UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UM MODELO DE PLATAFORMA DE DESENVOLVIMENTO DE SISTEMAS DE COMPUTAÇÃO EMBARCADOS UTILIZANDO SOFTWARE LIVRE

FLORIANÓPOLIS, NOVEMBRO DE 2003

MARTA ADRIANA DA SILVA CRISTIANO

UM MODELO DE PLATAFORMA DE DESENVOLVIMENTO DE SISTEMAS DE COMPUTAÇÃO EMBARCADOS UTILIZANDO SOFTWARE LIVRE

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação da Universidade Federal de Santa Catarina.

Orientador: Prof. João Bosco da Mota Alves

FLORIANÓPOLIS, NOVEMBRO DE 2003

TERMO DE APROVAÇÃO

MARTA ADRIANA DA SILVA CRISTIANO

UM MODELO DE PLATAFORMA DE DESENVOLVIMENTO DE SISTEMAS DE COMPUTAÇÃO EMBARCADOS UTILIZANDO SOFTWARE LIVRE

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação na Área de Concentração de Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____
Prof. Raul Sidnei Wazlawick, Dr

Banca Examinadora

_____
Prof. João Bosco da Mota Alves, Dr

_____
Prof. João Candido Lima Dovicchi, Dr

_____
Prof. Benedito René Fischer, Dr

Dedico este trabalho a Deus que me acompanha e protege e quem permitiu que eu pudesse vencer mais esta etapa na minha vida.

Ao meu esposo Alcemir F. Cristiano, que tanto colaborou para o término deste trabalho e aos meus pais, incansáveis parceiros na jornada em busca do conhecimento.

E em memória de Jorge Amiltom de Medeiros Machado, Primo no sangue, mas irmão amado e amigo no coração, onde sempre esteve e para sempre estará.

Meus agradecimentos ao orientador, professor João Bosco da Mota Alves, que com paciência contribuiu com o desenvolvimento deste trabalho, e aos colegas do RexLab, especialmente ao Maurício de Paula e Luiz Rodrigues Maia Neto, que colaboraram imensamente com a realização deste.

À Muriel de Fátima Bernhardt, amiga e companheira de todos os momentos alegres e difíceis dessa caminhada em busca do conhecimento.

À Andréia Miranda, que com carinho acolhe a todos, esbanjando seu carisma, e a todos os amigos conquistados no período de desenvolvimento deste trabalho.

SUMÁRIO

LISTA DE FIGURAS

RESUMO

Sistemas de computação embarcados, presentes em equipamentos como Microondas, telefones celulares, automóveis, naves espaciais, etc., já fazem parte do cotidiano. Especialmente nas soluções dedicadas a aplicações simples e sem conectividade, onde o hardware é projetado para resolver uma aplicação específica, e o sistema operacional e todo o software aplicativo, auto-suficientes, possuem interface de usuário limitada: celulares, microondas, etc.

Com o crescimento da Internet, estes sistemas começaram a crescer em sua complexidade. Para resolver problemas clássicos de sistemas embarcados, passou a ser exigido dos desenvolvedores uma atenção maior à conectividade, para enviar e receber dados ou prover um método automatizado para atualizar as versões de software. Uma das principais preocupações no desenvolvimento em soluções embarcadas está na escolha de um hardware que seja compatível com cada nova versão.

Com o rápido desenvolvimento do setor de alta tecnologia, soluções de hardware e software devem ser flexíveis o bastante para satisfazer as necessidades da constante evolução de mercados e clientes. Dessa forma, a união de hardwares com a utilização de softwares de códigos abertos tornaram-se o mais viável, no que diz respeito a essas novas especificações com custos mais baixos.

O objetivo desse trabalho é apresentar um modelo de plataforma de desenvolvimento de

sistemas de computação embarcados com software livre, comparando-o com sistemas comerciais atuais, e visando baixar custos com sistemas similares, como CYCLADES ou FREESCO desenvolvido em LINUX, ou ainda JUNIPER ou PicoBSD desenvolvido em FreeBSD.

Um exemplo de aplicação utilizando equipamentos atualmente considerados obsoletos em junção com códigos livres, que apresentam funcionalidade tal, capaz de ter seu código fonte reduzido a ponto de rodar a partir de um disquete, dispensando o HD, ilustra a viabilidade do modelo proposto.

ABSTRACT

Computation systems embedded, presents in equipments as Microwaves, cellular telephones, automobiles, spaceships, etc., they are already part of the daily. Especially in the solutions dedicated to simple applications and without connectivity, where the hardware is projected to solve a specific application, and the operating system and the whole software application, self-sufficient, they possess limited user interface: cellular, microwaves, etc.

With the growth of Internet, these systems began to grow in your complexity. To solve classic problems of embedded systems, it became demanded of the desenvolvedores a larger attention to the connectivity, to send and to receive data or to provide an automated method to update the software versions. One of the main concerns in the development in embedded solutions is in the choice of a hardware that is compatible with each new version.

With the fast development of the section of high technology, hardware solutions and software they should be flexible enough to satisfy the needs of the constant evolution of markets and customers. In that way, the hardware union with the use of software of open codes became the viable, in what he/she tells respect the those new specifications with lower costs.

The objective of that work is to present a model of platform of development of

computation systems embedded with free software, comparing him with current commercial systems, and seeking to lower costs with similar systems, like CYCLADES or FREESCO developed in LINUX, or JUNIPER or PicoBSD still developed in FreeBSD.

An application example using equipments now considered obsolete in junction with free codes, that they present such functionality, capable to have your code source reduced to the point of to run starting from a diskette, sparing HD, it illustrates the viability of the proposed model.

1 INTRODUÇÃO

Os sistemas embarcados estão presente em todas as áreas do nosso cotidiano. A grande maioria dos microprocessadores fabricados no mundo atualmente, são usados em dispositivos chamados de “Embedded Systems” ou Sistemas Embarcados, que utilizam microprocessadores e softwares para seu controle, tais como os telefones celulares, caixas automáticos, veículos, aviões, automação industrial e comercial, equipamentos médicos e em uma infinidade de espécies de equipamentos e dispositivos.

Em um sistema embarcado clássico o hardware é designado para resolver uma aplicação específica. O sistema operacional era desenvolvido internamente e todo o software era auto-suficiente havendo porém uma interface de usuário limitada. É o caso dos celulares, microondas, etc.

Com o crescimento da Internet, os sistemas embarcados também começaram a crescer em sua complexidade. Para resolver problemas de sistemas embarcados, os desenvolvedores desses sistemas tiveram que se preocupar com a conectividade para enviar e receber dados ou prover um método automatizado para atualizar as versões de software. Dessa forma, ouve um aumento considerável ao esforço despendido por parte dos desenvolvedores desses sistemas, forçando-os a

avaliar a união dos hardwares de cada aplicação com a utilização de software de códigos abertos.

As exigências de desempenho somadas ao aumento de escalabilidade à lista de requerimentos desses sistemas, limitam a escolha de soluções computacionais. Uma das principais preocupações no desenvolvimento em soluções embarcadas está na escolha de um hardware que seja compatível com cada nova versão. Com o rápido desenvolvimento do setor de alta tecnologia, soluções de hardware e software devem ser flexíveis o bastante para satisfazer as necessidades da constante evolução de mercados e clientes.

Sistemas Embarcados têm várias características comuns:

1) funcionamento simples: Um sistema embarcado normalmente executa um único programa, repetidamente. Por exemplo, um pager sempre é um pager. Em contraste, um PC executa uma variedade de programas, como planilhas eletrônicas, processadores de textos, jogos e vídeos, somados freqüentemente a novos programas.

2) limitações: Todo sistema embarcado deve ser especialmente pequenos. A classificação é segundo o tamanho, desempenho e consumo de energia. Sistemas embarcados têm que ter baixo custo, ser pequeno e ter processamento rápido além de consumir o mínimo de energia.

3) Execução em Tempo Real: Muitos sistemas embarcados têm que reagir continuamente com mudanças no ambiente, e tem que responder dentro de um determinado tempo. Por exemplo, o controlador de um carro monitora continuamente, e reage, a aceleração e aos sensores do freio. Ele tem que calcular a aceleração ou desaceleração repetidamente dentro de um tempo limitado; um resultado atrasado poderia resultar em uma falha para manter controle do carro. Em contraste, um PC não necessita trabalhar com tal velocidade para mantê-lo em funcionamento. Esta demora pode tornar-se inconveniente para o usuário mas não resulta em uma falha do sistema.

Os sistemas embutidos adquirem características cada vez mais complexas, abrangendo conectividade, poderoso gerenciamento, configurabilidade facilitada e visando ainda baixos custos. Diante disso, é preciso a utilização de sistemas operacionais confiáveis e de custos baixos ou “free”, o que gera uma grande procura por sistemas open source.

Por isso, este trabalho empenha-se em aprofundar conhecimento em sistemas de código aberto, de modo a justificar sua empregabilidade em sistemas embutidos.

2 DO UNIX AOS SISTEMAS ABERTOS

Para que se possa compreender a origem do software livre, torna-se necessário antes conhecer um pouco da história do UNIX.

A primeira versão UNIX foi desenvolvida em 1969 por Ken Thompson do grupo de pesquisas dos laboratórios Bell, com a participação de Dennis Ritchie[1], afinal, toda a organização básica como sistema de arquivos, Shell, processos, entre outros, tinham características diretas do MULTICS[2].

Thompson e Ritchie fizeram várias atualizações do sistema operacional em linguagem de máquina antes de reescreverem o código do novo sistema em C[3]. O UNIX, além de fornecer poder e flexibilidade, é simples e consistente, permitindo aos seus usuários compartilhamento tanto de hardware (memória, processadores, discos, impressoras) quanto de software (projetos grandes

[1] Ritchie já havia trabalhado no projeto MULTICS e por isso, tinha grande conhecimento no novo sistema operacional. [CEV – 03]

[2] Multics (Multiplexação de Informação e Serviço Computacional) um sistema operacional, baseado em compartilhamento de tempo, criado em 1965 e ainda em uso hoje. O sistema foi criado como um projeto conduzido pelo Prof. F. J. Corbato do MIT e o Bell Telephone Laboratories. [VLE – 18]

[3] Foi a linguagem de programação C que tornou o UNIX um sistema portável, gerando todo o seu sucesso.

de softwares, com a contribuição de muitos programadores).

No início dos anos 70, o sucesso do UNIX deu origem a inúmeras versões, desenvolvidas por grupos de programadores experientes, conforme as suas necessidades[4]. Com o tempo, estas versões personalizadas – e otimizadas - começaram a ser comercializadas por seus desenvolvedores.

Entretanto, no início dos anos 80, a funcionalidade do UNIX no comércio e nos negócios, começou a ser questionada. Segundo Asceno e Santos, foi com a contínua invasão aos mercados e ao controle de sistemas de interfaces pelas grandes companhias, que um grupo de fornecedores desenvolveu o conceito de “Sistemas Abertos” [ASC - 01].

Este conceito Open Source criou uma base estável e um ressurgimento do entusiasmo pela filosofia do UNIX.

Lentamente, muitos dos softwares já desenvolvidos são integrados ao trabalho de muitos destes grupos do UNIX. Ambientes completos como SunOS, Solaris, são exemplos dessa integração, assim como aplicações como compiladores.

Neste contexto, surge o *BSD. Apresentando agilidade, este sistema apresenta funcionalidade como a SunOS, e ainda sua licença permite sua redistribuição como software proprietário.

E mais adiante, Linus Trovalds, cria o “free Minux”, com centena de desenvolvedores envolvidos, o software é integrado ao GNU. Quase todas as aplicações criadas para Linux, estão

[4] Alguma semelhança com o atual Linux?

sob o GPL[5] – deve ser redistribuído com código fonte. O sistema tem um kernel porém muitas distribuições (Slackware, Debian, RedHat, Suse, etc.).

Apesar de, nem GNU/Linux nem FreeBSd terem dados específicos sobre número de usuários, programadores, companhias, presença em segmentos de mercado, entre outros dados que normalmente interessariam aos seus criadores, as vantagens dessa nova industria de software é evidente. Entre elas, GNU/Linux e FreeBSD competem com Windows-NT, entram em Universidades, na casa dos estudantes, e principalmente, em muitos ambientes onde a melhor escolha é a de custo mais baixo (ou free): Apache, Xfree86, GCC ou GNAT. Os Sistemas Abertos além de apresentarem baixo custo, contêm um vasto leque de aplicações e competitividade de iguais condições com sistemas proprietários.

O software livre gera a idéia de gratuidade, entretanto é preciso ter em mente que apesar de ser disponível livremente, há ainda os custos de estudos, infra-estrutura, manutenção e instalação, mesmo que sua distribuição se de pela internet ou CD_ROM que é relativamente barato. Cabe aqui um parêntese para quebrar um dos mitos mais comuns que permeiam o software livre, a idéia de gratuidade provêm da palavra “Free” que em inglês tem duas traduções: grátis ou livre. Na verdade, um sistema open source, necessariamente deve conter junto de si, seu código fonte, mas daí sua redistribuição ser de graça é outra coisa. Um sistema open source pode sim ser cobrado, e caso alguém queira pagar por ele, tem o direito de requerer seu fonte. Além disso, existe outras formas de arrecadar fundos para desenvolvedores de software livre, como cobrar pelo suporte e assessoria ao usuário que utiliza o sistema.

Os softwares, entretanto, apresentam reusabilidade, transportabilidade, adaptações para vários ambientes, confiabilidade e transparência (facilitando a criação de novos programas). A

[5] Veja subcapítulo 2.2.2 GPL (GNU Public License), pg 22.

qualidade é também evidente nesses softwares[6] levando-se em consideração que o usuário pode depura-lo e ainda ajudar em seu desenvolvimento.

2.1 CONCEITO DE SOFTWARE LIVRE

Com a origem dos sistemas abertos, outra discussão veio à tona na área computacional: o que caracteriza um sistema aberto e os seus códigos "livres".

Existem muitas categorias e variações. Por exemplo, é possível afirmar que ter um software a sua própria disposição, melhorando-o, aumentando sua funcionalidade e ajustando-o segundo sua própria necessidade define o termo em questão. Ou ainda, poder redistribuir o software a outros (com o seu código fonte, é claro) que poderá ser gratuito ou cobrado pelas melhorias efetuadas ou dispor dele de acordo com suas necessidade também pode representar tal definição.

O código fonte de um programa, normalmente escrito em uma linguagem de programação de alto nível, é absolutamente necessário para a compreensão de seu funcionamento, para modificações e melhoramentos. Se um programador tiver acesso ao código fonte de um programa, ele pode estudá-lo, adquirir conhecimento de todos seus detalhes, e trabalhar com ele como o próprio autor original faria.

Ironicamente, para garantir esta liberdade é necessário "proteger" este software com uma licença que imponha certas restrições no modo de usar e redistribuí-lo.

---

[6] BARAHONA chega a fazer um comentário em seu site, dizendo que "Talvez nós nunca usaríamos algum software proprietário se nós pudéssemos ver seu código de fonte". [BAR – 02].

Dal Pont e Cristiano, em sua monografia cujo teor defendia o uso livre de criações intelectuais, em uma crítica a atual legislação, acrescentam:

> "O equilíbrio entre os interesses envolvidos no uso restrito de bens de informação (que tem controles bem definidos) e o uso livre desses mesmos bens, baseada na utilização e compartilhamento de grande número de informações está em jogo" [PON – 14] pg. 24.

Há entretanto opiniões de que o software só terá garantia de ser "free" se limitar os modos de uso e de distribuição.

2.2 LICENÇAS

Atualmente, a licença sob o qual um programa é distribuído define os direitos que seus usuários têm sobre ele. Por exemplo, na maioria dos programas proprietário a licença retira os direitos de copiar, modificar, emprestar, alugar, usar em várias máquinas, etc. na realidade, as licenças normalmente especificam o que o proprietário do programa permite ao seu comprador.

No mundo do software livre, sua licença normalmente especifica as condições que estabelecem um compromisso entre vários objetivos:

- liberdade para redistribuição, modificações, utilização, etc.
- Assegura algumas condições impostas pelos autores (como citar o nome dos autores nos trabalhos derivados, por exemplo).
- Compromisso de que trabalhos derivados também serão software livre.

Os autores podem escolher proteger seus softwares com licenças diferentes de acordo com o grau com que eles querem que seus objetivos sejam cumpridos, e os detalhes que eles querem assegurar. Então, o autor de um programa normalmente escolhe muito cuidadosamente a

licença sob o qual será distribuído. E os usuários que redistribuem ou modificam esses software, têm que estudar sua licença cuidadosamente.

Embora cada autor possa usar uma licença diferente para seus programas, o fato é que quase todos softwares livres usam uma das licenças mais habituais (GPL, LGPL, Artisitic, BSD-like, Netscape- like, etc.)[7], às vezes com leves variações. Para simplificar isto, algumas organizações estão desenvolvendo "trademarks" que garante a todo software protegido sob suas licenças condições de fácil entendimento. Um exemplo notável de disto é a marca registrada Open Source.

2.2.1 BSD (Berkeley Software Distribution)

A licença do BSD eram liberadas pelo CSRG(Computer Science Research Group) da Universidade da Califórnia em Berkeley, sob o sistema operacional UNIX. Porém, como UNIX era proprietário, durante muitos anos, usuários de liberações BSD precisavam ainda de uma licença UNIX para utilizá-lo.

A liberação do BSD serviu de base para muitos sistemas operacionais proprietários como SunOS ou DEC's Ultrix.

Em meados de 1990, o CSRG fez muitas reivindicações para permitir a liberação sem nenhum código proprietário, resultando na autorização dos donos do UNIX pela liberação do BSD-lite[8] (reconhecido por não ter nenhum código proprietário). Esta liberação deu origem ao NetBSD, FreeBSD e ao OpenBSD.

---

[7] Vide anexos.

[8] A liberação do BSD-lite foi o último trabalho feito pelo CSRG antes de sua desintegração. [BAR – 02]

A licença BSD[9] é um ótimo exemplo de licença permissiva. Não impõe quase nenhuma restrição quanto ao que o usuário pode fazer com o software. Os principais pontos da licença são:

- São permitidas redistribuição, uso e modificações no software.
- Distribuições devem incluir cópias da licença, copyright e retratação.
- Reconhecimento das origens do software (Universidade de Califórnia) deve ser incluído em qualquer material de propaganda.

Em suma, pode-se fazer qualquer coisa com o software desde que inclua os dados dos seus criadores. Isto garante o marketing dos autores.

