03 [main] memlockall(): Success
04 Warning: can't disable memory paging!
05 99999:7:::
06 proxy:*:14181:0:99999:7:::
07 www-data:*:14181:0:99999:7:::
08 backup:*:14181:0:99999:99999:7:::
09 proxy:*:14181:0:99999:7:::
10 www-data:*:14181:0:99999:7:::
11 backup:*:14181:0:99999:7:::
12 list:*:14181:0:99999:7:::
13 irc:*:14181:0:99999:7:::
14 gnats:*:14181:0:99999:7:::
15 nobody:*7:::
16 nobody:*^C
17 [código cortado por pressionar Ctrl + C para terminar a transmissão]
18 - hping statistic -
19 0 packets transmitted, 0 packets received, 0% packet loss
20 round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Teste de perímetro

Teste de perímetro significa determinar exatamente o que o firewall bloqueia e o que ele deixa passar. Para fazer um teste de qualidade, pode-se capturar o endereço IP e as portas de origem dos pacotes:

```
sudo hping3 -a 10.0.44.45 -S james
➥-c 2 -p 80
```

O resultado do comando acima é que os pacotes vão parecer ter sido originados no sistema cujo endereço IP é o 10.0.44.45. Um pacote como esse é útil para se determinar se o firewall está permitindo a entrada e a saída de pacotes aleatórios da rede.

Nesses casos, não é preciso usar TCP. Com o hping, pode-se também criar pacotes UDP:

```
sudo hping3 alvo -c 2 --udp --
➥baseport 80 --destport 80
```

O comando acima envia dois pacotes UDP à porta 80 do sistema alvo,

Quadro 1: Versões

O *hping3* é a última versão do hping, sendo o *hping2* a versão mais importante antes dele. Vários programas ainda dependem do hping2, que esteve disponível por muito mais tempo para download do que o hping3, lançado, obviamente, há menos tempo.

É recomendável que as duas versões da ferramenta sejam instaladas. O hping3 pode ser utilizado como ferramenta isolada, mas é importante manter o hping2 instalado para o caso de algum aplicativo precisar utilizá-lo para funcionar, como o *scapy* (outra ferramenta para manipulação de pacotes) e o *idswakeup* (um programa para auditar sistemas de detecção de intrusões). O hping3 está equipado com um novo mecanismo de scripts em *Tcl,* sendo, portanto, muito mais poderoso do que uma ferramenta de linha de comando pura e simples.

As versões originais *hping* e hping2 funcionam como comandos isolados – ou seja, não iniciam uma sessão de shell interativa. Já se o hping3 for executado sem qualquer argumento, uma sessão hping é iniciada, mais ou menos como era o caso com o bom e velho comando `nslookup`.

a partir da porta 80 do sistema em que o comando foi executado.

Claro que é possível forjar o endereço IP do sistema de origem dos pacotes, bem como as portas de origem e destino:

```
sudo hping3 máquina_local -a
➥10.0.44.45 -c 2 --udp
➥--baseport 80 --destport 80
```

Envio de arquivos

Criar um túnel é um modo de descobrir o que o firewall é capaz de bloquear. Nos sistemas que deverão receber os pacotes, execute o seguinte comando:

```
sudo hping3 -i eth0 --listen
➥assinatura --icmp
```

Para enviar o conteúdo de um arquivo de uma máquina local para um sistema remoto de nome *james*, execute o seguinte comando:

```
sudo hping3 -I eth0 localhost
➥-- icmp -d 100 --sign assinatura
➥-- file /etc/shadow
```

No terminal do sistema receptor, pode-se observar a saída do arquivo que está sendo enviado (ver **exemplo 4**).

Perceba que o conteúdo do arquivo foi enviado através do firewall. Perceba também que o arquivo enviado tinha, propositalmente, conteúdo sensível do ponto de vista de segurança. A criação de um túnel *ad hoc* dessa forma permite a transferência de arquivos em ambas as direções – de dentro para fora e de fora para dentro – através do firewall. Adicionalmente, esse recurso é útil para

Quadro 2: Teste de penetração

Não basta saber como usar o hping3: é necessário também entender os fundamentos de um teste de penetração. Um teste como esse geralmente inclui os seguintes passos básicos:

- Identificação de recursos da rede: às vezes também chamado de *mapeamento da rede*, *identificação do alvo* ou *fotografia da rede*, este passo envolve a varredura dos sistemas à procura de portas abertas, a identificação de sistemas operacionais e a determinação do tipo de aplicações em operação por detrás das portas abertas.
- Busca de vulnerabilidades: essa busca se resume à procura de vulnerabilidades no servidor, no firewall e em sistemas operacionais de infraestruturas VoIP. Também é possível realizar testes projetados para subverter o esquema de autenticação vigente. Uma vez que a invasão ao sistema tenha sido realizada, confira prioridades aos recursos identificados. Por exemplo, um sistema pode ter uma vulnerabilidade séria que pode nem ser tão importante. Assim, pode-se conferir a esse sistema uma prioridade mais baixa que outros que sejam considerados mais essenciais, especialmente se tais sistemas não forem os prováveis escolhidos como palco para um ataque. Muitas vezes esse passo é considerado parte da identificação de recursos da rede, mas é mais indicado tratar essa atividade como algo separado. A determinação de vulnerabilidades é uma tarefa complexa que requer bastante pensamento analítico do profissional de TI.
- Teste de perímetro: é uma atividade clássica para o hping3. Por exemplo, o programa pode ser utilizado para gerar tráfego que testa se o firewall é capaz de bloquear pacotes manipulados.
- Teste de detecção de intrusão: neste passo, é gerado tráfego para verificar se o sistema de detecção de intrusão é capaz de identificar anomalias e problemas. Programas como o hping3 são simplesmente perfeitos para gerar esse tipo de tráfego "anormal".
- Levar em conta a política de segurança e a necessidade dos usuários: neste passo, deve-se determinar a eficácia da política de segurança e quão bem as aplicações da rede estão em conformidade com essa política. Também deve-se determinar quão alinhados os usuários estão com a política de segurança. Muito embora este último passo não seja realmente relevante para programas como o hping3, é importante entender que um auditor deve fazer mais que simplesmente varreduras em sistemas e geração de pacotes.

Quadro 3: Escolhendo um tipo de auditoria

Mesmo sob risco de simplificar demais, pode-se dizer que há dois tipos de auditoria: a cega e a não-cega. Uma auditoria cega é aquela na qual o auditor adota a perspectiva de um invasor que não conhece nada da rede que está invadindo e tem que descobrir todos os sistemas que fazem parte dela. No caso de uma auditoria não-cega, não é necessário descobrir quais são os sistemas disponíveis na rede; ao contrário, o auditor deve se concentrar na varredura de sistemas e na busca por vulnerabilidades. Qualquer que seja a abordagem adotada, a meta é descobrir os recursos disponíveis, mostrar como quebrar as defesas e demonstrar como um ataque poderia se espalhar para outros sistemas.

testar exatamente aquilo que um firewall é capaz de bloquear.