É importante notar que este tipo de licença não inclui qualquer restrição para garantir que os trabalhos derivados sejam livres. Na realidade, muitos sistemas operacionais derivados de liberações de BSD foram distribuídos como software proprietários[10].

2.2.2 GPL (GNU Public License)

A GNU Public License (GPL) é a licença sob o qual o software do projeto GNU é distribuído. Porém, hoje é possível achar toneladas de software sem conexão com o projeto GNU, mas distribuídos sob a GPL (um exemplo é o kernel do Linux). O GPL foi projetado para promover a produção de software free, e por causa disso proíbe algumas ações explícitas no software que poderia conduzir à integração de software GPL em programas proprietário.

A GPL usa como base legal a legislação em direito autorais, fazendo uso disso. Usa os direitos autorais a fim de que sejam usados para promover a distribuição de software garantindo

[9] Vide Anexos: Licença BSD. Pg 68.

[10] há muita discussão sobre esta licença pelo fato de suas cláusulas não contribuírem para o desenvolvimento de software livre quando permitem até mesmo sua própria redistribuição como produto proprietário. [BAR – 02].

liberdade ao usuário.

As principais características da GPL são:

- Permite a redistribuição, mas só se a disponibilização do código fonte também estiver garantida.
- Permite redistribuição do fonte (e obriga isto no caso de distribuição binária).
- Permite modificação sem restrições (e o trabalho derivado também deve estar sob a GPL).

2.2.3 LGPL (Library GPL)

A GNU Library General Public License (LGPL) foi projetado pela Free Software Foundation para proteger algumas bibliotecas que estavam sendo desenvolvidas pelo projeto GNU, mas que deveriam se unir a programas proprietários. Entre suas características:

- Proteger bibliotecas do projeto GNU e muitas outras;
- Projetado para permitir o uso de bibliotecas públicas através de software proprietário;
- Age como o GPL quando a biblioteca é redistribuída como tal.
- Permite a integração com qualquer outro software. Neste caso, não há quase nenhuma limitação.

Há ainda muitas outras licenças de software livre: Artistic (Perl, similar ao BSD) ou NPL (Netscape Public License, inclui certos privilégios para o “primeiro autor”).

3 LINUX

Linux é um Unix Freeware, que segue o mesmo padrão que os sistemas Unix. O Linux foi feito por Linus Torvalds[11], que depois de muito trabalho e motivação, conseguiu lançar o kernel 1.0. Entretanto, Linus Torvalds concluiu que precisava melhorar sua criação e que sem ajuda seria praticamente impossível progredir muito com aquele Linux, então ele pediu ajuda aos programadores através da Internet. Milhares de "hackers" e programadores ajudaram-no a fazer o sistema.

Em 5 de outubro de 1991 Linus Torvalds anunciou a primeira versão "oficial" do Linux, versão 0.02. Desde então muitos programadores têm respondido ao seu chamado, e têm ajudado a fazer do Linux o Sistema Operacional que é hoje. Muitas universidades, servidores e provedores estão usando com sucesso o Linux.

[11] O nome Linux vem da união do nome de seu criador com a palavra Unix (Linus + Unix = Linux).

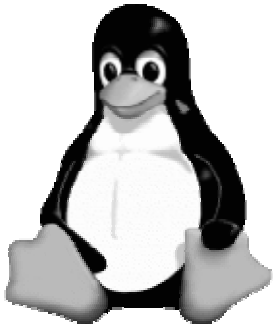

Figura 1 - Logo do Linux

O Kernel é o núcleo do sistema, o que gerencia a memória, que define qual sistema de arquivos o sistema operacional usa e como deve se comportar. Para um sistema funcional, só se precisa do kernel, as outras coisas são complementos, pacotes e aplicativos.

Fabricantes de programas estão cada vez mais interessados em colocar seus softwares para a plataforma Linux. Um exemplo é a Corel, que lançou a versão do WordPerfect para Linux.

A configuração necessária para rodar Linux é a família 386/486/586. Atualmente já existem plataformas como Linux para Macintosh, também. O espaço que o Linux usa é de no mínimo 10 a 30 MB, quem vai usar esse espaço é o Kernel, onde é possível testar o Linux e o sistema de arquivos que ele opera. Entretanto é importante salientar que quanto mais espaço, melhor o desempenho. A memória RAM deve ser de 4MB para cima[12].

3.1 LINUX EMBARCADO

[12] A memória RAM pode ser de até 2 MB, mas isto vai limitar consideravelmente seu desempenho. Quanto mais memória, mais programas poderá rodar no Linux. Em vista disso, o ideal para ter uma rapidez considerável, é 32 MB. O Linux suporta também memória virtual (swap).

Inegavelmente, o sistema operacional LINUX apresenta vantagens, tecnológicas e principalmente financeiras, que impulsionam desenvolvedores a optarem por ele em seus projetos de sistemas embarcados.

Segundo Bill Weinberg [WEI-17], Linux oferece grandes vantagens a embedded systems, como apoio de hardware, escalabilidade, desempenho excelente, confiabilidade e código aberto, sem contar as razões financeiras. Isto possibilita, principalmente, o desenvolvimento de sistemas para automação industrial que utilizam o máximo da capacidade de seu hardware, pois é possível dimensionar e otimizar o sistema operacional para a aplicação que será utilizada. Além disso, é um sistema que suporta um grande número de arquiteturas de hardware, rodando desde supercomputadores até PDAs oferecendo liberdade para a escolha da plataforma de modo que atenda os requisitos do projeto, levando em conta aspectos como: consumo de energia, desempenho e custo.

Quando se quer trabalhar com linux, basta somente escolher os componentes necessários para a aplicação em desenvolvimento, shells e utilitários, deixando-o bem compacto. Além disso, é importante lembrar que grande parte do custo final de um sistema embutido refere-se ao desenvolvimento do seu software, levando a uma freqüente procura por sistemas open source.

Entre as companhias mais conhecidas que utilizam Linux para sistemas embutidos pode-se citar[13]:

- BR MultiAccess da BR Connection;
- Internet Express da Gruponet;
- Projetos Cyclades;
- TopLinux da Topcomm.

[13] Dados divulgados em [NAK - 13].

3.1.1 FREESCO

O projeto FREESCO é um roteador Linux capaz de "substituir" os roteadores da CISCO, foi desenvolvido minimizando o sistema operacional e barateando os custos usuais da CISCO, visando facilidade de instalação e funcionalidade. O nome deriva do termo ciSCO FREE, ou Cisco Livre.

O FREESCO pode ser instalado em um disquete ou ainda no disco rígido de PC's 386 ou mais. Em seu requerimento mínimo está: 386sx ou maior, 8Mb RAM, HD, modens (sendo que HardModem e WinModems não são suportados) e conhecimentos em TCP/IP.

O roteador FREESCO, suporta até 10 redes (com 10 nic's ou com placas de rede de multi-port), e até 5 impressoras (sendo que 2 necessariamente devem ser seriais). Segundo o manual, suporta também até 10 modens, mas somente 4 regulares. E para instalação total na RAM, são necessários, no mínimo, 17Mb.

Além disso, permite DHCP (Dynamic Host Configuration Protocol), DNS, Print (Servidor de impressão), Telnet server, FTP server (File Transfer Protocol), protocolo PPP e um kernel experimental para jogos de redes.

3.1.2 Cyclades

O Cyclades–NL1000 VPN Router é um roteador multiprotocolo, compacto, fácil de instalar e com alta performance. Baseado em Linux, ele agrega poderosos recursos de segurança com a flexibilidade dos sistemas abertos.

O Cyclades-NL1000 é ideal para aplicações de internet e extranet. Ele pode ser

configurado para operar com dados criptografados, tanto em transmissão de dados, quanto nas funções de gerenciamentos e administração.

Figura 2 - Cyclades NL-1000[14]

Para garantir interoperabilidade, foi adotado protocolos padrões como IPSec, SSHv.2, SMNOv.3 e HTTPS. Recursos de firewall e de autenticação permitem configurações alinhadas às políticas de segurança segundo a empresa que o adquire.

Sua configuração de hardware é estabelecida em CPU de 50Mhz, Flash de 16 Mb e memória de 128 Mb SDRAM. Uma interface LAN Ethernet 10/100BT, uma interface WAN serial síncrona até 2Mbps e uma interface serial para dial-backup.

Tem implementados os seguintes protocolos:

- Rede: IP;
- Transporte: TCP, UDP;
- Enlace: Frame Relay, PPP síncrono e HDLC, IP Bridge;
- Roteamento: RIP, RIP II, OSPF, DNS Client, Proxy ARP, PPP/Frame Relay,…
- Gerenciamento: SNMP v.1, v.2, v.3, MIB I e II;
- Aplicação: Telnet Client e Server, FTP, ICMP e NTP.

[14] Detalhe no selo “Linux Inside”.

E seu valor de mercado gira em torno de R$ 10.000,00.

3.2 CRIANDO SUA PRÓPRIA DISTRIBUIÇÃO LINUX

Criar uma distribuição personalizada permite tanto minimizar o possível o sistema, como é o caso dos projetos embarcados, quanto maximizar de forma a facilitar o uso do sistema segundo a necessidade do usuário ou desenvolvedor.

Existem várias maneiras de criar uma distribuição personalizada, entretanto esta é baseada no livro "Entendendo e Dominando o Linux" de Carlos E. Morimoto [MOR – 12].

Para se criar uma instalação padrão pode optar por duas alternativas: o DD (comando que permite fazer cópia exata de um HD para outro) ou G4U (idem ao DD só que trabalha via rede).

Usando o DD, é possível fazer cópias diretas ou criar e restaurar imagens. A sintaxe do DD é "dd if=origem of=destino". No caso de dois HDs por exemplo, faça:

 # dd if=/dev/hda of=/dev/hdc[15]

e estará clonando o conteúdo do primeiro HD para o segundo.

A cópia é feita bit a bit, sendo assim ela é completa (com tabela de partição e setor de boot), independente do sistema operacional.

[15] No Linux, /dev/hda e /dev/hdb são o master e o slave da IDE primária, e o /dev/hdc e /dev/hdd são o master e o slave da IDE secundária.

Para salvar uma imagem faça:

 # dd if=/dev/hdc of=imagem.img

este comando salvará dentro do arquivo imagem.img, todo o conteúdo do **hdc**. Para restituí-lo ao sistema, basta inverter o comando:

dd if=imagem.img of=/dev/hdb

Usando o G4U[16] ("Ghost for Unix") é possível salvar ou recuperar imagens a partir de um servidor de FTP. Ele roda a partir de um disquete.

Para configurar um servidor é preciso habilitar um servidor DHCP (pode ser o compartilhamento de conexões do Windows ou o serviço DHCPD do Linux). Depois disso, deve-se criar uma conta[17] chamada "install" que é usada pelo G4U.

No caso do Linux, basta adicionar o usuário no sistema da seguinte forma:

adduser install (cria o usuário)

passwd install (define a senha)

e o diretório será o /home/install. O G4U detecta automaticamente placas de redes instaladas no cliente[18], durante o boot, assim como obtêm o endereço IP.

---

[16] Ghost for Unix é uma mini distribuição do NetBSD que complementa o DD.

[17] A conta criada deve ter permissão de escrita para a pasta que terá gravado o arquivo imagem.

[18] Veja em Anexo "Lista de Placas de Rede Compatíveis".

Para salvar o arquivo imagem no servidor use:

uploaddisk [IP_do_servidor] [nome_do_arquivo.gz] wd0[19]

E para recuperá-lo use:

slurpdisk [IP_do_servidor] [Nome_do_arquivo.gz] wd0.

3.2.1 Criando Uma Instalação Base

Para esta etapa o Slackware é uma boa sugestão, cuja configuração é feita direto no fonte. O ideal é fazer uma instalação mínima do Slack, apenas com os pacotes básicos para o sistema funcionar (aproximadamente 30 Mb). Depois disso, vai se acrescentando os pacotes que se fizerem necessários, tais como:

- **ide**: O Kernel com suporte a interfaces IDE.
- **aoutlibs**: Bibliotecas C utilizadas por vários programas.
- **gpm**: Acrescenta suporte a mouse em aplicativos de modo texto. Útil no lynx, mc e outros programas.
- **isapnp**: Facilita a instalação de placas ISA.
- **Kbd**: Layouts de teclado alternativos (ou seja, todos além do US :-)
- **minicom**:Um pacote com discador e outras ferramentas necessárias para estabelescer conexões via modem e cabo serial. Não é necessário se o PC for acessar a Web via rede.
- **pcmcia**: Inclui suporte a placas PCMCIA, necessário se você pretender usar a instalação também em notebooks. Este serviço fica ativado por default e é capaz de detectar qualquer placa de rede ou modem suportado pelo Kernel 2.2.

[19] "wd0" e "wd1" são os HDs primary máster e primary slave respectivamente, e os "wd2" e "wd3" são secondary máster e secondary slave. No caso de HDs SCSI as identificações são substatuídas por "sd0", "sd1", "sd2" e "sd3".

E algumas ferramentas e protocolos comuns:

- **mc**: Gerenciador de arquivos de modo texto.
- **vim**: A versão aperfeiçoada do Vi. Não é necessário pois o Slackware instala o Elvis, um editor semelhante ao vi por default.
- **manpages**: As páginas de manual, opcional.
- **tcpip1**: O pacote básico do TCP/IP, necessário para conectividade em rede.
- **tcpip2**: Inclui o DHCP, Ipchains, Ipfwadm e outras ferramentas. Não é necessário em caso de IPs estáticos, ou se não for utilizar o PC como roteador.
- **lynx**: O navegador de modo texto.
- **xbin**: O pacote básico do X.
- **xfnts**: O pacote mínimo de fontes obrigatório para rodar o X.
- **xlib**: Inclui bitmaps, arquivos de configuração e algumas bibliotecas necessárias para rodar o X.
- **xsvga**: Inclui o servidor X SVGA, que garante compatibilidade com a grande maioria das placas vídeo, mas com apenas 256 cores.

Após a instalação, deve-se otimizá-lo. Para isto, basta desativar os recursos desnecessários e ativar o suporte a hardware necessário para roda-lo em rede. Para desativar basta acrescentar uma tralha (#) no início de cada linha.

Em primeiro lugar é o arquivo **/etc/inetd.conf** onde estão os serviços de rede[20].

```
# These are standard services.
#
# ftp stream tcp nowait root /usr/sbin/tcpd wu.ftpd -l -i –a
# telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
#
```

[20] O servidor de FTP e telnet ficam ativados por default entre outros serviços.

No arquivo **/etc/rc.d/rc.M** existem serviços que talvez possam ser desativados, dependendo do que se vai trabalhar tais como cron, syslogd, lpd (porta paralela, necessário apenas para impressão), quota, sendmail, APM, GPM, Apache (linha webserver), Samba, etc. Da mesma forma no arquivo **/etc/rc.d/rc.inet2** com os serviços IPV4_Foward (só necessário se você for utilizar o micro como roteador), suporte a NFS (pasta de um servidor remoto), o serviço KLOGD (logs do sistema) e o servidor SSHD.

O suporte a hardware é feito no arquivo **/etc/rc.d/rc.modules**[21].

Se todos os micros da rede usarem placas de vídeo do mesmo modelo, basta configurar uma vez e copiar a imagem para todos os micros. Caso contrário, deve ser feita uma imagem diferente para cada placa, ou copiar a imagem padrão e configurar o **xf86config** micro por micro[22]. Depois disso, configura-se a estação para abrir o X automaticamente durante o boot editando o arquivo **/etc/inittab**, alterando o runlevel padrão de 3 para 4 (e não 5 como em outras distribuições).

Neste ponto já se tem uma rede operante, a partir daí pode se ir otimizando e adicionando recursos ao sistema até chegar ao nível que se deseja. Pastas “man”, “info” ou “doc” ocupam em média 10% do espaço total em disco, assim estas pastas podem ser excluídas. Módulos do Kernel (/bin/modules), programas e comandos (/usr/bin) desnecessários podem ser deletados também.

[21] Veja em anexo “Módulos de Redes Disponíveis”

[22] O Slackware 7.1 utiliza o Xfree 3.3 inclui suporte a quase 800 placas, o que facilita o uso. [MOR – 11].

4 FREEBSD

Desde quando estudantes da Universidade da Califórnia passaram a ter acesso ao código fonte do UNIX, este foi melhorado rapidamente. Partes inteiras do código original do sistema operacional UNIX foram substituídas a fim de construir comandos úteis, controles de programas correntes e um sistema de arquivos que levou o sistema operacional UNIX ao que é conhecido hoje. O CSRG (Computer Systems Research Group)[23] distribuiu este código alterado gratuitamente com a licença válida do UNIX, que veio a ser conhecido como "Berkeley Software Distribution ou BSD UNIX". [LUC – 09]. Depois de quase 15 anos de alterações, sobrou muito pouco do código original do UNIX, e a primeira versão realmente "free" foi liberada com o nome BSD4.4-Lite, da Universidade da Califórnia em Berkeley, e sua atualização BSD4.4-Lite2. Estes foram os avós de muitos sistemas operacionais, não só do FreeBSD, mas do NetBSD, Mac OS X, Open BSD, entre outros.

O FreeBSD é um sistema operacional para plataformas i386 e Alpha/AXP, baseado no 4.4BSD-Lite. A documentação do FreeBSD sobre as perguntas mais freqüentes sobre o sistema,

[23] CSRG – Computer Systems Research Group é o grupo de pesquisas da Universidade da Califórnia em Berkeley.

confirma a origem do FreeBSD segundo Michael Lucas [LUC – 09], quando faz a seguinte afirmação:

> "O FreeBSD também é baseado, indiretamente, na conversão de William Jolitz conhecida como '386BSD' para a plataforma i386 do 'Net/2' da Universidade da Califórnia, em Berkeley, apesar de pouquíssimo código original do 386BSD ainda exista no FreeBSD" [PRO – 10].

Figura 3 - Logo FreeBSD

O principal objetivo do Projeto FreeBSD é oferecer um software disponível a quaisquer pessoas e para qualquer finalidade, de forma que esse código ofereça o maior número possível de benefícios e formas de uso, sem custos e passível de alterações que o usuário julgar necessário[24]. E são estes objetivos que justificam a licença sob a qual o software é distribuído[25]. A licença BSD na verdade não define regras de utilização do software, mas como tratar o projeto FreeBSD ao utilizar o código distribuído por ele.

O nome FreeBSD pode ser entendido se dividido em duas partes:

Free – tanto no sentido de gratuidade quanto no sentido de liberdade (já que o usuário pode adquirir os sistema de graça e tem liberdade de fazer o que quiser com ele).

---

[24] Qualquer alteração efetuada no sistema, desde que respeite as regras da licença BSD, pode ser adicionado à árvore de código fonte do sistema. Mais detalhes em http://www.freebsd.org/.

[25] Veja em ANEXOS, Licença BSD.

BSD – “Berkeley Software Distribution”. Nome que o Grupo de Pesquisa de Ciência da Computação da Universidade de Berkeley – CSRG (Computer Systems Research Group) escolheu para esta distribuição do UNIX.

O FreeBSD destaca-se por oferecer um ambiente robusto e completo para as aplicações. Suporta uma grande gama de navegadores, programas de manipulação gráfica, ambientes de programação para servidores, serviços de redes e muito mais, sendo que a maioria das aplicações podem ser gerenciáveis pela coleção de “Ports”[26] do sistema. E o mais importante, é possível trabalhar sem interrupções.

Entre as principais características do FreeBSD pode-se destacar sua portabilidade – o sistema operacional FreeBSD roda na maioria dos hardwares atuais, como Intel x86-compatíveis (386, 486, Pentium, Celerom e AMD) – e facilidade na configuração dos pacotes de software para o sistema (isto é devido a sua completa documentação). Além destas, vale salientar também:

Preemptive Multitasking – multitarefa preempitiva com ajuste de prioridade dinâmica entre aplicações e usuários, mesmo com sobrecarga.

Multiuser – sistema multiusuário permitindo acesos simultâneos ao sistema.

Strong TCP/IP rede – redes TCP/IP com suporte a SLIP, PPP, NFS, DHCP, e NIS.

4.1 CONFIGURANDO O FREEBSD

[26] Veja em http://www.freebsd.org/ports/.

No FreeBSD assim como no Linux, é possível alterar as configurações do sistema operacional afim de personalizá-lo. Para tanto é necessário conhecer um pouco das opções de configuração.

Começando pelas opções iniciais de boot, temos:

SWAPFILE = “NO”
APM_ENABLE = “NO” - YES se desejar habilitar APM.
PCCARD_ENABLE = “NO” - YES para configurar dispositivos de PCCARD.
PCCARD_MEM = “DEFAULT” - Se pccard_enable=YES, então este será o endereço de memória do cartão.
PCCARD_IFCONFIG = “NO” - Configuração de ethernet de pccard especializada.

local_startup = “/usr/local/etc/rc.d /usr/X11R6/etc/rc.d”

Durante o startup, o FreeBSD executa alguns programas nos diretórios listados no “local_startup”, na ordem em que aparecem na lista. Semelhante ao /etc/rc.local. No caso de diretórios múltiplos, deve-se separá-los por espaços.

Quanto a configuração de rede, as opções básicas são:

hostname = “myname.my.domain”
NISDOMAINNAME = “NO” - caso use NIS, coloque o domínio.
FIREWALL = “NO” - veja /etc/rc.firewall, segundo o objetivo do usuário.
TCP_EXTENSIONS = “YES” - Permite RFC1323 & extensões de RFC1544 (ou NENHUM).
network_interfaces = “lo0” - Lista de interfaces de rede (lo0 é loopback).
ifconfig_lo0 = “inet 127.0.0.1”
ifconfig_lo0_alias0 = “inet 127.0.0.254 0xffffffff” de netmask.
SYSLOGD_ENABLE = “YES”

syslogd_flags = “ ”

INETD_ENABLE = “YES”

inetd_flags = “ ”

NAMED_ENABLE = “NO” - nomeia o servidor de DNS (ou NENHUM).

named_flags = “-b /etc/namedb/named.boot”

KERBEROS_SERVER_ENABLE = “NO”

RWHOD_ENABLE = “NO”

AMD_ENABLE = “NO”

amd_flags = “-a /net -c 1800 i386 de -k -d my.domain syslog de -l /host /etc/amd.map”.

TIMED_ENABLE = “NO”

timed_flags = “ ”

NTPDATE_ENABLE = “NO”

ntpdate_flags = “ ”

XNTPD_ENABLE = “NO”

xntpd_flags = “ ”

TICKADJ_ENABLE = “NO”

tickadj_flags = “-Aq”

NIS_CLIENT_ENABLE = “NO”

nis_client_flags = “ ”

NIS_YPSET_ENABLE = “NO”

nis_ypset_flags = “ ”

NIS_SERVER_ENABLE = “NO”

nis_server_flags = “ ”

NIS_YPXFRD_ENABLE = “NO”

nis_ypxfrd_flags = “ ”

NIS_YPPASSWDD_ENABLE = “NO”

nis_yppasswdd_flags = “ ” .

Quanto ao roteamento tem-se:

DEFAULTSCRIPTS = “NO”

static_routes = “ ” - ou a lista de rota estática.

GATEWAY_ENABLE = “NO” - ou YES se este host for usado como um gateway.

SCRIPTS_ENABLE = “YES” - ou habilitar um daemon scripts.

scripts = “routed” - Nome para o daemon scripts usar.

scripts_flags = “-q”

MROUTED_ENABLE = “NO” - multicast routeamento (veja /etc/mrouted.conf).
IPXGATEWAY_ENABLE = “NO” - ou YES para habilitar IPX scripts.
IPXROUTED_ENABLE = “NO”
ipxrouted_flags = “ ”
arpproxy_all = “ ” - substitui opção de kernel obsoleto ARP_PROXY_ALL.

Quanto ao Console

KEYMAP = “NO” - keymap em /usr/share/syscons/keymaps/*.
KEYRATE = “NO” - teclado: lento, normal, rápido (ou NENHUM).
KEYBELL = “NO”
KEYCHANGE = “NO”
cursor = “NO” - tipo de cursor {normal|blink|destructive} (ou NENHUM).
SCRNMAP = “NO”
FONT8X16 = “NO” - fonte 8x16 de /usr/share/syscons/fonts/*.
FONT8X14 = “NO” - fonte 8x14 de /usr/share/syscons/fonts/*.
FONT8X8 = “NO” - fonte 8x8 de /usr/share/syscons/fonts/*.
BLANKTIME = “NO”
saver = “NO”
MOUSED_TYPE = “NO” - Veja rc.conf(8) para colocações disponíveis.
moused_port = “/dev/cuaa0”
moused_flags = “ ”.

Outras configurações, relevantes conforme o objetivo de uso do sistema:

CRON_ENABLE = “YES”
LPD_ENABLE = “YES” - roda linha de impressão
lpd_flags = “ ” - flags de lpd (se habilitado).
SENDMAIL_ENABLE = “YES”
sendmail_flags = “-bd -q30m”
SAVECORE_ENABLE = “NO”
DUMPDEV = “NO”
CHECK_QUOTAS = “NO”

ACCOUNTING_ENABLE = “NO”
IBCS2_ENABLE = “NO”
LINUX_ENABLE = “NO” - Carrega o emulador Linux.
RAND_IRQS = “NO”.

4.3 PICOBSD

PicoBSD é uma pequena versão do FreeBSD 3.0. O PicoBSD permite ter acesso dialup seguro, um pequeno roteador diskless ou ainda um servidor dial-in.

A configuração mínima para rodar o PicoBSD é um PC 386sx e 8Mb de RAM. Não é necessário um HD, ele cabe todo em 1 disquete.

Como resultado da extrema limitação de tamanho, no PicoBSD não há bibliotecas dinâmicas e não há o diretório /usr/lib. Somente executáveis estáticos são executados. E para reduzir o tamanho dos executáveis, todos os executáveis de um disquete específico são jogados ao mesmo tempo dentro de um único executável.

Além da flexibilidade do FreeBSD, o código fonte é aberto, permitindo que o usuário possa construir uma instalação pequena que executa várias tarefas, tais como, estações de trabalho, acesso dial-up, customizador de disco, controlador embarcado (flash ou EEPROM), firewall, servidor de comunicação, roteador, e outras. Ideal para projetos de sistemas embarcados.

4.3.1 COMO CONSTRUIR UMA VERSÃO PERSONALIZADA DO PICOBSD

Adquira o arquivo picobsd.tgz. Nele contém todos os scripts necessários[27]. Descompacte o arquivo (será necessário 5 Mb aproximadamente).

Entre no diretório (cd build) e rode o script ./build. Selecione a linguagem, tamanho de MFS e um setup pre-instalado (personal dialup, dialin server ou router-like). Os detalhes de cada setup estão contidos nos diretórios dial/, router/, isp/ e net/ respectivamente.

Há vários diretórios que contêm algum fontes e arquivos config:

- build/ diretório principal
- dial/ arquivo de configuração para setup dialup
- conf/ arquivo de configuração do kernel
- crunch1/ programas de sistemas
- mfs.tree/ contêm a configuração de MFS
- lang/ contêm os arquivos de idiomas
- floppy.tree/ contêm a hierarquia de startup em disquete
- isp/ arquivos de configuração para servidor dialin
- net/ arquivos de configuração para scripts-like
- tinyware/ coleçao de utilitários do sistema
- tools/ ferramentas adicionais para a construção de sistema.

O disquete é necessário só durante o startup.

- Edite o set de programas instalados.
- São necessários 9MB de espaço livre em disco, e um diretório /mnt.
- Entre em cd build/ no script ./build. Selecione os parâmetros de construção. Tem-se então um arquivo “picobsd.bin” neste diretório.

[27] Existe o fonte do PicoBSD no repositório do FreeBSD, assim é possível encontra-lo em /src/release/picobsd.

Se houver erro durante o processo, provavelmente será por uma das razões:

- crunchgen não podem achar o diretório para o programa “proggy”: tenha certeza que o diretório fonte para ' proggy' é chamado ' proggy', caso contrário o crunchgen não o achará;
- falha no build;
- confira a árvore de fonte do sistema em .depend;
- veja se os programas individuais que usam Makefiles original podem ser construídos;
- falha de gravação - sistema de arquivo está cheio.

confira se o tamanho de MFS é informado corretamente enquanto estiver montando.

Finalmente, transfira este arquivo ao disquete:

```
dd if=picobsd.bin of=/dev/rfd0.
```

4.4 JUNIPER

A Empresa Juniper Networks foi fundada em 1996[28] e desde esta época apresenta produtos com performance, inteligência e escalabilidade. Entretanto hoje, apresenta soluções de alta tecnologia utilizando softwares livres (principalmente FreeBSD) para projetos como o da Hokkaido Telecomunications Networks Co. (HOTnet), que está transformando sua WAN para 10-Gigabit Multiprotocol Label Switching (MPLS) com os roteadores da juniper Networks [JUN-20].

Para demonstrar a qualidade dos projetos da empresa, é possível avaliar um firewall

[28] www.juniper.net/company/

desenvolvido com FreeBSD[29] que dói disponibilizado por seus criadores. Foi projetado para não remeter pacotes entre interfaces, porém implementa um proxy que permite a máquinas internas ter acesso a Internet como se estivessem diretamente conectados.

Juniper trabalha designando as interfaces de rede como confiáveis e não confiáveis[30]. Seu kernel suporta estas designações e seu uso quanto ao que fazer com as conexões que entrariam na máquina. Juniper pode diferenciar tipos de sessões, e gerar um daemon[31] diferente por sessão. As tentativas para estabelecer uma sessão além das regras pré-estabelecidas no fonte, seja por interface não confiável, seja por tentar um host diferente do firewall, é rejeitado.

O software Juniper possui licença própria que permite seu uso, com ou sem modificações, desde que sua redistribuição retenha os dados autorais anteriores a ultima alteração, e todo material publicitário que mencione características ou uso deste software tem que exibir o seguinte:

> "This product includes software developed by Obtuse Systems Corporation and its contributors.
> This product includes software developed by Eric Young (eay@mincom.oz.au)"(www.obtuse.com/juniper/)

Acrescenta ainda que o código desta licença não pode ser copiado e colocado sob outra licença, incluse GNU Public License[32].

---

[29] Seu código entretanto foi preparado para rodar também em Linux.

[30] O código fonte de Juniper encontra-se em anexo.

31 Um Daemon é chamado para tratar uma sessão não confiável. Um Proxy é chamado para controlar uma sessão capturada. Qualquer coisa que é capaz de ser chamado pode ser usado como um daemon. Proxys devem ser capazes de operar em tempo de execução.

[32] A Juniper Public License está na íntegra em anexo.

4.4.1 CONFIGURANDO O FIREWALL BÁSICO NO FREEBSD

O ipfw pode ser ativado de duas maneiras: usando o sistema de módulos do FreeBSD, ou recompilando o Kernel. Através de módulos é muito simples, basta digitar:

```
modload /lkm/ipfw_mod.o
```

Através da recompilação do Kernel, basta adicionar as linhas abaixo ao arquivo de configuração de seu Kernel.

```
options IPFIREWALL
options IPFIREWALL_VERBOSE
```

É preciso recompilar o Kernel e instalá-lo. Além disso, você precisa habilitar a opção gateway, no rc.conf adicionando gateway_enable=”YES”. No rc.conf você deve adicionar também as instruções abaixo:

```
firewall_enable=”YES”
firewall_script=”/etc/rc.firewall”
firewall_type=”simple”
firewall_quiet=”NO”
```

Desta maneira, basta criar as regras de filtragem. Elas são escritas no *script* rc.firewall, com a seguinte sintaxe:

```
ipfw [-q] add [number] action [log] proto from src to dst
[via name | ipno] [options]
```

No script, o ipfw é substituído por uma variável de sistema denominada ${ipfwcmd}, estabelecida no rc.firewall. Analisando a sintaxe do comando temos a ordem de adicionar (add) um número opcional. Este número está diretamente relacionado à ordem em que as regras são analisadas, sempre começando em 0 (zero) até 65535 (sessenta e cinco mil quinhentos e trinta e cinco). A ação (action) é a operação a ser realizada, como, por exemplo, aceitar ou recusar um

pacote. O proto é o protocolo, como por exemplo, TCP ou UDP. As posições src e dst são os endereços de origem e de destino do pacote, incluídas as portas de origem/destino. A variável via name representa a interface de rede do pacote. A variável options oferece algumas extensões que podem ser utilizadas.

As ações possíveis são:

- allow (*pass, permit, accept*): aceita o pacote;
- deny (*drop*): rejeita o pacote;
- reject: rejeita o pacote e manda de volta um pacote icmp de host unreachable;
- unreach code: rejeita o pacote e manda um pacote icmp de com o código code;
- reset: somente para TCP, rejeita e manda um TCP RST;
- count: incrementa o contador de pacotes desta regra;
- divert port: conduz o pacote para a porta port;
- tee port: faz uma cópia do pacote para a porta port;
- skipto number: pula para a regra number;

Os protocolos possíveis são:

- IP (*all*): todos os pacotes;
- TCP: pacotes TCP;
- UDP: Pacotes UDP;
- ICMP: Pacotes ICMP;

O src e dst são no seguinte formato: address/mask [port]e indica o endereço e a porta. Uma porta pode ser descrita com vírgulas, ou com traço, para indicar uma faixa de portas.

E o address/mask pode ser descrito na forma:

- ipno: somente o IP é combinado com a regra;
- ipno/bits: todos os pacotes da rede indicada pelo número de bits são

combinados;

- ipno:mask: semelhante ao anterior, porém indicando a *netmask;*

É possível ainda adicionar o prefixo not, invertendo a especificação de endereço.

As opções podem ser:

- frag: se o pacote é um fragmento de datagrama;
- in: se o pacote é um pacote de entrada;
- out: se o pacote é um pacote de saída;
- established: se o pacote é um pacote de uma conexão TCP já estabelecida;
- setup: se o pacote é uma requisição de um serviço TCP;

Algumas regras úteis:

$fwcmd add 65353 deny all from any to any: esta regra é ativada sempre, ela recusa qualquer pacote, se quiser abrir o *firewall* totalmente inclui-se:

$fwcmd add 65000 allow all from any to any: como o número é menor esta regra é avaliada primeiro e aceita todos os pacotes. Esta é a configuração de um *firewall* aberto.

É comum criar variáveis para facilitar a configuração, ou adotar as existentes nas configurações exemplo contidas no script rc.firewall:

```
# set these to your outside interface network and netmask and ip
oif=”ed0"
onet=”150.162.79.0"
omask=”255.255.255.0"
oip=”150.162.79.130"
# set these to your inside interface network and netmask and ip
iif=”ed1"
inet=”192.168.3.0"
imask=”255.255.255.0"
```

iip=”192.168.3.17"

Assim pode-se criar as seguintes regras:

$fwcmd add deny all from ${inet}:${imask} to any in via ${oif}

$fwcmd add deny all from ${onet}:${omask} to any in via ${iif}

Estas regras são utilizadas para prevenir o ataque do tipo *IP Spoofing*. *IP Spoofing* é uma técnica usada para que a máquina origem do ataque possa se fazer passar por uma máquina confiável da rede interna. Simplificando, não deve-se receber um pacote da rede interna (${inet}:${imask}) pela interface de rede externa (via ${oif}), assim como não se recebe pacotes da rede externa (${onet}:${omask}) pela interface de rede interna (via ${iif}):

$fwcmd add deny all from 192.168.0.0:255.255.0.0 to any via ${oif}

$fwcmd add deny all from 172.16.0.0:255.240.0.0 to any via ${oif}

$fwcmd add deny all from 10.0.0.0:255.0.0.0 to any via ${oif}

Existem algumas regras usadas exclusivamente para redes internas atrás de *firewall* com tradução de endereços (nat). Uma boa técnica para configurar a rede via pacote tipo TCP é permitir a passagem de todos os pacotes, cujo serviço já foi negociado. Com isso o *firewall* fica muito leve e prático de configurar:

$fwcmd add pass tcp from any to any established

$fwcmd add pass tcp from any to ${oip} 25 setup

$fwcmd add pass tcp from any to ${oip} 80 setup

A primeira regra (acima) habilita das conexões existentes. As duas próximas regras habilitam a chegada de *e-mail* e a chegada de requisições www.

$fwcmd add deny log tcp from any to any in via ${oif} setup - esta regra recusa e registra qualquer outra conexão do tipo TCP feito via interface externa.

$fwcmd add pass tcp from any to any setup - esta regra habilita todas as conexões TCP restantes (ou seja, da rede interna).

Vale ressaltar que a ordem das regras é muito importante. Se forem alteradas as duas últimas regras o *firewall* ficará aberto. Às vezes é necessário criar regras para a rede interna também, como, por exemplo, limitar o uso de um determinado serviço de rede, tais como IRC, salas de *chat*, ICQ etc.