Simulação de ataques

O ataque batizado com o nome *LAND* [3], que apareceu pela primeira vez em 1997, envolve o envio de um pacote manipulado com a marca *SYN* ativada para uma máquina alvo. O pacote manipulado tem o mesmo endereço IP de origem e a mesma porta de origem que a máquina alvo. Quando esse ataque apareceu pela primeira vez, fez com que sistemas Windows sem atualizações de segurança criassem um loop infinito para a conexão e travassem.

Muitos agressores se utilizaram desse erro de implementação para realizar ataques de negação de serviço que eram simples, amadores e muito irritantes. Usuários mais sofisticados perceberam que tais ataques eram úteis na consecução de um outro tipo de ataque: o roubo de credenciais de acesso (também conhecidos como *hijacking attack*).

Uma nova variação do ataque LAND apareceu em 2005 e a técnica usada por ele poderia aparecer novamente, especialmente em razão de sua simplicidade.

O hping3 pode auxiliar na tarefa de garantir que um sistema se torne imune a esse tipo de ataque. Suponha que um sistema com IP 192.168.2.3 cuja porta 139 esteja aberta deva ser testado. Para fazer isso, basta executar o seguinte comando:

```
sudo hping3 -S 192.168.2.3 -a
➥192.168.2.3 -k -s 139 -p 139
➥ --flood
```

Esse ataque poderia levar um sistema alvo desatualizado a travar. Observe também a opção `--flood`, que envia milhares de pacotes à máquina alvo.

Calculando estados

Suponha que se deseje determinar quão bem o firewall é capaz de registrar solicitações para protocolos da Microsoft em uma rede. Para usar o hping3 para gerar os pacotes para esse teste, bastam os seguintes comandos:

```
hping www.acme.net -S -c 1 -p 139
hping www.acme.net -S -A -c 1
➥ -p 139
hping www.acme.net -S -A -c 1
➥ -p 135
```

Esses comandos criam pacotes que o firewall registrará caso o recurso de manutenção de estado esteja habilitado. Para verificar isso, basta olhar o conteúdo dos arquivos de registro do firewall e usar um analisador de pacotes.

Árvore de Natal

Um pacote do tipo "Árvore de Natal" [4] é um pacote TCP com quase todos os seus flags ativados, e é útil para criar desvios para o firewall, além de poder ser usado para realizar diversas outras formas de ataque.

Para criar um pacote desse tipo usando o hping3, execute o comando:

```
hping3 -F -P -U 10.44.45.15 -p 0
```

Firewalls e tempo

Em muitos casos, um firewall rejeitará automaticamente pacotes que não tenham uma marcação de tempo. Para adicionar uma marcação de tempo (*timestamp*) aos pacotes, basta usar a opção `-timestamp`, conforme ilustra o comando a seguir:

```
hping3 -S 72.14.207.99 -p 80
➥ --tcp-timestamp
```

Os resultados desse teste ajudam a determinar se é necessário ativar a função de filtragem de marcação de tempo no firewall.

Conclusão

Realizar testes de penetração e determinar a eficácia de sistemas de detecção de intrusão envolvem muito talento, além de uma dose extra de paciência. Além disso, é necessário ter à mão as ferramentas corretas. O hping3 é indubitavelmente uma dessas ferramentas. Este artigo resumiu apenas alguns dos comandos sofisticados à disposição do administrador de redes que adotar o hping em seu rol de aplicativos de verificação de segurança. ■

Mais informações

[1] Projeto hping: http://www.hping.org

[2] Pacote RPM do hping: http://dag.wieers.com/rpm/packages/hping

[3] Ataque LAND: http://en.wikipedia.org/wiki/LAND

[5] Pacote "Árvore de Natal": http://en.wikipedia.org/wiki/Christmas_tree_packett835.html

[6] Modelagem de pacotes com o hping: http://www.governmentsecurity.org/archive/t835.html

Otimize scripts bash para processadores multi-core

Shell paralelo

Você não precisa de grandes dados numéricos para tirar proveito do processamento paralelo. Conheça algumas técnicas simples para paralelizar scripts bash cotidianos.

por Bernhard Bablok

Lars Sundström – www.sxc.hu

Se você deseja que um componente de software execute uma tarefa em paralelo, o primeiro desafio é dividir aquela tarefa em subtarefas específicas que o computador possa executar simultaneamente. Bibliotecas como a *OpenMP* ajudam programadores a realizar esse tipo de paralelismo.

Scripts Bash geralmente não trabalham com problemas numéricos, logo a maioria dos programadores não imagina um script desses como candidato à paralelização. O venerável shell Bash, contudo, é utilizado para outros tipos de tarefas que o aproximam da abordagem paralela. Por exemplo, um script bash costuma ser empregado para processar diversos arquivos de uma mesma maneira.

O **exemplo 1** mostra uma função em shell que processa todos os argumentos no script, um a um, e passa o resultado para um programa (`doSomething`). Nesse cenário, é fácil imaginar os benefícios de algumas técnicas de paralelismo (veja também o **quadro 1**).

Exemplo 1: Processamento serial

```
01 fazSerial() {
02         local item
03         for item in "$@" do
04                 fazAlgo "$item"
05         done
06 }
```

Quadro 1: Por que paralelizar?

Antes de começar a paralelizar seus scripts bash, é importante calcular em quais situações isso faz sentido e é viável. Surpreendentemente, essa questão é fácil de responder. O comando `sar u P ALL 1 0` ajuda a responder a questão. O comando `sar` faz parte do pacote *Sysstat*.

Para realizar o teste, inicie seu script em um outro console. O comando `sar` mostra a carga de cada processador encontrado em seu sistema (**figura 1**).

Além do valor *%idle*, o valor *%iowait* também é importante. O valor *%iowait* mostra se o processamento foi interrompido devido à espera do sistema por I/O ou qualquer outra razão.

Os valores do `sar` facilitam a decisão: a paralelização só faz sentido se um dos processadores estiver ocioso enquanto o(s) outro(s) está(ão) sobrecarregado(s) (como demonstrado na **figura 1**). Aplicações típicas causadores desse fenômeno são conversões de imagens ou músicas, que produzem grande sobrecarga na CPU, ou análises de arquivos de log que utilizam expressões regulares complexas.