Para regras UDP o modelo é o mesmo, porém como UDP não é orientado à conexão deve-se fazer o filtro de todos os pacotes, mas sem esquecer da resposta à requisição através do protocolo UDP, como por exemplo:

```
$fwcmd add pass udp from any to any 53

$fwcmd add pass udp from any 53 to any
```

O pacote de DNS sai da máquina em uma porta alta direcionada a outra máquina na porta 53, esta, por sua vez, responde em uma outra porta, normalmente acima da alta. Vale lembrar aqui que servidores de DNS secundário realizam a transferência de mapas de domínio via TCP na porta 53, logo deve-se habilitar a porta 53 para setup tcp do servidor de DNS secundário, conforme o exemplo a seguir:

```
$fwcmd add pass tcp from 150.162.1.3 to ${oip} 53 setup
```

$fwcmd add pass tcp from 150.162.1.3 to ${iip} 53 setup

Assim, não importa qual interface de rede, ele poderá fazer a conexão.

Esta regra deve ser colocada junto com as demais regras de TCP.

5 DESENVOLVENDO O PROTÓTIPO

Levando em consideração que este protótipo, antes de mais nada, deve ser reduzido o bastante para permitir outros desenvolvimentos para sistemas embarcados, toda esta plataforma foi baseada no PicoBSD. E como já foi apresentado antes, a conectividade é uma das maiores preocupação na área de Embedded Systems, por isso, o protótipo propõe fazer o PicoBSD – que é Open Source, livre e bastante pequeno – trabalhar como um roteador, demonstrando a viabilidade de comunicação via redes internas e externas. Desta forma, será possível perceber que com mínimas alterações no código de sistemas open source, pode-se desenvolver projetos embarcados bastante completos.

Para que um computador possa encontrar outro numa rede, é necessário um mecanismo que descreva o caminho que deve ser seguido, mostrando um destino e um gateway, este mecanismo é o “Routing” ou Roteador. Os destinos são sempre hosts individuais, subnets, e se nenhum se aplicar, um destino default. Os gateways também podem ser hosts individuais, ou ainda interfaces (links) ou Ethernet (endereços MAC).

Para demonstrar um roteador default, tem-se:

```
% netstat -r
Routing tables

Destination        Gateway            Flags    Refs    Use    Netif    Expire

default            outside-gw         UGSc       37    418      ppp        0
localhost          localhost          UH          0    181      lo0
test0              0:e0:b5:36:cf:4f   UHLW        5  63288      ed0       77
10.20.30.255       link#1             UHLW        1   2421
example.com        link#1             UC          0      0
host1              0:e0:a8:37:8:1e    UHLW        3   4601      lo0
host2              0:e0:a8:37:8:1e    UHLW        0      5      lo0
host2.example.com  link#1             UC          0      0
224                link#1             UC          0      0
```

Onde as primeiras duas linhas especificam um roteador default e uma rota localhost. A interface (Netif) tem uma tabela específica para uso do localhost – lo0, também conhecida como dispositivo loopback, que determina todo o tráfego do fluxo interno.

A próxima linha trata de endereços MAC. O FreeBSD identifica qualquer host automaticamente e relaciona uma rota para aquele host diretamente sobre a interface Ethernet, ed0. A coluna “Expire” é o tempo determinado que a rota pode ficar ociosa à espera de resposta de um host, quando este tempo expira, a rota é apagada automaticamente. Alguns hosts usam o mecanismo RIP (Routing Information Protocol) que determinam a rota pelo caminho mais curto.

O FreeBSD também determina rotas de subnet para a subnet local, Por exemplo: 10.20.30.255 é o endereço de radiodifusão para a subnet 10.20.30, e “example.com” é o nome de domínio associado com aquela subnet. O link#1 refere-se ao primeiro cartão Ethernet da máquina. Tanto hosts da rede local quanto subnets locais tem suas rotas automaticamente configuradas por um daemon chamado routed. Caso contrário, só as rotas que são estaticamente definidas serão utilizadas.

A linha “host1” faz referência ao endereço Ethernet que deseja enviar pacotes. E a linha “host2” é um exemplo do uso do ifconfig[33].

Por fim, o significado de cada símbolo na coluna Flags:

- U - Up: A rota está ativa.
- H - Host: O rota destino é um único host.
- G - Gateway: Envie qualquer coisa para este destino que ele saberá a quem enviar isto.
- S - Static: Rota configurada manualmente, não foi gerada automaticamente pelo sistema.
- C - Clone: Gera uma rota nova baseada nas rotas comuns da máquina. Este tipo de rota normalmente é usado para redes locais.
- W - WASCLONED: Indica uma rota baseado em outra rota da rede de área local (Clone).
- L - Link: Rota que envolve referências de hardware da Ethernet.

Quando há a necessidades de conexão a um host remoto, o roteador confere a tabela de roteamento para determinar se um caminho existe. Se o host remoto está em uma subnet, então checa para saber se pode conectar ao longo daquela interface.

Se todas as rotas falharem, o sistema tem ainda a opção da rota default. Esta rota é um tipo especial de “gateway route” (normalmente único no sistema), e sempre é marcado com um “C” na coluna Flags. Para hosts em uma rede de área local, este gateway pode ser uma máquina com conexão direta para o mundo externo.

Há ainda as configurações de hosts em duas redes diferentes. De duas, uma: ou a máquina tem dois cartões Ethernet, cada um com um endereço de subnet separado, ou tem um

[33] ifconfig é um utilitário usado para nomear um endereço em uma interface de rede ou configurar parâmetros da interface de rede.

cartão Ethernet, e está usando ifconfig. O primeiro caso é usado se duas redes Ethernet estiverem fisicamente separadas, e o segundo se há um segmento de rede física, mas duas subnets logicamente separadas. De qualquer modo, as tabelas de roteamento são fixadas de forma que cada subnet sabe que esta máquina é um gateway para o outra subnet. Esta configuração com a máquina que age como um roteador entre duas subnets, é freqüentemente usada quando se precisa implementar filtros de pacotes ou firewall. Para tanto, isto deve ser habilitado no FreeBSD.

Um roteador de rede é simplesmente um sistema que remete pacotes de uma interface para outra. É possível habilitar esta característica no FreeBSD (que não é default) mudando a seguinte variável em rc.conf[34]:

```
gateway_enable=YES    # Set to YES if this host will be a gateway
```

Agora o novo roteador precisa de rotas para definir o tráfego.

5.1 CONFIGURANDO O FREEBSD COMO ROTEADOR

Usando a interface do ClosedBSD[35] é possível configurar os cartões de rede adequadamente. Para testar o protótipo deste trabalho, o primeiro passo foi copiar o arquivo imagem do ClosedBSD com o seguinte comando: $dd if=closed.bin of=/dev/fd0. Com esta imagem em um disquete, bastar dar um boot por ele. Depois de aparecer a licença BSD, um menu é exibido.

[34] O arquivo rc.conf contém informação descritiva sobre o host local. Configuração detalha da interface de rede e quais serviços devem ter início durante o boot.

[35] ClosedBSD é um firewall e um utilitário NAT e cabe em um único disquete ou CDROM, e não requer disco rígido. ClosedBSD é baseado no kernel do FreeBSD. (http://www.closedbsd.org).

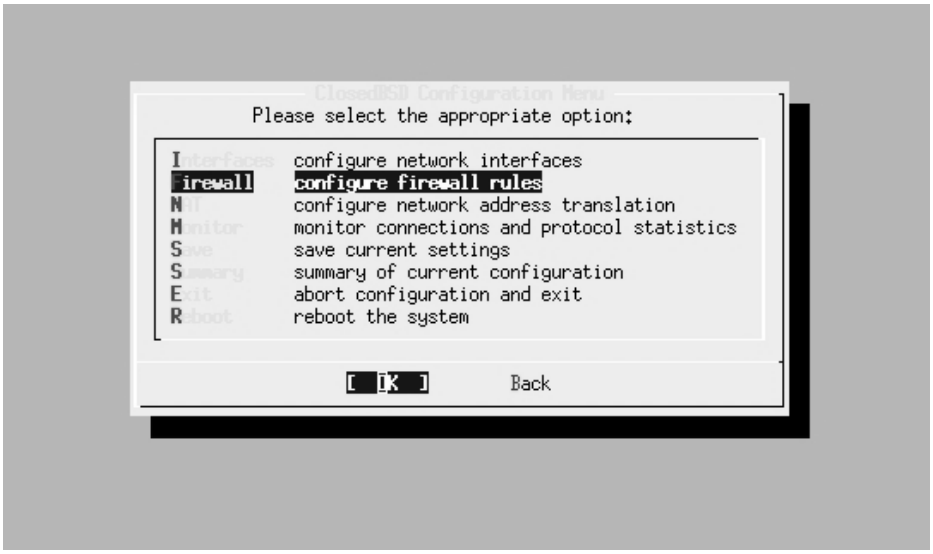

Figura 4 - Menu Principal do ClosedBSD. [SIM – 16]

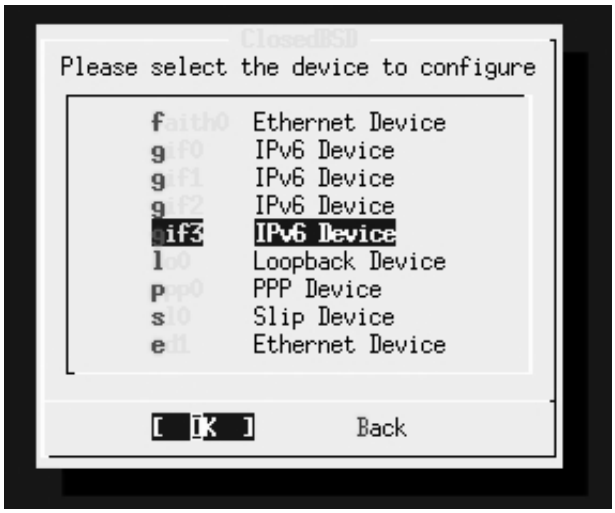

Figura 5 - Menu de Configuração do ClosedBSD. [SIM - 16]

As configurações de interface para este protótipo ficaram da seguinte forma:

lo0	127.0.0.1	255.0.0.0
rl0	10.7.12.12	255.255.255.0
rl1	192.168.7.1	255.255.255.0 (rede interna)

e as placas utilizadas, tanto em rl0 quanto em rl1, foram uma encore ENL832-TX+ (10/100). E o gateway default ficou com a rl0 cujo IP é rl0: 10.7.12.254.

O NAT (Network Address Translation, via rl0) e o firewall também foram configurados, que neste caso permite que passe tudo:

```
add allow ip from any to any
add allow tcp from any to any
add allow udp from any to any
add allow icmp from any to any
```

e pronto! Reiniciando o computador, o roteador já está funcionando.

6 CONCLUSÕES

Embedded Systems estão presente em um grande número de aplicações, como os sistemas de controle de vôo ou sistemas de processamento de sinais de satélites de comunicação. Isto para citar somente sistemas de situação crítica em que o fator tempo é determinante para o bom resultado desses sistemas. Entretanto, sistemas embarcados, controlam cada vez mais aplicações da vida moderna.

O ponto forte para desenvolvimento de sistemas embarcados, está na procura da simplicidade de configurações e por opções de baixo custo. Por isso, um estudo sobre os sistemas que oferecem tais opções torna-se tão importante. Sistemas livres, tais como FreeBSD ou Linux, são as grandes ferramentas disponíveis atualmente. Além do que, estes sistemas agregam eficiência e baixo custo na lista de qualidades que os identificam.

A maior diferença entre FreeBSD e Linux está na licença sob os quais são distribuídos (*BSD ou GNU/GPL), entretanto, no decorrer deste trabalho, ficou claro que tanto um quanto outro tem potencial para promover desenvolvimento na área de projetos embarcados, e a escolha final depende exclusivamente do desenvolvedor que deverá optar por aquele com que mais se identifica.

Além disso, existe toda uma filosofia que deve ser preservada por desenvolvedores em geral. Sistemas livres permite a disseminação do conhecimento para todos, independente de credo, raça ou classe social, e ainda oferece produtos confiáveis e seguros.

Enfim, durante o decorrer dessa pesquisa fica evidente que, tanto Linux quanto FreeBSD são adequados para sistemas embarcados, entretanto o desenvolvimento do protótipo deste trabalho, que é um roteador, foi feito sobre o PicoBSD por acreditar que esta aplicação, neste sistema, minimiza o trabalho do desenvolvedor, isso não quer dizer, que para outras propostas, o sistema Linux não seja mais adequado.

7 ANEXOS[36]

LICENÇA GNU LIBRARY GENERAL PUBLIC LICENSE (LGPL)[37]

Copyright (C) 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and

[36] Todas as licenças em anexo foram retiradas do site http://projects.openresources.com/.

[37] Esta é a licença GNU Library General Public License (LGPL), versão 2 (junho de 1991). Ela é projetada para proteger as bibliotecas desenvolvidas sob a Fundação de Software Livre, e também é usada para muitas outras bibliotecas.

change free software--to make sure the software is free for all its users.

This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link a program with the library, you must provide complete object files to the recipients so that they can relink them with the library, after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Our method of protecting your rights has two steps: (1) copyright the library, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the library.

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public

License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a ``work based on the library'' and a ``work that uses the library''. The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary General Public License rather than by this special one.

GNU LIBRARY GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library which contains a notice placed by the

copyright holder or other authorized party saying it may be distributed under the terms of this Library General Public License (also called “this License”). Each licensee is addressed as “you”.

A “library” means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The “Library”', below, refers to any such software library or work which has been distributed under these terms. A “work based on the Library” means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term “modification”.)

“Source code” for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this

License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if

the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based

on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that

distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Library General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and ``any later version'', you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY ``AS IS'' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR

CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

LICENÇA BSD[38]

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: “This product includes software developed by the University of California, Berkeley and its contributors".

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

[38] Esta é a licença aplicada às distribuições do Computer Science Research Group, da Universidade de Califórnia em Berkeley. [BAR – 02]

SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LICENÇA XFREE86 PROJECT[39]

[39] Esta é a licença do XFree86 3.3.2, distribuída pelo Projeto de XFree86. [BAR – 02]

LICENÇA TCL/TK[40]

This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files.

The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

GOVERNMENT USE: If you are acquiring this software on behalf of the U.S. government, the Government shall have only "Restricted Rights" in the software and related documentation as defined in the Federal Acquisition Regulations (FARs) in Clause 52.227.19 (c) (2). If you are

[40] Esta é a licença sob o qual é distribuído a versão 7.6 do Tcl e versão 4.2 do Tk. [BAR – 02]

acquiring the software on behalf of the Department of Defense, the software shall be classified as "Commercial Computer Software" and the Government shall have only "Restricted Rights" as defined in Clause 252.227-7013 (c) (1) of DFARs. Notwithstanding the foregoing, the authors grant the U.S. Government and others acting in its behalf permission to use and distribute the software in accordance with the terms specified in this license.

ARTISTIC LICENSE[41]

Preamble:

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the Package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

It also grants you the rights to reuse parts of a Package in your own programs without transferring this License to those programs, provided that you meet some reasonable requirements.

Definitions:

“Package” refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.

“Standard Version” refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder as specified below.

“Copyright Holder” is whoever is named in the copyright or copyrights for the package.

“You” is you, if you're thinking about copying or distributing this Package.

“Reasonable copying fee” is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)

“Freely Available” means that no fee is charged for the item itself, though there may be fees

[41] Licença artística. Esta é a Licença Artística, usada entre outros por Larry Wall como uma das licenças de distribuição do Perl. [BAR – 02].

involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

(a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as uunet.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

(b) use the modified Package only within your corporation or organization.

(c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.

(d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:

(a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.

(b) accompany the distribution with the machine-readable source of the Package with your modifications.

(c) give non-standard executables non-standard names, and clearly document the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.

(d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.

6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package. If such scripts or library files are aggregated with this Package via the so-called “undump” or “unexec” methods of producing a binary executable image, then distribution of such an image shall neither be construed as a distribution of this Package nor shall it fall under the restrictions of Paragraphs 3 and 4, provided that you do not represent such an executable image as a Standard Version of this Package.

7. You may reuse parts of this Package in your own programs, provided that you explicitly state where you got them from, in the source code (and, left to your courtesy, in the documentation), duplicating all the associated copyright notices and disclaimers. Besides your changes, if any, must be clearly marked as such. Parts reused that way will no longer fall under this license if, and only if, the name of your program(s) have no immediate connection with the name of the Package itself or its associated programs. You may then apply whatever restrictions you wish on the reused parts or choose to place them in the Public Domain--this will apply only within the context of your package.

8. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

9. THIS PACKAGE IS PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

APPLE PUBLIC SOURCE LICENSE[42]

Version 1.0 - March 16, 1999

Please read this License carefully before downloading this software. By downloading and using this software, you are agreeing to be bound by the terms of this License. If you do not or cannot agree to the terms of this License, please do not download or use the software.

1. General Definitions. This License applies to any program or other work which Apple Computer, Inc. ("Apple") publicly announces as subject to this Apple Public Source License and which contains a notice placed by Apple identifying such program or work as "Original Code" and stating that it is subject to the terms of this Apple Public Source License version 1.0 (or subsequent version thereof), as it may be revised from time to time by Apple ("License"). As used in this License:

1.1 "Applicable Patents" mean: (a) in the case where Apple is the grantor of rights, (i) patents or patent applications that are now or hereafter acquired, owned by or assigned to Apple and (ii) whose claims cover subject matter contained in the Original Code, but only to the extent necessary to use, reproduce and/or distribute the Original Code without infringement; and (b) in the case where You are the grantor of rights, (i) patents and patent applications that are now or hereafter acquired, owned by or assigned to You and (ii) whose claims cover subject matter in Your Modifications, taken alone or in combination with Original Code.

1.2 "Covered Code" means the Original Code, Modifications, the combination of Original Code and any Modifications, and/or any respective portions thereof.

1.3 "Deploy" means to use, sublicense or distribute Covered Code other than for Your internal research and development (R&D), and includes without limitation, any and all internal use or

---

[42] Esta é versão 1.0 da licença anunciada pela Apple no dia 16 de março de 1999, aplicada ao código desenvolvido por ela, disponibilizado para download na Rede. Entretanto seu conteúdo deixa dúvidas quanto a definição, desta empresa, quanto a “software livre”. A licença está disponível no site http://www.apple.com/publicsource. [BAR – 02]

distribution of Covered Code within Your business or organization except for R&D use, as well as direct or indirect sublicensing or distribution of Covered Code by You to any third party in any form or manner.

1.4 "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.5 "Modifications" mean any addition to, deletion from, and/or change to, the substance and/or structure of Covered Code. When code is released as a series of files, a Modification is: (a) any addition to or deletion from the contents of a file containing Covered Code; and/or (b) any new file or other representation of computer program statements that contains any part of Covered Code.

1.6 "Original Code" means the Source Code of a program or other work as originally made available by Apple under this License, including the Source Code of any updates or upgrades to such programs or works made available by Apple under this License, and that has been expressly identified by Apple as such in the header file(s) of such work.

1.7 "Source Code" means the human readable form of a program or other work that is suitable for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an executable (object code).

1.8 "You" or "Your" means an individual or a legal entity exercising rights under this License. For legal entities, "You" or "Your" includes any entity which controls, is controlled by, or is under common control with, You, where "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of fifty percent (50%) or more of the outstanding shares or beneficial ownership of such entity.

2. Permitted Uses; Conditions & Restrictions. Subject to the terms and conditions of this License, Apple hereby grants You, effective on the date You accept this License and download the Original Code, a world-wide, royalty-free, non-exclusive license, to the extent of Apple's Applicable Patents and copyrights covering the Original Code, to do the following:

2.1 You may use, copy, modify and distribute Original Code, with or without Modifications, solely for Your internal research and development, provided that You must in each instance:

(a) retain and reproduce in all copies of Original Code the copyright and other proprietary notices

and disclaimers of Apple as they appear in the Original Code, and keep intact all notices in the Original Code that refer to this License;

(b) include a copy of this License with every copy of Source Code of Covered Code and documentation You distribute, and You may not offer or impose any terms on such Source Code that alter or restrict this License or the recipients' rights hereunder, except as permitted under Section 6; and

(c) completely and accurately document all Modifications that you have made and the date of each such Modification, designate the version of the Original Code you used, prominently include a file carrying such information with the Modifications, and duplicate the notice in Exhibit A in each file of the Source Code of all such Modifications.

2.2 You may Deploy Covered Code, provided that You must in each instance:

(a) satisfy all the conditions of Section 2.1 with respect to the Source Code of the Covered Code;

(b) make all Your Deployed Modifications publicly available in Source Code form via electronic distribution (e.g. download from a web site) under the terms of this License and subject to the license grants set forth in Section 3 below, and any additional terms You may choose to offer under Section 6. You must continue to make the Source Code of Your Deployed Modifications available for as long as you Deploy the Covered Code or twelve (12) months from the date of initial Deployment, whichever is longer;

(c) must notify Apple and other third parties of how to obtain Your Deployed Modifications by filling out and submitting the required information found at http://www.apple.com/publicsource/modifications.html; and

(d) if you Deploy Covered Code in object code, executable form only, include a prominent notice, in the code itself as well as in related documentation, stating that Source Code of the Covered Code is available under the terms of this License with information on how and where to obtain such Source Code.

3. Your Grants. In consideration of, and as a condition to, the licenses granted to You under this License:

(a) You hereby grant to Apple and all third parties a non-exclusive, royalty-free license, under Your Applicable Patents and other intellectual property rights owned or controlled by You, to use, reproduce, modify, distribute and Deploy Your Modifications of the same scope and extent as Apple's licenses under Sections 2.1 and 2.2; and

(b) You hereby grant to Apple and its subsidiaries a non-exclusive, worldwide, royalty-free, perpetual and irrevocable license, under Your Applicable Patents and other intellectual property rights owned or controlled by You, to use, reproduce, execute, compile, display, perform, modify or have modified (for Apple and/or its subsidiaries), sublicense and distribute Your Modifications, in any form, through multiple tiers of distribution.

4. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In each such instance, You must make sure the requirements of this License are fulfilled for the Covered Code or any portion thereof.

5. Limitations on Patent License. Except as expressly stated in Section 2, no other patent rights, express or implied, are granted by Apple herein. Modifications and/or Larger Works may require additional patent licenses from Apple which Apple may grant in its sole discretion.

6. Additional Terms. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations and/or other rights consistent with the scope of the license granted herein ("Additional Terms") to one or more recipients of Covered Code. However, You may do so only on Your own behalf and as Your sole responsibility, and not on behalf of Apple. You must obtain the recipient's agreement that any such Additional Terms are offered by You alone, and You hereby agree to indemnify, defend and hold Apple harmless for any liability incurred by or claims asserted against Apple by reason of any such Additional Terms.

7. Versions of the License. Apple may publish revised and/or new versions of this License from time to time. Each version will be given a distinguishing version number. Once Original Code has been published under a particular version of this License, You may continue to use it under the terms of that version. You may also choose to use such Original Code under the terms of any subsequent version of this License published by Apple. No one other than Apple has the right to modify the terms applicable to Covered Code created under this License.

8. NO WARRANTY OR SUPPORT. The Original Code may contain in whole or in part pre-

release, untested, or not fully tested works. The Original Code may contain errors that could cause failures or loss of data, and may be incomplete or contain inaccuracies. You expressly acknowledge and agree that use of the Original Code, or any portion thereof, is at Your sole and entire risk. The Original Code is provided "AS IS" and without warranty, upgrades or support of any kind and Apple and Apple's licensor(s) (for the purposes of Sections 8 and 9, Apple and Apple's licensor(s) are collectively referred to as "Apple") EXPRESSLY DISCLAIM ALL WARRANTIES AND/OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES AND/OR CONDITIONS OF MERCHANTABILITY OR SATISFACTORY QUALITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. APPLE DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE ORIGINAL CODE WILL MEET YOUR REQUIREMENTS, OR THAT THE OPERATION OF THE ORIGINAL CODE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THE ORIGINAL CODE WILL BE CORRECTED. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY APPLE OR AN APPLE AUTHORIZED REPRESENTATIVE SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY. You acknowledge that the Original Code is not intended for use in the operation of nuclear facilities, aircraft navigation, communication systems, or air traffic control machines in which case the failure of the Original Code could lead to death, personal injury, or severe physical or environmental damage.

9. Liability.

9.1 Infringement. If any of the Original Code becomes the subject of a claim of infringement ("Affected Original Code"), Apple may, at its sole discretion and option: (a) attempt to procure the rights necessary for You to continue using the Affected Original Code; (b) modify the Affected Original Code so that it is no longer infringing; or (c) terminate Your rights to use the Affected Original Code, effective immediately upon Apple's posting of a notice to such effect on the Apple web site that is used for implementation of this License.

9.2 LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES SHALL APPLE BE LIABLE FOR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO THIS LICENSE OR YOUR USE OR INABILITY TO USE THE ORIGINAL CODE, OR ANY PORTION THEREOF, whether under a theory of contract, warranty, tort (including negligence), products liability or otherwise, even if APPLE has been advised of the possibility of such damages AND NOTWITHSTANDING THE FAILURE OF ESSENTIAL PURPOSE OF ANY REMEDY. In no event shall Apple's total liability to You for all

damages under this License exceed the amount of fifty dollars ($50.00).

10. Trademarks. This License does not grant any rights to use the trademarks or trade names "Apple", "Apple Computer", "Mac OS X", "Mac OS X Server" or any other trademarks or trade names belonging to Apple (collectively "Apple Marks") and no Apple Marks may be used to endorse or promote products derived from the Original Code other than as permitted by and in strict compliance at all times with Apple's third party trademark usage guidelines which are posted at http://www.apple.com/legal/guidelinesfor3rdparties.html.

11. Ownership. Apple retains all rights, title and interest in and to the Original Code and any Modifications made by or on behalf of Apple ("Apple Modifications"), and such Apple Modifications will not be automatically subject to this License. Apple may, at its sole discretion, choose to license such Apple Modifications under this License, or on different terms from those contained in this License or may choose not to license them at all. Apple's development, use, reproduction, modification, sublicensing and distribution of Covered Code will not be subject to this License.

12. Termination.

12.1 Termination. This License and the rights granted hereunder will terminate:

(a) automatically without notice from Apple if You fail to comply with any term(s) of this License and fail to cure such breach within 30 days of becoming aware of such breach;

(b) immediately in the event of the circumstances described in Sections 9.1 and/or 13.6(b); or

(c) automatically without notice from Apple if You, at any time during the term of this License, commence an action for patent infringement against Apple.

12.2 Effect of Termination. Upon termination, You agree to immediately stop any further use, reproduction, modification and distribution of the Covered Code, or Affected Original Code in the case of termination under Section 9.1, and to destroy all copies of the Covered Code or Affected Original Code (in the case of termination under Section 9.1) that are in your possession or control. All sublicenses to the Covered Code which have been properly granted prior to termination shall survive any termination of this License. Provisions which, by their nature, should remain in effect beyond the termination of this License shall survive, including but not limited to Sections 3, 5, 8, 9,

10, 11, 12.2 and 13. Neither party will be liable to the other for compensation, indemnity or damages of any sort solely as a result of terminating this License in accordance with its terms, and termination of this License will be without prejudice to any other right or remedy of either party.

13. Miscellaneous.

13.1 Export Law Assurances. You may not use or otherwise export or re-export the Original Code except as authorized by United States law and the laws of the jurisdiction in which the Original Code was obtained. In particular, but without limitation, the Original Code may not be exported or re-exported (a) into (or to a national or resident of) any U.S. embargoed country or (b) to anyone on the U.S. Treasury Department's list of Specially Designated Nationals or the U.S. Department of Commerce's Table of Denial Orders. By using the Original Code, You represent and warrant that You are not located in, under control of, or a national or resident of any such country or on any such list.

13.2 Government End Users. The Covered Code is a ``commercial item'' as defined in FAR 2.101. Government software and technical data rights in the Covered Code include only those rights customarily provided to the public as defined in this License. This customary commercial license in technical data and software is provided in accordance with FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for Department of Defense purchases, DFAR 252.227-7015 (Technical Data - Commercial Items) and 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). Accordingly, all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

13.3 Relationship of Parties. This License will not be construed as creating an agency, partnership, joint venture or any other form of legal association between You and Apple, and You will not represent to the contrary, whether expressly, by implication, appearance or otherwise.

13.4 Independent Development. Nothing in this License will impair Apple's right to acquire, license, develop, have others develop for it, market and/or distribute technology or products that perform the same or similar functions as, or otherwise compete with, Modifications, Larger Works, technology or products that You may develop, produce, market or distribute.

13.5 Waiver; Construction. Failure by Apple to enforce any provision of this License will not be deemed a waiver of future enforcement of that or any other provision. Any law or regulation which provides that the language of a contract shall be construed against the drafter will not apply to this

License.

13.6 Severability. (a) If for any reason a court of competent jurisdiction finds any provision of this License, or portion thereof, to be unenforceable, that provision of the License will be enforced to the maximum extent permissible so as to effect the economic benefits and intent of the parties, and the remainder of this License will continue in full force and effect. (b) Notwithstanding the foregoing, if applicable law prohibits or restricts You from fully and/or specifically complying with Sections 2 and/or 3 or prevents the enforceability of either of those Sections, this License will immediately terminate and You must immediately discontinue any use of the Covered Code and destroy all copies of it that are in your possession or control.

13.7 Dispute Resolution. Any litigation or other dispute resolution between You and Apple relating to this License shall take place in the Northern District of California, and You and Apple hereby consent to the personal jurisdiction of, and venue in, the state and federal courts within that District with respect to this License. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded.

13.8 Entire Agreement; Governing Law. This License constitutes the entire agreement between the parties with respect to the subject matter hereof. This License shall be governed by the laws of the United States and the State of California, except that body of California law concerning conflicts of law.

Where You are located in the province of Quebec, Canada, the following clause applies: The parties hereby confirm that they have requested that this License and all related documents be drafted in English. Les parties ont exigé que le présent contrat et tous les documents connexes soient rédigés en anglais.

EXHIBIT A.

“Portions Copyright © 1999 Apple Computer, Inc. All Rights Reserved.

This file contains Original Code and/or Modifications of Original Code as defined in and that are subject to the Apple Public Source License Version 1.0 (the 'License'). You may not use this file except in compliance with the License. Please obtain a copy of the License at http://www.apple.com/publicsource and read it before using this file.

The Original Code and all software distributed under the License are distributed on an 'AS IS' basis, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, AND APPLE HEREBY DISCLAIMS ALL SUCH WARRANTIES, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. Please see the License for the specific language governing rights and limitations under the License.

JUNIPER PUBLIC LICENSE[43]

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgements:

> This product includes software developed by Obtuse Systems Corporation and its contributors.
> This product includes software developed by Eric Young (eay@mincom.oz.au)

4. Neither the name of the Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY OBTUSE SYSTEMS CORPORATION “AS

[43] www.obtuse.com/juniper/

IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL OBTUSE SYSTEMS CORPORATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

PARTE DO CÓDIGO FONTE JUNIPER

Rotinas de controle de acesso[44].

```
/*
 * Access control routines for juniperd
 */

char *juniper_acl_copyright =
"Copyright 1996 - Obtuse Systems Corporation - All rights reserved.";
char *juniper_acl_rcsid = "Juniper $Id: acl.c,v 1.2 1999/02/14 00:28:10
beck Exp $";

/* RCS information:
 * $Id: acl.c,v 1.2 1999/02/14 00:28:10 beck Exp $
 */

#include<stdio.h>
#include<ctype.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<arpa/inet.h>
#include<string.h>
#include<syslog.h>
#include<netdb.h>
#include<tparser.h>
#include<juniper_util.h>
#include<juniperd.h>

int paranoia(struct peer_info *pi) {

  /*
   * Does this connection trigger paranoia? Any connection will that
   * does not fully resolve in the forward and reverse directions.
   * that is:
   *
   * 1) reverse lookup of peername address IP, gives a hostname H,
   * (indicated by pi->peer_dirty_reverse_he and pi-
>peer_clean_reverse_name
   * existing)
   * 2) a forward lookup of H contains the address IP (indicated by
   * pi->peer_ok_addr existing)
   *
   *
```