Processos diretamente relacionados a I/O não são bons candidatos. Mesmo que você possa paralelizar a cópia de 200 arquivos de um diretório para outro, não haverá redução no tempo de execução da tarefa, pois o gargalo da operação é a leitura e escrita no disco rígido, e não o processador.

Geralmente, se as etapas de processamento dependem umas das outras ou se a ordem do processamento é crucial, não há alternativa senão a execução sequencial. Um algoritmo diferente pode ajudar, mas a abordagem de paralelização proposta neste artigo não ajudará.

Além disso, administradores devem lembrar que nem sempre um sistema que ocupa todo o potencial de processamento oferece vantagens. À medida que você precisa realizar suas tarefas cotidianas nessa máquina (ler emails, navegar na Internet, redigir textos etc.) durante a execução de tarefas que sobrecarregam a CPU, a alternativa sequencial tradicional oferecerá melhor resposta do que a execução paralela, que prejudicará o desempenho como um todo.

Exemplo 2: Processamento paralelo massivo

```
01 fazMuitoParalelo() {
02     local item
03     for item in "$@" ;  do
04         fazAlgo "$item" &
05     done
06     wait
07 }
```

Exemplo 3: Entrada serial paralela

```
01 fazMuitoParalelo2() {
02     local item
03     while read item ; do
04         fazAlgo "$item" &
05     done
06     wait
07 }
08
09 criaItensDeTrabalho | fazMuitoParalelo
```

Força bruta

Pequenas modificações no código do **exemplo 1** criam a alternativa com processamento paralelo, mostrada no **exemplo 2**. O **exemplo 2** inicia um processo separado para cada argumento. Na **linha 6**, o script aguarda até que seus processos-filhos tenham finalizado. Essa abordagem pode causar problemas: se o sistema for saturado por um número excessivo de processos, a sobrecarga aumentará por causa das muitas mudanças de contexto. Em ambientes com memória limitada, o sistema perderá velocidade gradualmente conforme alterna entre um processo e outro. Em algumas situações, contudo, essa abordagem simplista da paralelização é satisfatória.

O **exemplo 3** é uma variação do **exemplo 2**. Nele, os argumentos são desconhecidos; em vez disso, um processo separado (`createWorkItems`) os cria sequencialmente – o que poderia ser uma chamada ao comando `find` para varrer um grande sistema de arquivos. Se as taxas de disparo e processamento, que dependem do número de processadores disponíveis, forem aproximadamente iguais, a sobrecarga do sistema não será notada. Se esse não for o caso, será necessária uma solução mais elaborada.

O script no **exemplo 4** distribui os argumentos dependendo do número de processadores e então trabalha as sequências. A **linha 1** do script determina o número de processadores (`PMAX`) no sistema. Se o processo depender fortemente de I/O (entrada/saída), faz sentido colocar o número de processos superior ao de `PMAX` para permitir que um processo trabalhe enquanto outro aguarda pelo I/O.

Infelizmente, o Bash utiliza apenas vetores unidimensionais, o que complica a construção das **linhas 6 e 13**. Para cada processo, o script cria uma longa linha contendo os argumentos para o processo dentro de um elemento do vetor (linhas 5 até 9). O script então dispara `PMAX` processos paralelos (**linhas 11 a 14**). A **linha 12** evita processos vazios (no caso de apenas dois argumentos numa máquina com quatro núcleos), e o `eval` na **linha 12** assegura que o shell interprete corretamente as aspas da **linha 6**.

Exemplo 4: Paralelismo com balanceamento de carga

```
01 ${PMAX:=`ls -d /sys/devices/system/cpu/cpu[0-9]* | wc l`}
02
03 fazParalelo() {
04     local itens item processoAtual=0
05     for item in "$@" ; do
06         itens[$processoAtual]="${itens[$processoAtual]} \"$item
07         shift
08         let processoAtual=$(( (processoAtual+1)%PMAX ))
09     done
10
11     for (( processoAtual=0 ; processoAtual<PMAX ; processoAtual
12         [ -n "${itens[$processoAtual]}" ] &&
13         eval fazSequencial ${itens[$processoAtual]} &
14     done
15     wait
16 }
```

Lançador dinâmico

O esquema mostrado no **exemplo 4** é opcional se o tempo de processamento médio para cada item não variar muito. Porém, não se deve confiar muito nesse modelo. Por exemplo, ao converter várias faixas de um CD, os diferentes comprimentos de cada uma farão com que alguns processos terminem antes dos outros. Outra situação em que essa abordagem pode ser problemática é a conversão de imagens a partir de uma câmera digital. Algumas câmeras criam arquivos de miniatura ou *JPEG* além dos arquivos *raw*. Se cada um dos demais arquivos usar o formato raw e precisar ser convertido, metade dos processos de conversão terminarão muito antes porque o esquema de processamento atribui todos os arquivos raw a um único processo e todos os arquivos JPEG a outro.

O método de processamento no **exemplo 4** também será prejudicado se nem todos os argumentos forem declarados com antecedência. Se os argumentos gerados posteriormente no script acontecerem em sequência, não faz sentido esperar até que todos sejam criados e então distribuí-los pelos processos.

A solução para esse problema é usar processos auxiliares e um lançador dinâmico. Nesse cenário, o script dispara certo número de processos auxiliares. O lançador aceita as tarefas e as distribui da forma mais inteligente possível para os auxiliares. Em contraste com as soluções paralelas demonstradas anteriormente, em que todos os processos auxiliares precisavam receber todos os argumentos no início, o lançador repassa-os aos auxiliares depois que eles iniciaram.

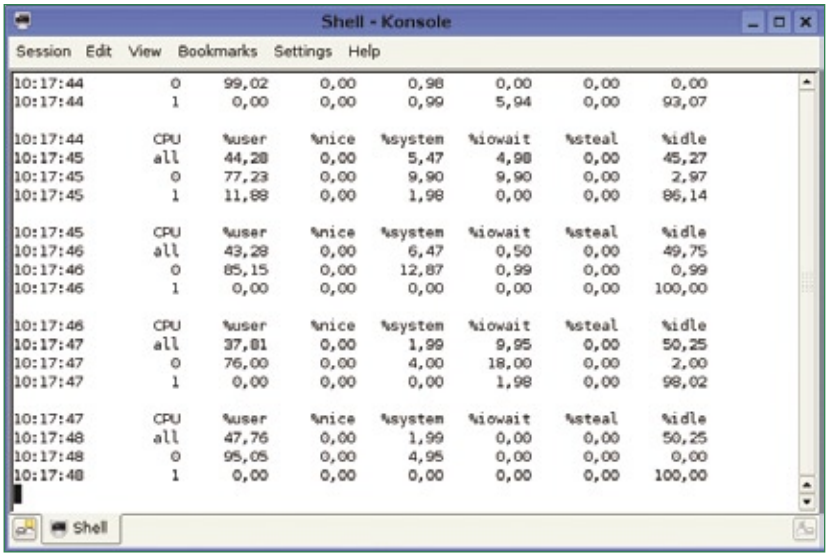


Figura 1 A saída do `sar` mostra a carga em todos os processadores. A paralelização só faz sentido se alguns núcleos estiverem sobrecarregados enquanto outros estiverem ociosos.

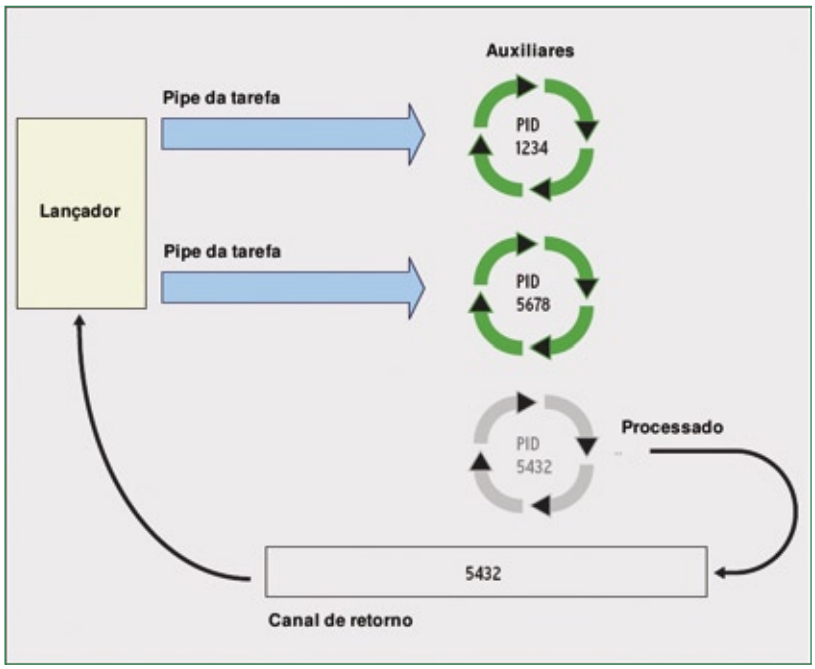


Figura 2 O lançador e os processos auxiliares usam pipes para se comunicarem.

Pipes nomeados (*named pipes*) e FIFOs são utilizados como canais de comunicação. No início, o lançador abre um pipe para cada auxiliar e envia novas tarefas para esse pipe (**figura 2**). Outro pipe é compartilhado pelo lançador e pelos auxiliares e age como um canal de retorno. Quando um auxiliar se torna ocioso, escreve seu ID no pipe. O lançador lê o ID do auxiliar no pipe após cada tarefa e envia a próxima tarefa para ele.

O **exemplo 5** mostra uma implementação desse conceito. Nas **linhas 1 a 4**, o programa define algumas constantes caso isso ainda não tenha sido feito. Normalmente, o usuário define apenas a variável `_cmd`. A função `dispatchWork` nas **linhas 54 a 72** é a parte pública da interface. A função começa criando um diretório temporário para todos os pipes na

linha 55 (chamado de `controlDir` no script). O comando `mkfifo` na **linha 58** define o canal de retorno.

A **linha 59** exige alguma explicação. Nela, o shell abre um canal de retorno para leitura e escrita, mesmo que ele precise apenas de leitura. O problema é que o acesso somente-leitura num pipe bloqueia a chamada de sistema. Um problema semelhante ocorre na função `startWorker()`, que cria um pipe para cada processo auxiliar (**linha 37**) e os abre para leitura e escrita (**linha 40**).

A instrução `eval` adicional na **linha 40** é necessária porque o Bash processa o redirecionamento da entrada **antes** da substituição de variáveis. Isso explica as barras invertidas antes dos sinais de menor e maior.

O **exemplo 5** simplesmente contém funções – outros scripts incluem esse arquivo e podem usar a função `dispatchWork` (**exemplo 5**).

Armadilhas

O script do **exemplo 5** tem algumas questões que merecem atenção. Por exemplo, o comando `kill` pode deixar processos auxiliares órfãos (que eventualmente poderiam ser tratados por uma variável de limite de tempo). Além disso, se houver mais de seis processos, o script usará descritores de arquivo (números de canal) maiores que 9. De acordo com o manual do Bash, é necessário ser cuidadoso

Exemplo 5: Lançador dinâmico

```
01 ${DEBUG:=0}
02 ${_cmd:=echo}
03 ${PMAX:=`ls /sys/devices/system/
➥cpu/cpu* | wc l`}
04 ${FDOFF:=4}
05
06 itemDeTrabalho() {
07     eval $_cmd \"$1\"
08 }
09
10 itensDeTrabalho() {
11     local linha fifoTrabalhador="$1"
➥fifoLancador="$2" id="$3" fd
12     exec 3<>"$fifoLancador"
13     while [ ! echo "$id" >&3 ]; do
14         sleep 1
15     done
16     let fd=id+FDOFF
17     while true ; do
18         read -r -u $fd linha
19         if [ $? -ne 0 ]; then
20             break
21         fi
22         if [ "$linha" = "EOF" ]; then
23             break
24         else
25             itemDeTrabalho "$linha"
26             while [ ! echo "$id" >&3
➥]; do
27                 sleep 1
28             done
29         fi
30     done
31     rm f "$fifoTrabalhador"
32 }
33
34 iniciaTrabalhador() {
35     local i fd fifo
36     for (( i=0 i<PMAX ++i )); do
37         fifoTrabalhador="$dirControle
➥/trabalhador$i"
38         mkfifo "$fifoTrabalhador"
39         let fd=i+FDOFF
40         eval exec $fd\<\>
➥"$fifoTrabalhador"
41         itensDeTrabalho "$fifoTrabalhado
➥"$fifoLancador" "$i" &
42     done
43 }
44
45 paraTrabalhador() {
46     local i fifo
47     for (( i=0 ; i<PMAX ; ++i )); do
48         fifo="$dirControle/trabalhador$i
49         echo "EOF" > "$fifo"
50     done
51     wait
52 }
53
54 lancaTrabalho() {
55     local idLivre fifoLancador
➥dirControle=`mktemp d`
56
57     fifoLancador="$dirControle/lancador"
58     mkfifo "$fifoLancador"
59     exec 3<>"$fifoLancador"
60
61     iniciaTrabalhador
62
63     while read -r -u 0 linha; do
64         read -u 3 idLivre
65         echo "$linha" >> "$dirControle
➥/trabalhador$idLivre"
66     done
67
68     paraTrabalhador
69
70     rm -f "$fifoLancador"
71     rm -fr "$dirControle"
72 }
```

Quadro 2: Benefícios

Nos dois testes realizados, foi usada a abordagem do lançador dinâmico demonstrada no artigo. No primeiro caso, o script converteu 20 arquivos *raw* para o formato *TIFF* numa máquina Intel Quad-core (Q9450 com 2.67 GHz e cache de 2x 6 MB L2).