[44] www.obtuse.com/juniper/

```
   * We also get paranoid if the cleaned up hostname or ident reply
differs
   * from the one from the lookup, i.e. someone can be trying
   * `/bin/rm -rf /`@evil.org :-)
   *
   * Anytime paranoia is triggered, we syslog why, and return 1;
   * otherwise we return 0;
   */

  /* If we got an ident reply, was there any sillyness in it? */
  if ((pi->peer_dirty_ident != NULL) && (pi->peer_clean_ident != NULL)) {
    if (strcmp(pi->peer_dirty_ident, pi->peer_clean_ident) != 0) {
      syslog(LOG_NOTICE, "NOTICE - Suspicious characters in ident
response from %s, cleaned to %s",
           ((pi->peer_clean_reverse_name == NULL)?
            inet_ntoa(pi->peer_sa->sin_addr):pi-
>peer_clean_reverse_name),
           pi->peer_clean_ident);
      return(1);
    }
  }
  /* Did we get a reverse lookup? */
  if ((pi->peer_dirty_reverse_name == NULL) || (pi-
>peer_clean_reverse_name == NULL)){
    syslog(LOG_NOTICE, "NOTICE - Could not resolve %s to a hostname",
         inet_ntoa(pi->peer_sa->sin_addr));
    return(1);
  }
  /* Was there any sillyness in the hostname */
  if (strcmp(pi->peer_dirty_reverse_name, pi->peer_clean_reverse_name) !=
0) {
    syslog(LOG_NOTICE, "NOTICE - Suspicious characters in hostname
resolved from %s, cleaned to %s",
         inet_ntoa(pi->peer_sa->sin_addr),
         pi->peer_clean_reverse_name);
    return(1);
  }
  /* Did we forward resolve? */
  if (pi->peer_dirty_forward_name == NULL) {
    syslog(LOG_NOTICE, "NOTICE - Could not resolve hostname %s",
         pi->peer_clean_reverse_name);
    return(1);
  }
  /* Did we find a matching address on the forward resolve? */
  if (pi->peer_ok_addr == NULL) {
    syslog(LOG_NOTICE, "NOTICE - Possible spoof, address %s claims to be
host %s",
         inet_ntoa(pi->peer_sa->sin_addr),
         pi->peer_clean_reverse_name);
    return(1);
  }
  /* Golly! we made it through all that and we're still ok.. */
  return(0);
}

int match_pattern(char *pat, char *string) {
  char c;

  while (1) {
```

```
    c = *pat;
    pat++;
    switch(c) {
    case '\0' :
      return(*string == '\0');
    case '*' :
      c=*pat;
      while (c == '*') {
      pat++;
      c = *pat;
      }
      if (c == '\0')  {
      return(1);
      }
      while (*string != '\0') {
      if (match_pattern(pat, string)) {
        return(1);
      }
      string++;
      }
      return(0);
    default:
      if (c != *string) {
      return(0);
      }
      string++;
      break;
    }
  }
}


int match_name(char *pat, const char *string, int no_ip_check) {
  /*
   * Match a name against an acl pattern. the name is forced to
   * lowercase first.
   */

  if (string == NULL) {
    if (strcmp(pat,"*") == 0) {
      return(1);
    }
    if (strcmp(pat,"UNKNOWN") == 0) {
      return(1);
    }
    return(0);
  } else if (strcmp(pat, "KNOWN") == 0) {
    return(*string != '\0');
  } else if ((!no_ip_check) && (inet_addr(string) != -1)) {
    /* We're not expecting the string to look like an ip address, but it
     * does. This means someone *could* be playing some very silly DNS
     * games with a dotted decimal hostname. So we fail this match.
     *
     * We can safely syslog the name, since we know inet_addr converted
it
     * to an address successfully.
     */
    syslog(LOG_ALERT, "ALERT - Name \"%s\" looks like an ip address.
possible acl subversion attempt!", string);
```

```
    return(0);
  }
  else {
    int i;
    char *buf;
    buf = strdup(string);
    /* force string to lower case */
    for (i = 0; i < strlen(string); i++) {
      buf[i] = tolower(buf[i]);
    }
    if (match_pattern(pat, buf)) {
      free(buf);
      return(1);
    }
    free(buf);
    return(0);
  }
}


void log_toast(struct peer_info *pi, char *reason) {
  char *peer_sa_ntoa, *my_sa_ntoa;

  /* acl's have decided we don't like connection pi, make an
   * appropriate syslog to that effect
   */
  peer_sa_ntoa = strdup( inet_ntoa(pi->peer_sa->sin_addr) );
  my_sa_ntoa = strdup( inet_ntoa(pi->my_sa->sin_addr) );
  syslog(LOG_NOTICE, "NOTICE - Refused connection from %s@%s:%u to %s:%u,
reason: %s",
       ((pi->peer_clean_ident == NULL)?"UNKNOWN":
        pi->peer_clean_ident),
       ((pi->peer_clean_reverse_name == NULL)?
        peer_sa_ntoa:
        pi->peer_clean_reverse_name),
       ntohs(pi->peer_sa->sin_port),
       ((pi->my_clean_reverse_name == NULL)?
        my_sa_ntoa:
        pi->my_clean_reverse_name),
       ntohs(pi->my_sa->sin_port),
       reason);
  free(peer_sa_ntoa);
  free(my_sa_ntoa);
}

int match_proxy(proxy_desc_t *pdp, struct peer_info *pi, char *pat) {
  static char buf[32];
  if (match_name(pat, pdp->pd_name, 0)) {
    return(1);
  }
  snprintf(buf, 32, "%u", ntohs(pi->my_sa->sin_port));
  if (match_name(pat, buf, 1)) {
    return(1);
  }
  return(0);
}

int match_connection(struct peer_info *pi, char *pat) {
```

```
char*from;
char*fromp;
char*fromu;
char*to;
char*top;

if ((to = strchr(pat, '>')) != NULL) {
  *to='\0';
  to++;
  if ((top = strchr(to, ':')) != NULL) {
    *top='\0';
    top++;
  }
}
else {
  top=NULL;
}
if ((from = strrchr(pat, '@')) != NULL) {
  *from='\0';
  from++;
  fromu = pat;
}
else {
  from = pat;
  fromu = NULL;
}
if ((fromp = strrchr(from, ':')) != NULL) {
  *fromp='\0';
  fromp++;
}

/* does peer host match? */

if (!match_name(from, inet_ntoa(pi->peer_sa->sin_addr), 1)) {
  if (!match_name(from,
              ((pi->peer_dirty_reverse_name != NULL)
               ?pi->peer_dirty_reverse_name:NULL), 0)) {
    return(0);
  }
}


/* Yep it does, does peer port match? */

if (fromp != NULL) {
  char tbuf[32];
  tbuf[0]='\0';
  snprintf(tbuf, 32, "%d", ntohs(pi->peer_sa->sin_port));
  if (!match_name(fromp, tbuf, 1)) {
    return(0);
  }
}


/* Peer port now either matches or we didn't care.
 * let's check the user if there is one.
 */

if (fromu != NULL) {
```

```
    if (!match_name(fromu, pi->peer_dirty_ident, 0)) {
      return(0);
    }
  }


  /*
   * Peer user now either matches or we didn't care.
   * let's check where the connection is going.
   */

  if (to != NULL) {
    if (!match_name(to, inet_ntoa(pi->my_sa->sin_addr), 1)) {
      if ((pi->my_dirty_reverse_name != NULL) &&
        (!match_name(to, pi->my_dirty_reverse_name, 0))) {
      return(0);
      }
    }
  }

  /* Destination now matches if it was there
   * finally, check destination port
   */

  if (top != NULL) {
    char tbuf[32];
    tbuf[0]='\0';
    snprintf(tbuf, 32, "%d", ntohs(pi->my_sa->sin_port));
    if (!match_name(top, tbuf, 1)) {
      return(0);
    }
  }

  /* Gosh, everything passed. I guess it's a match! */
  return(1);
}

void juniper_check_connection(struct peer_info *pi, proxy_desc_t *pdp,
                        char *acl_file) {
  parser_file_t *acf;
  char tbuf[1024];

  /* a NULL acl_file means none was selected in the juniperd.conf
   * file, therefore, we won't deny any connections, so we
   * simply return.
   */
  if (acl_file == NULL) {
    return;
  }


  /* if we have given a name of an access control file, then the rule
   * is that we will specify everything allowed in there, so the default
   * action is to toss it unless it gets allowed.
   */

  /* open the access control file */
  acf = parser_openfile(acl_file) ;
  if (acf == NULL) {
```

```
    syslog(LOG_ERR, "ERROR - Couldn't open Access file %s for reading",
acl_file);
    log_toast(pi, "Couldn't read access file");
    exit(-1);
  }

  while (tokenize_line(acf) == 0) {
    tokenized_line_t *tlp;
    tlp = acf->pf_tokenized_line;
    if (tlp->tl_ntokens < 2) {
      syslog(LOG_ERR, "ERROR - Not enough fields line %d, in %s",
           tlp->tl_lnum,
           acf->pf_fname);
    }
    else {
      if (match_proxy(pdp, pi, tlp->tl_tokens[1])) {
      if (strcmp(tlp->tl_tokens[0], "allow") == 0) {
        int i;
        for (i = 2; i < tlp->tl_ntokens; i++) {
          if (match_connection(pi, tlp->tl_tokens[i])) {
            parser_closefile(acf);
            return;
          }
        }
      }

      else if (strcmp(tlp->tl_tokens[0], "paranoid") == 0) {
        if (tlp->tl_ntokens != 2) {
          syslog(LOG_ERR, "ERROR - \"paranoid\" takes only one argument,
line %d, in %s",
                 tlp->tl_lnum,
                 acf->pf_fname);
        }
        if (paranoia(pi)) {
          snprintf(tbuf, 1024, "triggered paranoia, line %d of %s",
                 tlp->tl_lnum,
                 acf->pf_fname);
          log_toast(pi, tbuf);
          exit(-1);
        }
      }

      else if (strcmp(tlp->tl_tokens[0], "check") == 0) {
        if (tlp->tl_ntokens != 2) {
          syslog(LOG_ERR, "ERROR - \"check\" takes only one argument,
line %d, in %s",
                 tlp->tl_lnum,
                 acf->pf_fname);
        }
        paranoia(pi); /* do paranoia check, but continue */
      }

      else if (strcmp(tlp->tl_tokens[0], "deny") == 0) {
        int i;
        for (i = 2; i < tlp->tl_ntokens; i++) {
          if (match_connection(pi, tlp->tl_tokens[i])) {
            snprintf(tbuf, 1024,  "access denied by line %d of %s",
                 tlp->tl_lnum,
                 acf->pf_fname);
```

```
            log_toast(pi, tbuf);
            exit(-1);
          }
        }
      }

#ifdef KIDENT

      else if (strcmp(tlp->tl_tokens[0], "kident") == 0) {
         int i;
         for (i = 2; i < tlp->tl_ntokens; i++) {
            if (match_connection(pi, tlp->tl_tokens[i])) {
             /* This means we expect connections from this host
              * to be handing us kerberos ident info. We must convert
              * what we have from any ident response extract ther
              * username and verify the validity of the kerberos ticket
              */
             juniper_krb_ident(pi);
             /* pi->clean_ident should now be the real username extracted
              * from the kerberos info. We now continue, and all future
              * rules will match against this info.
              */
            }
         }
      }

      else {
        syslog(LOG_ERR, "ERROR - Action must be \"allow\", \"paranoid\",
\"deny\", or \"kident\" at line %d in %s",
             tlp->tl_lnum,
             acf->pf_fname);
      }
#else
      else {
        syslog(LOG_ERR, "ERROR - Action must be \"allow\", \"paranoid\",
or \"deny\" at line %d in %s",
             tlp->tl_lnum,
             acf->pf_fname);
      }
#endif
      }
    }
  }
  /* If we read the whole file and got through it, we didn't match
   * something to explicitly allow or deny it, so the default action
   * is always to deny it
   */
  snprintf(tbuf, 1024, "no allowing match in %s",
        acf->pf_fname);
  log_toast(pi, tbuf);
  exit(-1);
}
```

Juniper firewall – Rotinas de kernel

```
/*
 * Juniper firewall - kernel routines
```

```
 * ------------------------------------------------------------
 * $Id: juniper_firewall.c,v 1.2 1999/02/13 20:14:41 beck Exp $
 * ------------------------------------------------------------
 *
 * Copyright (c) 1996,1997,1998,1999 Obtuse Systems Corporation
 * all rights reserved
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' WITHOUT ANY WARRANTIES
 * OF ANY KIND. USE OF THE SOFTWARE (WITH OR WITHOUT MODIFICATIONS)
 * IN THIS OR ANY FORM FOR ANY PURPOSE IS AT YOUR OWN RISK. IF
 * YOU FIND THIS TO BE UNACCEPTABLE THEN DON'T USE THE SOFTWARE.
 *
 * This software is part of Obtuse Systems Corporation's Juniper Firewall
 * Toolkit. Use of this software is covered by the terms of the Juniper
 * License Agreement. A copy of this agreement is in the file LICENSE
 * included with the Juniper release.
 *
 * Contact info@obtuse.com if your copy of the file LICENSE is missing.
 */

#ifdef __linux__
#ifdef 0
#define V12  /* Define for kernel < 1.3 */
#endif
/* linux kernel includes */
#include <linux/types.h>
#include <linux/kernel.h>
#include <linux/sched.h>
#include <linux/mm.h>
#include <linux/string.h>
#include <linux/errno.h>
#include <linux/config.h>

#include <linux/socket.h>
#include <linux/sockios.h>
#include <linux/in.h>
#include <linux/inet.h>
#include <linux/netdevice.h>
#include <linux/etherdevice.h>
#include <linux/proc_fs.h>
#include <linux/stat.h>

#ifdef V12
#include "snmp.h"
#include "ip.h"
#include "protocol.h"
#include "route.h"
#include "tcp.h"
#include "udp.h"
#else
#include <net/snmp.h>
#include <net/ip.h>
#include <net/protocol.h>
#include <net/route.h>
#include <net/tcp.h>
#include <net/udp.h>
#endif

#include <linux/skbuff.h>
```

```
#ifdef V12
#include "sock.h"
#include "arp.h"
#include "icmp.h"
#include "raw.h"
#else
#include <net/sock.h>
#include <net/arp.h>
#include <net/icmp.h>
#include <net/raw.h>
#include <net/checksum.h>
#endif
#include <linux/igmp.h>
#include <linux/ip_fw.h>
#include <linux/juniper_firewall.h>
#define printf printk /* sigh */

#else

/* Bezerkeley kernel includes */
#include <sys/param.h>
#include <sys/types.h>
#include <sys/malloc.h>
#include <sys/mbuf.h>
#include <sys/errno.h>
#include <sys/protosw.h>
#include <sys/socket.h>
#include <sys/proc.h>

#include <machine/cpu.h>

#include <net/if.h>
#include <net/route.h>
#include <net/netisr.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#ifdef __FreeBSD__
#include <sys/queue.h>
#include <sys/systm.h>
#include <netinet/ip_var.h>
#endif
#include <netinet/in_pcb.h>
#include <netinet/in_var.h>
#include <netinet/juniper_firewall.h>
#include <sys/systm.h>
#include <sys/syslog.h>

int juniper_silent = 0; /* Let people make the kernel slightly quieter */
#endif

/*
 * Sanity check on JUNIPER_{$OS} variables
 */

#ifdef JUNIPER_LINUX
    /* life is wonderful in Linux land */
#  ifdef JUNIPER_BSD
    %%% life is weird - both JUNIPER_LINUX and JUNIPER_BSD are defined
```

```
#  endif
#else
#  ifdef JUNIPER_BSD
    /* life is wonderful in Berkeley land */
#  else
    %%% life is NOT wonderful - one of JUNIPER_LINUX or JUNIPER_BSD must
be defined
#  endif
#endif

/*
 * Vector of trusted interface names
 */

static int max_trusted_if_count = 0, trusted_if_count = 0;
static char **trusted_ifs = 0;

/*
 * Define an interface as trusted.
 */

int
juniper_define_trusted_if( char *ifname, int ifname_len )
{
    register int ix;


    /*
     * Bail out now if we aren't root. should be checked by caller
     * if we want to catch it gracefully. If sanity check fails here
     * get out fast, something's wrong.
     */

#ifdef JUNIPER_LINUX