Se os arquivos forem repassados para o `ufrawbatch` de uma só vez, o programa leva 132 segundos (interagindo de maneira autônoma pelos arquivos). O lançador dinâmico e as variáveis `PMAX=12` e `PMAX=4` reduziram o tempo de processamento de 134 segundos para 68 e 35 segundos. A eficiência desse método com quatro processadores, portanto, é de aproximadamente 95 por cento; ou seja, o tempo de execução caiu para um quarto.

A diferença entre essa abordagem e a paralelização estática é sutil. A razão para essa pequena diferença é que os arquivos originais são praticamente do mesmo tamanho. Dessa forma, todos os processadores são igualmente carregados.

O segundo cenário utiliza outro método para sobrecarregar o processador para converter arquivos *WAV* para *MP3*, mas dessa vez com condições mais adversas. O script lê e escreve num servidor NFS com uma conexão de rede de 100 Mbps. Observações interessantes sobre esse cenário são que, primeiro, o método escala muito bem (com tempos de execução de 207, 107 e 55 segundos, isto é, eficiência de 94 por cento nos quatro processadores).

Na segunda execução, quando os arquivos estavam no cache do servidor NFS, a diferença foi mínima comparada ao teste local. Finalmente, a utilização de cinco processos auxiliares no lugar de quatro produziu resultados um pouco melhores.

O efeito de mais processos auxiliares é mais acentuado no caso de fluxos de dados mais "estreitos". Contudo, classificar os arquivos WAV em ordem decrescente ocasionou melhor resultado na conversão. No fim do processamento, apenas um processador estava ocupado com o último arquivo, o que causou uma alteração desproporcional no tempo final. É possível implementar mais melhorias, mas essas já bastam. No caso de simulações complexas que levam horas ou dias, é recomendável testar com outras otimizações.

O equilíbrio de energia de um computador trabalhando com carga máxima é um pouco melhor do que se ele trabalhasse em processamento sequencial com apenas um núcleo. Contudo, economiza-se mais energia desligando o monitor enquanto o computador está ocupado com tarefas complexas de processamento.

com essa prática – seja lá o que isso quer dizer – porque o Bash já pode estar usando esses descritores internamente. Como alternativa, pode-se alterar o limite para o número de canais (**linha 4**).

Há outras implementações possíveis. Por exemplo, o lançador e os auxiliares poderiam usar arquivos para se comunicar. O lançador poderia então escrever as tarefas em arquivos específicos de cada auxiliar. Os auxiliares verificariam a presença de seu arquivo, processariam as tarefas ali definidas e então os apagariam. Na outra ponta, o lançador buscaria os lançadores sem arquivos correspondentes e saberia imediatamente quais deles estão ociosos.

Obviamente, essa solução não é das melhores, em razão da necessidade da verificação contínua de arquivos.

Uma versão mais longa do **exemplo 5** está disponível em [1]. Essa versão ampliada suporta chamadas de `dispatchWork` na linha de comando:

```
$ dispatchWork c "doSomething"
➥file1 file2 […]
```

A versão ampliada também possui comentários e outras opções para teste que permitem ao administrador monitorar os scripts.

Várias máquinas

Se a eficiência do processamento paralelo na máquina local não for suficiente, o mesmo princípio pode ser aplicado a uma rede. Nesse caso, um lançador de primeiro estágio pode usar comunicação TCP/IP para comunicar-se com lançadores de segundo estágio em outras máquinas. Os lançadores de segundo estágio comunicam-se então com os auxiliares locais. Porém, essa abordagem só é viável numa rede suficientemente segura.

Conclusões

Com apenas algumas linhas de código, é possível utilizar as técnicas demonstradas aqui para paralelizar scripts shell já existentes e desfrutar de benefícios potencialmente significativos (veja o **quadro 2**). Outras linguagens de script podem usar essa abordagem; contudo, algumas oferecem alternativas melhores. Por exemplo, *Python* usa um *fork* explícito (`os.fork()`) além dos pipes (`os.pipe()`), o que permite soluções de baixo nível muito próximas à eficiência do *C*. ■

Mais informações

[1] Código-fonte do lançador dinâmico: http://www.linuxmagazine.com.br/arquivos/lm51/programacao_bash.tar.gz

Linux.local

O maior diretório de empresas que oferecem produtos, soluções e serviços em Linux e Software Livre, organizado por Estado. Sentiu falta do nome de sua empresa aqui? Entre em contato com a gente: **11 4082-1300** *ou* **anuncios@linuxmagazine.com.br**

Fornecedor de Hardware = 1
Redes e Telefonia / PBX = 2
Integrador de Soluções = 3
Literatura / Editora = 4
Fornecedor de Software = 5
Consultoria / Treinamento = 6

Empresa	Cidade	Endereço	Telefone	Web	1	2	3	4	5	6
Ceará										
F13 Tecnologia	Fortaleza	Rua Coronel Solon, 480 – Bairro de Fátima Fortaleza - CE - CEP: 60040-270	85 3252-3836	www.f13.com.br		✔	✔		✔	✔
Espírito Santo										
Linux Shopp	Vila Velha	Rua São Simão (Correspondência), 18 – CEP: 29113-120	27 3082-0932	www.linuxshopp.com.br	✔	✔		✔	✔	
Megawork Consultoria e Sistemas	Vitória	Rua Chapot Presvot, 389 – Praia do Canto – CEP: 29055-410 sl 201, 202	27 3315-2370	www.megawork.com.br			✔		✔	✔
Spirit Linux	Vitória	Rua Marins Alvarino, 150 – CEP: 29047-660	27 3227-5543	www.spiritlinux.com.br			✔		✔	✔
Minas Gerais										
Instituto Online	Belo Horizonte	Av. Bias Fortes, 932, Sala 204 – CEP: 30170-011	31 3224-7920	www.institutoonline.com.br				✔		✔
Linux Place	Belo Horizonte	Rua do Ouro, 136, Sala 301 – Serra – CEP: 30220-000	31 3284-0575	corporate.linuxplace.com.br		✔	✔		✔	✔
Microhard	Belo Horizonte	Rua República da Argentina, 520 – Sion – CEP: 30315-490	31 3281-5522	www.microhard.com.br	✔	✔	✔		✔	✔
TurboSite	Belo Horizonte	Rua Paraíba, 966, Sala 303 – Savassi – CEP: 30130-141	0800 702-9004	www.turbosite.com.br	✔				✔	✔
Paraná										
iSolve	Curitiba	Av. Cândido de Abreu, 526, Cj. 1206B – CEP: 80530-000	41 252-2977	www.isolve.com.br		✔	✔			✔
Mandriva Conectiva	Curitiba	Rua Tocantins, 89 – Cristo Rei – CEP: 80050-430	41 3360-2600	www.mandriva.com.br			✔	✔	✔	✔
Telway Tecnologia	Curitiba	Rua Francisco Rocha 1830/71	41 3203-0375	www.telway.com.br					✔	✔
Rio de Janeiro										
Múltipla Tecnologia da Informação	Rio de Janeiro	Av. Rio Branco, 37, 14° andar – CEP: 20090-003	21 2203-2622	www.multipla-ti.com.br	✔		✔		✔	✔
NSI Training	Rio de Janeiro	Rua Araújo Porto Alegre, 71, 4º andar Centro – CEP: 20030-012	21 2220-7055	www.nsi.com.br				✔		✔
Open IT	Rio de Janeiro	Rua do Mercado, 34, Sl, 402 – Centro – CEP: 20010-120	21 2508-9103	www.openit.com.br				✔		✔
Unipi Tecnologias	Campos dos Goytacazes	Av. Alberto Torres, 303, 1ºandar – Centro – CEP: 28035-581	22 2725-1041	www.unipi.com.br			✔	✔	✔	✔
Rio Grande do Sul										
4up Soluções Corporativas	Novo Hamburgo	Pso. Calçadão Osvaldo Cruz, 54 sl. 301 CEP: 93510-015	51 3581-4383	www.4up.com.br		✔	✔		✔	✔
Definitiva Informática	Novo Hamburgo	Rua General Osório, 402 - Hamburgo Velho	51 3594 3140	www.definitiva.com.br	✔		✔		✔	✔
Solis	Lajeado	Av. 7 de Setembro, 184, sala 401 – Bairro Moinhos CEP: 95900-000	51 3714-6653	www.solis.coop.br		✔	✔	✔	✔	✔
DualCon	Novo Hamburgo	Rua Joaquim Pedro Soares, 1099, Sl. 305 – Centro	51 3593-5437	www.dualcon.com.br	✔		✔		✔	✔
Datarecover	Porto Alegre	Av. Carlos Gomes, 403, Sala 908, Centro Comercial Atrium Center – Bela Vista – CEP: 90480-003	51 3018-1200	www.datarecover.com.br	✔		✔			
LM2 Consulting	Porto Alegre	Rua Germano Petersen Junior, 101-Sl 202 – Higienópolis – CEP: 90540-140	51 3018-1007	www.lm2.com.br			✔		✔	✔
Lnx-IT Informação e Tecnologia	Porto Alegre	Av. Venâncio Aires, 1137 – Rio Branco – CEP: 90.040.193	51 3331-1446	www.lnx-it.inf.br	✔		✔		✔	✔
Plugin	Porto Alegre	Av. Júlio de Castilhos, 132, 11º andar Centro – CEP: 90030-130	51 4003-1001	www.plugin.com.br	✔		✔		✔	
TeHospedo	Porto Alegre	Rua dos Andradas, 1234/610 – Centro – CEP: 90020-008	51 3286-3799	www.tehospedo.com.br	✔	✔				
São Paulo										
Ws Host	Arthur Nogueira	Rua Jerere, 36 – Vista Alegre – CEP: 13280-000	19 3846-1137	www.wshost.com.br	✔		✔		✔	
DigiVoice	Barueri	Al. Juruá, 159, Térreo – Alphaville – CEP: 06455-010	11 4195-2557	www.digivoice.com.br	✔	✔	✔		✔	✔
Dextra Sistemas	Campinas	Rua Antônio Paioli, 320 – Pq. das Universidades – CEP: 13086-045	19 3256-6722	www.dextra.com.br			✔		✔	✔
Insigne Free Software do Brasil	Campinas	Av. Andrades Neves, 1579 – Castelo – CEP: 13070-001	19 3213-2100	www.insignesoftware.com			✔		✔	✔
Microcamp	Campinas	Av. Thomaz Alves, 20 – Centro – CEP: 13010-160	19 3236-1915	www.microcamp.com.br				✔		✔
PC2 Consultoria em Software Livre	Carapicuíba	Rua Edeia, 500 - CEP: 06350-080	11 3213-6388	www.pc2consultoria.com	✔					✔
Savant Tecnologia	Diadema	Av. Senador Vitorino Freire, 465 – CEP: 09910-550	11 5034-4199	www.savant.com.br	✔	✔	✔			✔
Epopéia Informática	Marília	Rua Goiás, 392 – Bairro Cascata – CEP: 17509-140	14 3413-1137	www.epopeia.com.br						✔
Redentor	Osasco	Rua Costante Piovan, 150 – Jd. Três Montanhas – CEP: 06263-270	11 2106-9392	www.redentor.ind.br	✔					
Go-Global	Santana de Parnaíba	Av. Yojiro Takaoca, 4384, Ed. Shopping Service, Cj. 1013 – CEP: 06541-038	11 2173-4211	www.go-global.com.br			✔		✔	✔
AW2NET	Santo André	Rua Edson Soares, 59 – CEP: 09760-350	11 4990-0065	www.aw2net.com.br			✔		✔	✔
Async Open Source	São Carlos	Rua Orlando Damiano, 2212 – CEP 13560-450	16 3376-0125	www.async.com.br	✔				✔	✔