    if (!suser())
      return EPERM;
#endif

    /*
     * Make sure that there is a 0 byte somewhere inside the parameter.
     */

    for ( ix = 0; ix < ifname_len; ix += 1 ) {
      if ( ifname[ix] == '\0' ) {
          break;
      }
    }
    if ( ix == ifname_len ) {
      return(EINVAL);
    }

    /*
     * Look for the interface (this ain't called much so speed ain't of
the
     * essence)
     */

    for ( ix = 0; ix < trusted_if_count; ix += 1 ) {
      if ( strcmp(trusted_ifs[ix],ifname) == 0 ) {
```

```
          break;
      }
    }

    /*
     * If we found it then it is already trusted
     */

    if ( ix < trusted_if_count ) {
      return(EALREADY); /* Already trusted */
    }

    /*
     * Do we need to expand the vector?
     */

    if ( trusted_if_count == max_trusted_if_count ) {
      char **old_vector;
      int new_max;

      old_vector = trusted_ifs;
      new_max = max_trusted_if_count + 10;
#ifdef JUNIPER_BSD
      trusted_ifs = (char **)malloc(new_max*sizeof(char *), M_DEVBUF,
M_DONTWAIT);
#endif
#ifdef JUNIPER_LINUX
      trusted_ifs = (char **)kmalloc(new_max*sizeof(char *), GFP_KERNEL);
#endif
      if ( trusted_ifs == 0 ) {
          return(ENOSPC);
      }
      if ( old_vector != 0 ) {
          for ( ix = 0; ix < trusted_if_count; ix += 1 ) {
            trusted_ifs[ix] = old_vector[ix];
            /* Say no to leaks. free the old ones :-)*/
            }
#ifdef JUNIPER_BSD
          free(old_vector,M_DEVBUF);
#endif
#ifdef JUNIPER_LINUX
          kfree_s(old_vector, max_trusted_if_count*sizeof(char *));
#endif
      }
      max_trusted_if_count = new_max;
    }

    /*
     * Add the interface to the vector
     */

    {
#ifdef JUNIPER_BSD
      register char *p;
      p = (char *)malloc( strlen(ifname) + 1, M_DEVBUF, M_DONTWAIT );
      if ( p == 0 ) {
      return(ENOSPC);
      }
      strcpy(p,ifname);
```

```
      trusted_ifs[trusted_if_count++] = p;
#endif
#ifdef JUNIPER_LINUX
      trusted_ifs[trusted_if_count++] = ifname;
#endif
    }
    log(LOG_INFO, "Added trusted interface: %s\n",
trusted_ifs[trusted_if_count - 1]);

    return(0);
}

/*
 * Is a specified interface name a trusted interface?
 */

int
juniper_trusted_ifname(const char *ifname)
{
    register int ix;


    for ( ix = 0; ix < trusted_if_count; ix += 1 ) {
      register char *pp = trusted_ifs[ix];
      register const char *pt = ifname;
      register char ch;

      /* optimize out all strcmp calls :-) */
      while ((ch = *pt++) == *pp++)
        if (ch == 0)
          return(1);
      }
    return(0);
}


/*
 * Is a specified interface a trusted interface?
 *
 * Invalid names are not trusted.
 */

#ifdef JUNIPER_BSD
int
juniper_trusted_if(struct ifnet *ifp)
{
    if ( ifp == NULL ) {
      return(-1);
    }

#ifdef IFNAMSIZ
    /* NetBSD > 1.1 has if_xname == name+unit, unlike
     * older/other *BSD's which have if_name and if_unit
     * seperate.
     */

    if ( strlen(ifp->if_xname) > JUNIPER_MAX_IF_NAMELEN ) {
      return(0);
    }
```

```
    return( juniper_trusted_ifname( ifp->if_xname ) );
#else
    /* We're on a *BSD that has if_name and if_unit seperately */

    if ( strlen(ifp->if_name) > JUNIPER_MAX_IF_NAMELEN ) {
      return(0);
    }

    /* Close your eyes . . . */

{
    char tbuf[JUNIPER_MAX_IF_NAMELEN+10];
    register char *p, *q;
    register int unum;

    p = &tbuf[JUNIPER_MAX_IF_NAMELEN+10];
    *--p = '\0';
    unum = ifp->if_unit;
    if ( unum == 0 ) {
      *--p = '0';
    } else {

      while ( unum > 0 ) {
      *--p = '0' + (unum % 10);
      unum /= 10;
      }

    }
    unum = strlen(ifp->if_name);
    q = ifp->if_name + unum;
    while ( unum-- > 0 ) {
      *--p = *--q;
    }
    if ( q != ifp->if_name ) {
      printf("p=%x, q=%x, if_name=%x\n",
      p,q,ifp->if_name);
      panic("backwards strcpy screwed up");
    }

    return( juniper_trusted_ifname( p ) );
}
#endif
}
#endif

#ifdef JUNIPER_LINUX
int
juniper_trusted_if(struct device *dev)
{

    if ( dev == NULL ) {
      return(-1);
    }
    if ( strlen(dev->name) > JUNIPER_MAX_IF_NAMELEN ) {
      return(0);
    }
    return( juniper_trusted_ifname( dev->name ) );
}
#endif
```

```
/*
 * Flush all the trusted interfaces
 */

static
void
juniper_flush_trusted_ifs(void)
{
    register int ix;

    for ( ix = 0; ix < trusted_if_count; ix += 1 ) {
#ifdef JUNIPER_BSD
      free(trusted_ifs[ix],M_DEVBUF);
#endif
#ifdef JUNIPER_LINUX
      kfree_s(trusted_ifs[ix], strlen(trusted_ifs[ix])+1);
#endif
    }
    trusted_if_count = 0;
    return;
}

#ifdef JUNIPER_BSD /* Start of BSD-only functions */
/*
 * Process normal IP [gs]etsockopt calls
 *
 * Return values:
 *    -1 : option not recognized
 *     0 : option recognized and processed
 *    >0 : errno value to return to caller
 */

int
juniper_ip_sockopt(int op, int optname, struct mbuf **m,struct inpcb
*inp)
{
    int error;
    struct proc *p;

    error = 0;

    switch ( optname ) {

    case IP_JUNIPER_DEFINE_TRUSTED_IF:

      /* Only root can use this option */

      p = curproc; /* %%% won't work on an SMP %%% */
      error = suser(p->p_ucred, &p->p_acflag);
      if ( error == 0 ) {
          if ( op == PRCO_SETOPT ) {
            if ( *m == 0 ) {
                error = EINVAL;
            } else {
                error = juniper_define_trusted_if( mtod(*m,char*),(*m)-
>m_len );
            }
```

```
            } else {
              /* There is no get form of this request */
              error = EINVAL;
            }
          }
          return(error);

        case IP_JUNIPER_FLUSH_TRUSTED_IFS:

          /* Only root can use this option */

          p = curproc; /* %%% won't work on an SMP %%% */
          error = suser(p->p_ucred, &p->p_acflag);
          if ( error == 0 ) {
              if ( op == PRCO_SETOPT ) {
                juniper_flush_trusted_ifs();
                error = 0;
              } else {
                /* There is no get form of this request */
                error = EINVAL;
              }
          }
          return(error);

        case IP_JUNIPER_SOCKET:

          if ( op == PRCO_SETOPT ) {
              if ( m == 0 ) {
                error = EINVAL;
              } else if ( *mtod(*m, int *) == 0 ) {
                inp->inp_juniper_socket = 0;
              } else {
                inp->inp_juniper_socket = 1;
              }
          } else {
              *m = m_get(M_WAIT, MT_SOOPTS);
              *mtod(*m, int *) = inp->inp_juniper_socket;
              (*m)->m_len = sizeof(int);
          }
          return(error);

        default:

          return(-1);

        }

    }

    /*
     * Contemplate capturing a packet that isn't for this host
     */

    int
    juniper_capture_packet(register struct mbuf *m)
    {
        register struct ip *ip;
        register struct in_ifaddr *ia;
```

```
    ip = mtod(m, struct ip *);

    /*
     * If it isn't coming from a known interface then drop the packet.
     *
     * Unknown interfaces are untrusted (by definition)
     */

    if ( (m->m_flags & M_PKTHDR) == 0 || m->m_pkthdr.rcvif == 0 ) {

      log(LOG_INFO, "juniper(unknown):  dropping not-for-us packet from
%d.%d.%d.%d to %d.%d.%d.%d\n",
      ((unsigned char *)&ip->ip_src.s_addr)[0],
      ((unsigned char *)&ip->ip_src.s_addr)[1],
      ((unsigned char *)&ip->ip_src.s_addr)[2],
      ((unsigned char *)&ip->ip_src.s_addr)[3],
      ((unsigned char *)&ip->ip_dst.s_addr)[0],
      ((unsigned char *)&ip->ip_dst.s_addr)[1],
      ((unsigned char *)&ip->ip_dst.s_addr)[2],
      ((unsigned char *)&ip->ip_dst.s_addr)[3]);
      return(-1);
    }

    /*
     * If it isn't coming from a trusted interface then drop the packet.
     */

    if ( juniper_trusted_if(m->m_pkthdr.rcvif) <= 0 ) {
#ifdef IFNAMSIZ
      log(LOG_INFO, "juniper(%s):  dropping not-for-us packet from
%d.%d.%d.%d to %d.%d.%d.%d\n",
      m->m_pkthdr.rcvif->if_xname,
#else
      log(LOG_INFO "juniper(%s%d):  dropping not-for-us packet from
%d.%d.%d.%d to %d.%d.%d.%d\n",
      m->m_pkthdr.rcvif->if_name,
      m->m_pkthdr.rcvif->if_unit,
#endif
      ((unsigned char *)&ip->ip_src.s_addr)[0],
      ((unsigned char *)&ip->ip_src.s_addr)[1],
      ((unsigned char *)&ip->ip_src.s_addr)[2],
      ((unsigned char *)&ip->ip_src.s_addr)[3],
      ((unsigned char *)&ip->ip_dst.s_addr)[0],
      ((unsigned char *)&ip->ip_dst.s_addr)[1],
      ((unsigned char *)&ip->ip_dst.s_addr)[2],
      ((unsigned char *)&ip->ip_dst.s_addr)[3]);
      return(-1);
    }

    /*
     * If it is something that can't be forwarded then don't capture it
     */

    if ( m->m_flags & M_BCAST || in_canforward(ip->ip_dst) == 0 ) {
      return(-1); /* Punt! */
    }

    /*
     * Which interface would this packet be sent out via (if we let it)?
```

```
     */

    ia = (struct in_ifaddr *) ip_rtaddr(ip->ip_dst);
    if ( ia == NULL ) {
      return(-1); /* No route - punt! */
    }

    /*
     * Is out-bound interface trusted?
     */

    if ( juniper_trusted_if(ia->ia_ifa.ifa_ifp) > 0 ) {
      return(0);  /* Interface is trusted - we won't capture it
                   * although we might forward it in "ip_input.c".
                   */
    }

    return(1);          /* Out-bound interface is untrusted - capture it.
*/
}
#endif  /* end of BSD-only functions */



#ifdef JUNIPER_LINUX /* start of Linux-only functions */

int juniper_capture_packet(register struct sk_buff *skb)
{
    register struct iphdr *iph = skb->h.iph;
    struct rtable *rte;

    /*
     * If it isn't coming from a known interface then drop the packet.
     *
     * Unknown interfaces are untrusted (by definition)
     */

    if ( skb->dev == 0 ) {
      printf("juniper(unknown):  dropping not-for-us packet from
%d.%d.%d.%d to %d.%d.%d.%d\n",
          ((unsigned char *)&iph->saddr)[0],
          ((unsigned char *)&iph->saddr)[1],
          ((unsigned char *)&iph->saddr)[2],
          ((unsigned char *)&iph->saddr)[3],
          ((unsigned char *)&iph->daddr)[0],
          ((unsigned char *)&iph->daddr)[1],
          ((unsigned char *)&iph->daddr)[2],
          ((unsigned char *)&iph->daddr)[3]);
      return(-1);
    }

    /*
     * If it isn't coming from a trusted interface then drop the packet.
     */

    if ( juniper_trusted_if(skb->dev) <= 0 ) {
      printf("juniper(%s):  dropping not-for-us packet from %d.%d.%d.%d
to %d.%d.%d.%d\n",
          skb->dev->name,
```

```
          ((unsigned char *)&iph->saddr)[0],
          ((unsigned char *)&iph->saddr)[1],
          ((unsigned char *)&iph->saddr)[2],
          ((unsigned char *)&iph->saddr)[3],
          ((unsigned char *)&iph->daddr)[0],
          ((unsigned char *)&iph->daddr)[1],
          ((unsigned char *)&iph->daddr)[2],
          ((unsigned char *)&iph->daddr)[3]);
      return(-1);
    }

    /*
     * If it is something that can't be forwarded then don't capture it
     */

    if ( skb->pkt_type == PACKET_MULTICAST ) {
      return(0);  /* Let something else deal with it. */
    }

    /*
     * Which interface would this packet be sent out via (if we let it)?
     */


#ifdef V12
    rte = ip_rt_route(iph->daddr, NULL, NULL);
#else
    rte = ip_rt_route(iph->daddr, 0);
#endif
    if ( rte == NULL ) {
      return(0);  /* No route - let something else deal with it. */
    }

    /*
     * Is out-bound interface trusted?
     */


    if ( juniper_trusted_if(rte->rt_dev) > 0 ) {
      return(0);  /* Interface is trusted - we won't capture it
                   * although we might forward it in "ip_input.c".
                   */
    }

    return(1);          /* Interface is untrusted - capture the packet.
*/
}


/*
 * Process all Linux IP [gs]etsockopt calls.
 *
 * Call values:
 *      op : 1 for set, 0 for get
 *      opval : user memory
 *      oplen : length of opval
 *
 * Return values:
 *    ENOPROTOOPT : option not recognized
```

```
 *     0 : option recognized and processed
 *    >0 : errno value to return to caller
 *
 */

int
juniper_linux_ip_sockopt(int op, int optname, struct sock *sk, char
*optval, int *optlen)
{
    int error;

    error = 0;
    switch ( optname ) {

/* privileged socket options */
    case IP_JUNIPER_DEFINE_TRUSTED_IF:
      if (!suser()) {
      printk(KERN_ERR "juniper: non-root user attempted to define trusted
interface\n");
      return(EPERM);
      }
      if ( op == 1 ) {
      char * newifname;
      error=verify_area(VERIFY_READ,optval,*optlen);
      if(error)
        return error;
      newifname = kmalloc(*optlen * sizeof(char), GFP_KERNEL);
      if ( newifname == 0 ) {
        return ENOSPC;
      }
      memcpy_fromfs(newifname, optval,*optlen);
      error = juniper_define_trusted_if(newifname, *optlen);
      if ( error != 0 ) {
        kfree_s(newifname, *optlen * sizeof(char));
      }
      } else {
      /* There is no get form of this request */
      error = EINVAL;
      }
      return(error);
    case IP_JUNIPER_FLUSH_TRUSTED_IFS:
      if (!suser()) {
      printk(KERN_ERR "juniper:  non-root user attempted to flush trusted
interfaces\n");
      return(EPERM);
      }
      if ( op == 1 ) {
      juniper_flush_trusted_ifs();
      error = 0;
      } else {
      /* There is no get form of this request */
      error = EINVAL;
      }
      return(error);

      /* normal options */

    case IP_JUNIPER_SOCKET:
      if ( op == 1 ) {
```

```
      if ( optval == NULL ) {
        error = EINVAL;
      } else {
        int val;
        error=verify_area(VERIFY_READ, optval, sizeof(int));
        if(error)
          return error;
#ifdef V12
        val = get_fs_long((int *) optval);
#else
        val = get_user((int *) optval);
#endif
        if ( val == 0 ) {
          sk->sk_juniper_socket = 0;
        } else {
          sk->sk_juniper_socket = 1;
        }
      }
      } else {
      error=verify_area(VERIFY_WRITE, optval, sizeof(int));
      if(error)
        return error;
      error=verify_area(VERIFY_WRITE, optlen, sizeof(int));
      if(error)
        return error;
#ifdef V12
      put_fs_long(sizeof(int), (int *) optlen);
      put_fs_long((int)sk->sk_juniper_socket, (int *) optval);
#else
      put_user(sizeof(int), (int *) optlen);
      put_user((int)sk->sk_juniper_socket, (int *) optval);
#endif
      }
      return(error);

    default:
      return(ENOPROTOOPT);

    }
}

#endif  /* end of Linux-only functions */
```