Empresa	Cidade	Endereço	Telefone	Web	1	2	3	4	5	6
São Paulo (continuação)										
Delix Internet	São José do Rio Preto	Rua Voluntário de São Paulo, 3066 9º – Centro – CEP: 15015-909	11 4062-9889	www.delixhosting.com.br	✔		✔		✔	
4Linux	São Paulo	Rua Teixeira da Silva, 660, 6º andar – CEP: 04002-031	11 2125-4747	www.4linux.com.br					✔	✔
A Casa do Linux	São Paulo	Al. Jaú, 490 – Jd. Paulista – CEP: 01420-000	11 3549-5151	www.acasadolinux.com.br			✔		✔	✔
Accenture do Brasil Ltda.	São Paulo	Rua Alexandre Dumas, 2051 – Chácara Santo Antônio – CEP: 04717-004	11 5188-3000	www.accenture.com.br			✔		✔	✔
ACR Informática	São Paulo	Rua Lincoln de Albuquerque, 65 –Perdizes – CEP: 05004-010	11 3873-1515	www.acrinformatica.com.br	✔					✔
Agit Informática	São Paulo	Rua Major Quedinho, 111, 5º andar, Cj. 508 – Centro – CEP: 01050-030	11 3255-4945	www.agit.com.br	✔	✔				✔
Altbit - Informática Comércio e Serviços LTDA.	São Paulo	Av. Francisco Matarazzo, 229, Cj. 57 – Água Branca – CEP 05001-000	11 3879-9390	www.altbit.com.br	✔		✔		✔	✔
AS2M -WPC Consultoria	São Paulo	Rua Três Rios, 131, Cj. 61A – Bom Retiro – CEP: 01123-001	11 3228-3709	www.wpc.com.br			✔		✔	✔
Big Host	São Paulo	Rua Dr. Miguel Couto, 58 – Centro – CEP: 01008-010	11 3033-4000	www.bighost.com.br	✔				✔	✔
Blanes	São Paulo	Rua André Ampére, 153 – 9º andar – Conj. 91 CEP: 04562-907 (próx. Av. L. C. Berrini)	11 5506-9677	www.blanes.com.br	✔	✔	✔		✔	✔
Commlogik do Brasil Ltda.	São Paulo	Av. das Nações Unidas, 13.797, Bloco II, 6º andar – Morumbi – CEP: 04794-000	11 5503-1011	www.commlogik.com.br	✔	✔	✔		✔	✔
Computer Consulting Projeto e Consultoria Ltda.	São Paulo	Rua Vergueiro, 6455, Cj. 06 – Alto do Ipiranga – CEP: 04273-100	11 5062-3927	www.computerconsulting.com.br	✔		✔		✔	✔
Consist Consultoria, Sistemas e Representações Ltda.	São Paulo	Av. das Nações Unidas, 20.727 – CEP: 04795-100	11 5693-7210	www.consist.com.br			✔	✔	✔	✔
Domínio Tecnologia	São Paulo	Rua das Carnaubeiras, 98 – Metrô Conceição – CEP: 04343-080	11 5017-0040	www.dominiotecnologia.com.br	✔					✔
EDS do Brasil	São Paulo	Av. Pres. Juscelino Kubistcheck, 1830 Torre 4 - 5º andar	11 3707-4100	www.eds.com		✔	✔			✔
Ética Tecnologia	São Paulo	Rua Nova York, 945 – Brooklin – CEP:04560-002	11 5093-3025	www.etica.net	✔		✔		✔	✔
Getronics ICT Solutions and Services	São Paulo	Rua Verbo Divino, 1207 – CEP: 04719-002	11 5187-2700	www.getronics.com/br			✔		✔	✔
Hewlett-Packard Brasil Ltda.	São Paulo	Av. das Nações Unidas, 12.901, 25º andar – CEP: 04578-000	11 5502-5000	www.hp.com.br	✔		✔	✔	✔	✔
IBM Brasil Ltda.	São Paulo	Rua Tutóia, 1157 – CEP: 04007-900	0800-7074 837	www.br.ibm.com	✔		✔		✔	✔
iFractal	São Paulo	Rua Fiação da Saúde, 145, Conj. 66 – Saúde – CEP: 04144-020	11 5078-6618	www.ifractal.com.br			✔		✔	✔
Integral	São Paulo	Rua Dr. Gentil Leite Martins, 295, 2º andar Jd. Prudência – CEP: 04648-001	11 5545-2600	www.integral.com.br	✔				✔	
Itautec S.A.	São Paulo	Av. Paulista, 2028 – CEP: 01310-200	11 3543-5543	www.itautec.com.br	✔	✔	✔		✔	✔
Kenos Consultoria	São Paulo	Av: Fagundes Filho, 13, Conj 53 – CEP: 04304-000	11 40821305	www.kenos.com.br					✔	✔
Konsultex Informatica	São Paulo	Av. Dr. Guilherme Dumont Villares, 1410 6 andar, CEP: 05640-003	11 3773-9009	www.konsultex.com.br			✔		✔	✔
Linux Komputer Informática	São Paulo	Av. Dr. Lino de Moraes Leme, 185 – CEP: 04360-001	11 5034-4191	www.komputer.com.br	✔		✔		✔	✔
Linux Mall	São Paulo	Rua Machado Bittencourt, 190, Cj. 2087 – CEP: 04044-001	11 5087-9441	www.linuxmall.com.br	✔			✔	✔	
Livraria Tempo Real	São Paulo	Al. Santos, 1202 – Cerqueira César – CEP: 01418-100	11 3266-2988	www.temporeal.com.br				✔	✔	✔
Locasite Internet Service	São Paulo	Av. Brigadeiro Luiz Antonio, 2482, 3º andar – Centro – CEP: 01402-000	11 2121-4555	www.locasite.com.br	✔				✔	✔
Microsiga	São Paulo	Av. Braz Leme, 1631 – CEP: 02511-000	11 3981-7200	www.microsiga.com.br			✔		✔	✔
Novatec Editora Ltda.	São Paulo	Rua Luis Antonio dos Santos, 110 – Santana – CEP: 02460-000	11 6979-0071	www.novateceditora.com.br				✔		
Novell América Latina	São Paulo	Rua Funchal, 418 – Vila Olímpia	11 3345-3900	www.novell.com/brasil			✔		✔	✔
Oracle do Brasil Sistemas Ltda.	São Paulo	Av. Alfredo Egídio de Souza Aranha, 100 – Bloco B – 5º andar – CEP: 04726-170	11 5189-3000	www.oracle.com.br					✔	✔
Proelbra Tecnologia Eletrônica Ltda.	São Paulo	Av. Rouxinol, 1.041, Cj. 204, 2º andar Moema – CEP: 04516-001	11 5052- 8044	www.proelbra.com.br	✔		✔			✔
Provider	São Paulo	Av. Cardoso de Melo, 1450, 6º andar – Vila Olímpia – CEP: 04548-005	11 2165-6500	www.e-provider.com.br			✔		✔	✔
Red Hat Brasil	São Paulo	Av. Brigadeiro Faria Lima, 3900, Cj 81 8º andar Itaim Bibi – CEP: 04538-132	11 3529-6000	www.redhat.com.br			✔		✔	✔
Samurai Projetos Especiais	São Paulo	Rua Barão do Triunfo, 550, 6º andar – CEP: 04602-002	11 5097-3014	www.samurai.com.br			✔		✔	✔
SAP Brasil	São Paulo	Av. das Nações Unidas, 11.541, 16º andar – CEP: 04578-000	11 5503-2400	www.sap.com.br			✔		✔	✔
Simples Consultoria	São Paulo	Rua Mourato Coelho, 299, Cj. 02 Pinheiros – CEP: 05417-010	11 3898-2121	www.simplesconsultoria.com.br			✔		✔	✔
Smart Solutions	São Paulo	Av. Jabaquara, 2940 cj 56 e 57	11 5052-5958	www.smart-tec.com.br		✔	✔		✔	✔
Snap IT	São Paulo	Rua João Gomes Junior, 131 – Jd. Bonfiglioli – CEP: 05299-000	11 3731-8008	www.snapit.com.br			✔		✔	✔
Stefanini IT Solutions	São Paulo	Av. Brig. Faria Lima, 1355, 19º – Pinheiros – CEP: 01452-919	11 3039-2000	www.stefanini.com.br			✔		✔	✔
Sun Microsystems	São Paulo	Rua Alexandre Dumas, 2016 – CEP: 04717-004	11 5187-2100	www.sun.com.br	✔		✔		✔	✔
Sybase Brasil	São Paulo	Av. Juscelino Kubitschek, 510, 9º andar Itaim Bibi – CEP: 04543-000	11 3046-7388	www.sybase.com.br					✔	✔
The Source	São Paulo	Rua Marquês de Abrantes, 203 – Chácara Tatuapé – CEP: 03060-020	11 6698-5090	www.thesource.com.br			✔		✔	✔
Unisys Brasil Ltda.	São Paulo	R. Alexandre Dumas 1658 – 6º, 7º e 8º andares – Chácara Santo Antônio – CEP: 04717-004	11 3305-7000	www.unisys.com.br	✔		✔		✔	✔
Utah	São Paulo	Av. Paulista, 925, 13º andar – Cerqueira César – CEP: 01311-916	11 3145-5888	www.utah.com.br			✔		✔	✔
Visuelles	São Paulo	Rua Eng. Domicio Diele Pacheco e Silva, 585 – Interlagos – CEP: 04455-310	11 5614-1010	www.visuelles.com.br			✔		✔	✔
Webnow	São Paulo	Av. Nações Unidas, 12.995, 10º andar, Ed. Plaza Centenário – Chácara Itaim – CEP: 04578-000	11 5503-6510	www.webnow.com.br	✔		✔		✔	
WRL Informática Ltda.	São Paulo	Rua Santa Ifigênia, 211/213, Box 02– Centro – CEP: 01207-001	11 3362-1334	www.wrl.com.br	✔		✔		✔	
Systech	Taquaritinga	Rua São José, 1126 – Centro - Caixa Postal 71 – CEP: 15.900-000	16 3252-7308	www.systech-ltd.com.br	✔	✔			✔	
2MI Tecnologia e Informação	Embu	Rua José Bonifácio, 55 – Jd. Independência – SP CEP: 06826-080	11 4203-3937	www.2mi.com.br		✔	✔		✔	✔

Calendário de eventos

Evento	Data	Local	Website
Bossa Conference	08 a 11 de março	Porto de Galinhas, PE	www.bossaconference.org
FISL	24 a 27 de junho	Porto Alegre, RS	www.fisl.softwarelivre.org
Interop	02 e 03 de setembro	São Paulo, SP	www.interopsaopaulo.com.br
Futurecom 2009	13 a 16 de outubro	São Paulo, SP	www.futurecom2009.com.br

Índice de anunciantes

Nerdson – Os quadrinhos mensais da Linux Magazine

PREVIEW

Na Linux Magazine #52

DESTAQUE

Administração virtual

Com a rapidez da evolução da Tecnologia da Informação, a virtualização já é notícia antiga. Gerentes de TI em todo o mundo já planejam estratégias para empregar a virtualização e desfrutar de seus inúmeros benefícios. Depois de experimentar todas as principais ferramentas – Xen, VMware, KVM, VirtualBox –, você já está pronto para o próximo passo.

A **Linux Magazine** 52 vai apresentar algumas técnicas e ferramentas muito interessantes para instalar, configurar e gerenciar ambientes virtuais. Você conhecerá a distribuição Rocks pra clusters virtuais usada pela NASA, aprenderá a usar o *Cobbler* no provisionamento da infraestrutura virtualizada e verá como usar o VMware Studio e o SUSE Studio para criar suas próprias *appliances* virtuais. Pode acreditar, a virtualização sempre pode ser mais interessante. ■

PROGRAMAÇÃO

Processador Cell

O revolucionário processador Cell chama a atenção onde quer que vá. Sua arquitetura bastante diferente dos tradicionais chips *x86*, mas também distante dos Power da IBM, coloca-o numa categoria diferente de qualquer coisa jamais vista. Porém, hoje já temos produtos tanto para consumidores finais (o Playstation 3 da Sony e o laptop Qosmio da Toshiba) quanto servidores *blade* que usam a nova tecnologia.

Mesmo que você não tenha o prazer de dispor de um processador Cell, a IBM distribui um SDK gratuito para a arquitetura, que inclui um emulador capaz de rodar em x86, x86-64 e PowerPC. Então, veja como usar os PPEs e SPEs dessa máquina hiper-veloz para processar seus dados a uma velocidade estonteante. ■

Na EasyLinux #14

Desktop seguro

Muitas pesquisas recentes com usuários Linux vêm constatando que a segurança é um dos principais motivos para a adoção do sistema aberto. Infelizmente, simplesmente usar um sistema seguro como o Linux não é suficiente para garantir a segurança na grande rede. Por outro lado, reforçar a segurança até o nível de paranóia costuma comprometer seriamente a conveniência da navegação, além de ser muito trabalhoso. Veja as medidas que qualquer usuário pode (e deve) tomar para aumentar a segurança do sistema Linux. ■

A melhor parte de todos os sistemas

Empresas como Apple e Microsoft investem pesado em design para deixarem seus sistemas mais atraentes. Como resultado, tanto o Mac OS quanto o Windows Vista têm forte apelo visual. Porém, engana-se quem pensa que é impossível alcançar um grau de beleza semelhante no Linux. Na Easy Linux 14, vamos mostrar o caminho das pedras para deixar seu Linux com a cara do Mac OS X ou do Vista, seja por pura diversão, seja para facilitar o uso do Linux por quem já está habituado a esses sistemas e tem dificuldade de adaptação ao pinguim. ■

Primeiro Super-Computador Híbrido da Europa

25 TB de Memória Principal, 1.000 TB de Armazenamento gerenciados através do Lustre®

O primeiro da Europa

Potência total acumulada de 295 Teraflops

Produção
8544 Núcleos (CPU's) Intel®
103 Teraflops

Pesquisa
46 080 Núcleos (CPU's) NVIDIA
192 Teraflops

BULL
Architect of an Open World™