Análise gramatical do arquivo juniperd.conf[45]

```
/*
 * Parse the juniperd.conf file
 */
```

[45] www.obtuse.com/juniper/

```
char *juniper_parse_config_copyright =
"Copyright 1996 - Obtuse Systems Corporation - All rights reserved.";
char *juniper_parse_config_rcsid = "Juniper $Id: parse_conf_file.c,v 1.4
1999/03/14 03:27:29 beck Exp $";

/*
 * RCS information:
 * $Id: parse_conf_file.c,v 1.4 1999/03/14 03:27:29 beck Exp $
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <pwd.h>
#include <netdb.h>
#include <ctype.h>
#include <fcntl.h>
#include <errno.h>

#include <tparser.h>

#include <juniper_util.h>
#include <juniperd.h>

void
free_proxy_cmd_desc(proxy_cmd_desc_t *pcdp)
{
    char **pp;

    if ( pcdp == NULL ) {
      return;
    }
    if ( pcdp->pcd_parms != NULL ) {
      for ( pp = pcdp->pcd_parms; *pp != NULL; pp += 1 ) {
          free(*pp);
      }
      free(pcdp->pcd_parms);
    }
    if (pcdp->pcd_binary != NULL)
      free(pcdp->pcd_binary);
    if (pcdp->pcd_binary_md5 != NULL)
      free(pcdp->pcd_binary_md5);
    if (pcdp->pcd_binary_stat != NULL)
      free(pcdp->pcd_binary_stat);
}

void
free_proxy_desc(proxy_desc_t *pdp)
{
    if ( pdp->pd_name != NULL ) free(pdp->pd_name);

    free_proxy_cmd_desc( pdp->pd_captured_proxy );
    free_proxy_cmd_desc( pdp->pd_trusted_daemon );
    free_proxy_cmd_desc( pdp->pd_untrusted_daemon );

    free(pdp);
```

```
}

void
free_conf_file(conf_file_t *cfp)
{
    proxy_desc_t **pdpp;
    char **tifpp;

    for ( pdpp = cfp->jc_proxies; *pdpp != NULL; pdpp += 1 ) {
      free_proxy_desc(*pdpp);
    }
    free(cfp->jc_proxies);

    for ( tifpp = cfp->jc_trusted_ifs; *tifpp != NULL; tifpp += 1 ) {
      free(*tifpp);
    }
    free(cfp->jc_trusted_ifs);

    if ( cfp->jc_proxy_directory != NULL ) free(cfp->jc_proxy_directory);

    free(cfp);
}

char *
parse_trusted_if(parser_file_t *pf)
{
    struct tokenized_line *tlp;
    char *p;

    tlp = pf->pf_tokenized_line;
    if ( tlp->tl_tokens[1] == NULL ) {
      parser_error(pf,"missing interface name on trusted-if line");
      return(NULL);
    }

    if ( tlp->tl_tokens[2] != NULL ) {
      parser_error(pf,"too many parameters on trusted-if line");
      return(NULL);
    }

    p = tlp->tl_tokens[1];

    if ( *p == '\0' ) {
      parser_error(pf,"null interface name on trusted-if line");
      return(NULL);
    }

    return( strdup( tlp->tl_tokens[1] ) );
}

char *
parse_proxy_directory(parser_file_t *pf)
{
    struct tokenized_line *tlp;
    char *p;
    struct stat sbuf;

    tlp = pf->pf_tokenized_line;
    if ( tlp->tl_tokens[1] == NULL ) {
```

```
      parser_error(pf,"missing directory name on proxy-directory line");
      return(NULL);
    }

    if ( tlp->tl_tokens[2] != NULL ) {
      parser_error(pf,"too many parameters on proxy-directory line");
      return(NULL);
    }

    p = tlp->tl_tokens[1];

    if ( *p == '\0' ) {
      parser_error(pf,"null interface name on proxy-directory line");
      return(NULL);
    }

    if ( *p != '/' ) {
      parser_error(pf,"directory name on proxy-directory line must start
with a /");
      return(NULL);
    }

    if ( stat(p,&sbuf) != 0 ) {
      parser_error(pf,"can't stat \"%s\"",p);
      return(NULL);
    }
    if ( (sbuf.st_mode & S_IFDIR) == 0 ) {
      parser_error(pf,"\"%s\" isn't a directory",p);
      return(NULL);
    }

    return( strdup( tlp->tl_tokens[1] ) );
}

int
parse_proxy_port(parser_file_t *pf, proxy_desc_t *pdp)
{
    struct tokenized_line *tlp;
    char *nextch;

    tlp = pf->pf_tokenized_line;

    if ( tlp->tl_tokens[1] == NULL ) {
      parser_error(pf,"missing port name/number");
      return(1);
    }

    if ( tlp->tl_tokens[2] != NULL ) {
      parser_error(pf,"junk after port name/number");
      return(1);
    }

    if ( isdigit(*(tlp->tl_tokens[1])) ) {
      long pnum;

      pnum = strtol( tlp->tl_tokens[1], &nextch, 10 );
      if ( *nextch != '\0' || pnum > 65535 ) {
          parser_error(pf,"invalid port number");
          return(1);
```

```
      }
      pdp->pd_port = htons( (u_short)pnum );

      return(0);
    } else {
      struct servent *svp;

      if ( pdp->pd_protocol != IPPROTO_TCP ) {
          parser_error(pf,"protocol isn't IPPROTO_TCP");
          return(1);
      }
      svp = getservbyname( tlp->tl_tokens[1], "tcp" );
      if ( svp == NULL ) {
          parser_error(pf,"unknown service name \"%s\"",tlp-
>tl_tokens[1]);
          return(1);
      }
      pdp->pd_port = svp->s_port;

      return(0);
    }

}

int
parse_proxy_username(parser_file_t *pf, proxy_desc_t *pdp)
{
    struct tokenized_line *tlp;
    struct passwd *pwp;

    tlp = pf->pf_tokenized_line;

    if ( tlp->tl_tokens[1] == NULL ) {
      parser_error(pf,"missing user id/name");
      return(1);
    }

    if ( tlp->tl_tokens[2] != NULL ) {
      parser_error(pf,"junk after user name");
      return(1);
    }

    pwp = getpwnam(tlp->tl_tokens[1]);
    if ( pwp == NULL ) {
      parser_error(pf,"invalid/unknown user name \"%s\"",tlp-
>tl_tokens[1]);
      return(1);
    }

    pdp->pd_username = strdup(tlp->tl_tokens[1]);
    pdp->pd_uid = pwp->pw_uid;
    pdp->pd_gid = pwp->pw_gid;

    return( 0 );
}

int
parse_proxy_options(parser_file_t *pf, proxy_desc_t *pdp, int
got_options)
```

```
{
    struct tokenized_line *tlp;
    static int got_option_vpc,
    got_option_pc,
    got_option_upc,
    got_option_wp,
    got_option_ui,
    got_option_ti,
    got_option_ci,
    got_option_debug,
    got_option_acct,
    got_option_timeout,
    got_option_maxsessions,
    got_option_honest,
    got_option_ipnat,
    got_option_ipnat_bc,
    got_option_no_rdns,
    got_option_no_t_rdns,
    got_option_no_u_rdns,
    got_option_no_c_rdns;
    int got_error = 0;
    char **tp;

    tlp = pf->pf_tokenized_line;

    if ( tlp->tl_tokens[1] == NULL ) {
      parser_error(pf,"no options specified on options line");
      return(1);
    }

    if (!got_options) {
      got_option_vpc = 0;
      got_option_upc = 0;
      got_option_pc = 0;
      got_option_wp = 0;
      got_option_debug = 0;
      got_option_acct = 0;
      got_option_timeout = 0;
      got_option_maxsessions = 0;
      got_option_honest = 0;
      got_option_ipnat = 0;
      got_option_ipnat_bc = 0;
      got_option_no_rdns = 0;
      got_option_no_t_rdns = 0;
      got_option_no_u_rdns = 0;
      got_option_no_c_rdns = 0;
      got_option_ui = 0;
      got_option_ti = 0;
      got_option_ci = 0;
    }

    for ( tp = tlp->tl_tokens + 1; *tp != NULL; tp += 1 ) {

      if ( strcmp(*tp,"verify-privileged-client") == 0 ) {

          if ( got_option_vpc ) {
            parser_warning(pf,
            "verify-privileged-client option appears more than once");
          }
```

```
            if ( got_option_pc ) {
              parser_warning(pf,
              "option verify-privileged-client option is implied %s",
              "by privileged-client option");
            }
            pdp->pd_verify_privileged_client = 1;
            got_option_vpc = 1;

        } else if ( strcmp(*tp,"use-privileged-port") == 0 ) {

            if ( got_option_upc ) {
              parser_warning(pf,"use-privileged-port option appears %s"
              "more than once");
            }
            if ( got_option_pc ) {
              parser_warning(pf,"option use-privileged-port option is %s",
              "implied by privileged-client option");
            }
            pdp->pd_use_privileged_port = 1;
            got_option_upc = 1;

        } else if ( strcmp(*tp,"privileged-client") == 0 ) {

            if ( got_option_pc ) {
              parser_warning(pf,"privileged-client option appears more %s",
              "than once");
            }
            if ( got_option_vpc ) {
              parser_warning(pf,"option verify-privileged-client option
%s",
              "is implied by privileged-client option");
            }
            if ( got_option_upc ) {
              parser_warning(pf,"option use-privileged-port option is %s",
              "implied by privileged-client option");
            }
            pdp->pd_verify_privileged_client = 1;
            pdp->pd_use_privileged_port = 1;
            got_option_pc = 1;

        } else if ( strcmp(*tp,"wildports") == 0 ) {

            if ( got_option_wp ) {
              parser_warning(pf,"wildports option appears %s"
              "more than once");
            }
            pdp->pd_wildports = 1;
            got_option_wp = 1;

        } else if ( strcmp(*tp,"acct") == 0 ) {

            if ( got_option_acct ) {
              parser_warning(pf,"acct option appears %s"
              "more than once");
            }
            pdp->pd_acct = 1;
            got_option_acct = 1;

        } else if ( strncmp(*tp,"debug",
```

```
                strlen("debug")) == 0 ) {
    char *cp;
    cp=(*tp)+strlen("debug");
    if ( got_option_debug ) {
      parser_warning(pf,"debug option appears %s"
      "more than once");
    }
    if (!*cp) {
      pdp->pd_debug = 1;
      got_option_debug = 1;
    }
    else {
      if (*cp == '=') {
      cp++;
      if (sscanf(cp, "%d", &(pdp->pd_debug)) == 1) {
        got_option_debug = 1;
      } else {
        parser_error(pf,"unknown argument to debug \"%s\"",*cp);
        got_error = 1;
      }
      }
      else {
      parser_error(pf,"unknown option \"%s\"",*tp);
      got_error = 1;
      }
    }
} else if ( strncmp(*tp,"timeout",
                strlen("timeout")) == 0 ) {
    char *cp;
    cp=(*tp)+strlen("timeout");
    if ( got_option_timeout ) {
      parser_warning(pf,"timeout option appears %s"
      "more than once");
    }
    if (!*cp) {
      parser_error(pf,"timeout option requires a time argument");
      got_error = 1;
    }
    else {
      if (*cp == '=') {
      cp++;
      if (sscanf(cp, "%d", &(pdp->pd_timeout)) == 1) {
        got_option_timeout = 1;
      } else {
        parser_error(pf,"unknown argument to timeout \"%s\"",cp);
        got_error = 1;
      }
      }
      else {
      parser_error(pf,"unknown option \"%s\"",*tp);
      got_error = 1;
      }
    }
} else if ( strncmp(*tp,"maxsessions",
                strlen("maxsessions")) == 0 ) {
    char *cp;
    cp=(*tp)+strlen("maxsessions");
    if ( got_option_maxsessions ) {
      parser_warning(pf,"maxsessions option appears %s"
```

```
            "more than once");
          }
          if (!*cp) {
            parser_error(pf,"maxsessions option requires a numeric
argument");
            got_error = 1;
          }
          else {
            if (*cp == '=') {
            int maxsessions;

            cp++;
            if (sscanf(cp, "%d", &(maxsessions)) == 1) {
              got_option_maxsessions = 1;
              if ( maxsessions < 3 ) {
                parser_error(pf,"invalid maxsessions value \"%s\" (must
be at least 3)",cp);
                got_error = 1;
              } else {
                pdp->pd_maxsessions = maxsessions;
              }
            } else {
              parser_error(pf,"unknown argument to maxsessions
\"%s\"",cp);
              got_error = 1;
            }
            }
            else {
            parser_error(pf,"unknown option \"%s\"",*tp);
            got_error = 1;
            }
          }
      } else if ( strncmp(*tp,"captured-ident",
                      strlen("captured-ident")) == 0 ) {
          char *cp;
          cp=(*tp)+strlen("captured-ident");
          if ( got_option_ci ) {
            parser_warning(pf,"captured-ident option appears %s"
            "more than once");
          }
          if (!*cp) {
            pdp->ci = 10;
            got_option_ci = 1;
          }
          else {
            if (*cp == '=') {
            cp++;
            if (sscanf(cp, "%d", &(pdp->ci)) == 1) {
              got_option_ci = 1;
            } else {
              parser_error(pf,"unknown argument to captured-ident
\"%s\"",*cp);
              got_error = 1;
            }
            }
            else {
            parser_error(pf,"unknown option \"%s\"",*tp);
            got_error = 1;
            }
```

```
        }
      } else if ( strncmp(*tp,"trusted-ident",
                      strlen("trusted-ident")) == 0 ) {
          char *cp;
          cp=(*tp)+strlen("trusted-ident");
          if ( got_option_ti ) {
            parser_warning(pf,"trusted-ident option appears %s"
            "more than once");
          }
          if (!*cp) {
            pdp->ti = 10;
            got_option_ti = 1;
          }
          else {
            if (*cp == '=') {
            cp++;
            if (sscanf(cp, "%d", &(pdp->ti)) == 1) {
              got_option_ti = 1;
            } else {
              parser_error(pf,"unknown argument to trusted-ident
\"%s\"",*cp);
              got_error = 1;
            }
            }
            else {
            parser_error(pf,"unknown option \"%s\"",*tp);
            got_error = 1;
            }
          }

      } else if ( strncmp(*tp,"untrusted-ident",
                      strlen("untrusted-ident")) == 0 ) {
          char *cp;
          cp=(*tp)+strlen("untrusted-ident");
          if ( got_option_ui ) {
            parser_warning(pf,"untrusted-ident option appears %s"
            "more than once");
          }
          if (!*cp) {
            pdp->ui = 10;
            got_option_ui = 1;
          }
          else {
            if (*cp == '=') {
            cp++;
            if (sscanf(cp, "%d", &(pdp->ui)) == 1) {
              got_option_ui = 1;
            } else {
              parser_error(pf,"unknown argument to untrusted-ident
\"%s\"",*cp);
              got_error = 1;
            }
            }
            else {
            parser_error(pf,"unknown option \"%s\"",*tp);
            got_error = 1;
            }
          }
      } else if ( strcmp(*tp,"honest") == 0 ) {
```

```
            if ( got_option_honest ) {
              parser_warning(pf,"honest option appears %s"
              "more than once");
            }
            pdp->pd_honest = 1;
            got_option_honest = 1;

        } else if ( strcmp(*tp,"ipnat") == 0 ) {

            if ( got_option_ipnat ) {
              parser_warning(pf,"ipnat option appears more than once");
            }
            pdp->pd_ipnat = 1;
            got_option_ipnat = 1;

        } else if ( strncmp(*tp,"ipnat-backchannel",
                        strlen("ipnat-backchannel")) == 0 ) {

            char *cp;

            cp=(*tp)+strlen("ipnat-backchannel");
            if (*cp == '=') {
              if ( got_option_ipnat_bc ) {
              parser_warning(pf,"ipnat-backchannel option appears more than
once");
              }
              cp++;
              got_option_ipnat_bc = 1;
              pdp->pd_ipnat_bc = strdup(cp);
              if (pdp->pd_ipnat_bc == NULL) {
                parser_error(pf,"Malloc failed while parsing");
                got_error=1;
              }
            }
            else {
              parser_error(pf,"Arrgh unknown option \"%s\"",*tp);
              got_error = 1;
            }
        } else if ( strcmp(*tp,"no-rdns") == 0 ) {

            if ( got_option_no_rdns ) {
              parser_warning(pf,"no-rdns option appears %s"
              "more than once");
            } else if ( got_option_no_t_rdns || got_option_no_u_rdns
            || got_option_no_c_rdns ) {
              parser_warning(pf,"no-rdns option overrides %s",
              "more specific no-xx-rdns option(s)");
            }
            pdp->pd_no_t_rdns = 1;
            pdp->pd_no_u_rdns = 1;
            pdp->pd_no_c_rdns = 1;
            got_option_no_rdns = 1;

        } else if ( strcmp(*tp,"no-trusted-rdns") == 0 ) {

            if ( got_option_no_rdns ) {
              parser_warning(pf,"previous no-rdns option makes this %s"
              "no-trusted-rdns option redundant");
```

```
            } else if ( got_option_no_t_rdns ) {
              parser_warning(pf,"no-trusted-rdns option appears %s",
              "more than once");
            }
            pdp->pd_no_t_rdns = 1;
            got_option_no_t_rdns = 1;

        } else if ( strcmp(*tp,"no-untrusted-rdns") == 0 ) {

            if ( got_option_no_rdns ) {
              parser_warning(pf,"previous no-rdns option makes this %s"
              "no-untrusted-rdns option redundant");
            } else if ( got_option_no_u_rdns ) {
              parser_warning(pf,"no-untrusted-rdns option appears %s",
              "more than once");
            }
            pdp->pd_no_u_rdns = 1;
            got_option_no_u_rdns = 1;

        } else if ( strcmp(*tp,"no-captured-rdns") == 0 ) {

            if ( got_option_no_rdns ) {
              parser_warning(pf,"previous no-rdns option makes this %s"
              "no-captured-rdns option redundant");
            } else if ( got_option_no_c_rdns ) {
              parser_warning(pf,"no-captured-rdns option appears %s",
              "more than once");
            }
            pdp->pd_no_c_rdns = 1;
            got_option_no_c_rdns = 1;

        } else {
            parser_error(pf,"unknown option \"%s\"",*tp);
            got_error = 1;

        }

    }

    if ( got_error ) {
      return(1);
    } else {
      return(0);
    }
}

int
parse_proxy_proxycmd(parser_file_t *pf, int juniper_proxy,
proxy_cmd_desc_t **pcdpp)
{
    struct tokenized_line *tlp;
    proxy_cmd_desc_t *pcdp;
    char **pp;

    tlp = pf->pf_tokenized_line;

    *pcdpp = NULL;

    if ( tlp->tl_tokens[1] == 0 ) {
```

```
      parser_error(pf,"missing %s binary file name",
          juniper_proxy ? "proxy" : "daemon" );
      return(1);
    }

    pcdp = (proxy_cmd_desc_t *)calloc(sizeof(proxy_cmd_desc_t),1);

    if ( strcmp(tlp->tl_tokens[1],"NONE") == 0 ) {

      if ( tlp->tl_tokens[2] != NULL ) {
          parser_error(pf,"junk after NONE parameter");
          free_proxy_cmd_desc(pcdp);
          return(1);
      }
      free(pcdp);
      return(0);

    }

    if ( tlp->tl_tokens[2] == 0 ) {
      parser_error(pf,"missing %s name after binary file name",
          juniper_proxy ? "proxy" : "daemon" );
      free_proxy_cmd_desc(pcdp);
      return(1);
    }

    pcdp->pcd_binary = strdup(tlp->tl_tokens[1]);

    /*
     * Take the tokens from the tokenized line and use them as our
     * vector of parameters (after shifting them left two places).
     *
     * Note that 'taking the tokenized line tokens' is safe (i.e. it is
     * 'defined to work').
     */

    pcdp->pcd_parms = tlp->tl_tokens;
    tlp->tl_tokens = NULL;
    pp = pcdp->pcd_parms;
    while ( (*pp = pp[2]) != NULL ) {
      pp += 1;
    }

    /*
     * If the binary is PASSTHROUGH then this must be a juniper
     * proxy and there must be exactly one additional parm (i.e.
argv[0]).
     */

    if ( strcmp(pcdp->pcd_binary,"PASSTHROUGH") == 0
    ||   strcmp(pcdp->pcd_binary,"PASSTHRU") == 0 ) {
      if ( !juniper_proxy ) {
          parser_error(pf,"PASSTHROUGH is only allowed on captured-proxy
lines");
          free_proxy_cmd_desc(pcdp);
          return(1);
      }
      if ( pcdp->pcd_parms[1] != NULL ) {
          parser_error(pf,"no parameters allowed for PASSTHROUGH proxy");
```

```
            free_proxy_cmd_desc(pcdp);
            return(1);
          }
          pcdp->pcd_passthrough = 1;
        }


        /* OK, If we're gonna run it let's remember what its
         * md5 sum looks like, as well as a stat of it.
         */
        if (!pcdp->pcd_passthrough) {
          int fd;
          if ((fd = open(pcdp->pcd_binary, O_RDONLY)) < 0) {
            parser_error(pf,"Can't open %s for reading", pcdp->pcd_binary);
            free_proxy_cmd_desc(pcdp);
            return(1);
          }
          pcdp->pcd_binary_md5 = strdup(md5_file(fd));
          if (pcdp->pcd_binary_md5 == NULL) {
            parser_error(pf,"Oops. malloc said no!");
            free_proxy_cmd_desc(pcdp);
            return(1);
          }
          pcdp->pcd_binary_stat = (struct stat *)malloc(sizeof(struct stat));
          if (pcdp->pcd_binary_stat == NULL) {
            parser_error(pf,"Oops. malloc said no!");
            free_proxy_cmd_desc(pcdp);
            return(1);
          }
          if (fstat(fd, pcdp->pcd_binary_stat) != 0) {
            parser_error(pf,"Couldn't stat %s! reason : %s",
                        pcdp->pcd_binary, strerror(errno));
            free_proxy_cmd_desc(pcdp);
            return(1);
          }
        }

        *pcdpp = pcdp;
        return(0);
    }

    proxy_desc_t *
    parse_proxy_description(parser_file_t *pf)
    {
        struct tokenized_line *tlp;
        proxy_desc_t *pdp;
        int got_error = 0;
        int got_port = 0;
        int got_captured_proxy = 0;
        int got_trusted_daemon = 0;
        int got_untrusted_daemon = 0;
        int got_username = 0;
        int got_end_proxy = 0;
        int got_options = 0;
        int rval;

        tlp = pf->pf_tokenized_line;
        if ( tlp->tl_tokens[1] == NULL ) {
          parser_error(pf,"missing proxy name on proxy line");
```

```
      return(NULL);
    }

    pdp = (proxy_desc_t *)calloc(sizeof(proxy_desc_t),1);
    pdp->pd_name = strdup(tlp->tl_tokens[1]);
    pdp->pd_lnum = pf->pf_lnum;
    pdp->ci = pdp->ui = pdp->ti = 0;

    if ( tlp->tl_tokens[2] == NULL ) {
      pdp->pd_protocol = IPPROTO_TCP;
    } else {
      if ( strcmp(tlp->tl_tokens[2],"tcp") == 0
      ||   strcmp(tlp->tl_tokens[2],"tcp/ip") == 0 ) {
          pdp->pd_protocol = IPPROTO_TCP;
      } else if ( strcmp(tlp->tl_tokens[2],"udp") == 0
      ||        strcmp(tlp->tl_tokens[2],"udp/ip") == 0 ) {
          parser_error(pf,"udp protocol not supported (yet)");
          free_proxy_desc(pdp);
          return(NULL);
      } else {
          parser_error(pf,"unknown protocol");
          free_proxy_desc(pdp);
          return(NULL);
      }
      if ( tlp->tl_tokens[3] != NULL ) {
          parser_error(pf,"junk after protocol type on proxy line");
          free_proxy_desc(pdp);
          return(NULL);
      }
    }

    for (
    pf->pf_goterror = 0;
    (rval = tokenize_line(pf)) == 0;
    pf->pf_goterror = 0 ) {

      tlp = pf->pf_tokenized_line;

      if ( strcmp(tlp->tl_tokens[0],"end-proxy") == 0 ) {

          if ( tlp->tl_tokens[1] == NULL ) {
            got_end_proxy = 1;
            break;
          } else {
            parser_error(pf,"junk after end-proxy keyword");
            free_proxy_desc(pdp);
            return(NULL);
          }

      } else if ( strcmp(tlp->tl_tokens[0],"port") == 0 ) {

          if ( got_port ) {
            parser_error(pf,"more than one port line in proxy
description");
            got_error = 1;
          } else if ( parse_proxy_port(pf,pdp) != 0 ) {
            got_error = 1;
          }
          got_port = 1;
```

```
      } else if ( strcmp(tlp->tl_tokens[0],"username") == 0 ) {

          if ( got_username ) {
            parser_error(pf,"more than one username line in proxy
description");
            got_error = 1;
          } else if ( parse_proxy_username(pf,pdp) != 0 ) {
            got_error = 1;
          }
          got_username = 1;

      } else if ( strcmp(tlp->tl_tokens[0],"options") == 0 ) {

          if ( parse_proxy_options(pf,pdp,got_options) != 0 ) {
            got_error = 1;
          }
          got_options = 1;

      } else if ( strcmp(tlp->tl_tokens[0],"trusted-daemon") == 0 ) {

          if ( got_trusted_daemon ) {
            parser_error(pf,"more than one trusted-daemon line in proxy
description");
            got_error = 1;
          } else {
            if ( parse_proxy_proxycmd(pf,0,&pdp->pd_trusted_daemon) != 0
) {
                got_error = 1;
            }
          }
          got_trusted_daemon = 1;

      } else if ( strcmp(tlp->tl_tokens[0],"captured-proxy") == 0 ) {

          if ( got_captured_proxy ) {
            parser_error(pf,"more than one captured-proxy line in proxy
description");
            got_error = 1;
          } else {
            if ( parse_proxy_proxycmd(pf,1,&pdp->pd_captured_proxy) != 0
) {
                got_error = 1;
            }
          }
          got_captured_proxy = 1;

      } else if ( strcmp(tlp->tl_tokens[0],"untrusted-daemon") == 0 ) {

          if ( got_untrusted_daemon ) {
            parser_error(pf,"more than one untrusted-daemon line in proxy
description");
            got_error = 1;
          } else {
            if ( parse_proxy_proxycmd(pf,1,&pdp->pd_untrusted_daemon) !=
0 ) {
                got_error = 1;
            }
          }
```

```
          got_untrusted_daemon = 1;

      } else {

          parser_error(pf,"unknown keyword in proxy description starting
on line %d",pdp->pd_lnum);
          got_error = 1;

      }

    }

    if ( !got_port ) {
      parser_error(pf,"missing port line in proxy description");
      got_error = 1;
    }

    if ( !got_end_proxy ) {
      parser_error(pf,"unexpected EOF in proxy description");
      got_error = 1;
    }

    if ( !got_username ) {
      struct passwd *pwp;
      pdp->pd_username = strdup("nobody");
      pwp = getpwnam("nobody");
      if ( pwp == NULL ) {
          parser_error(pf,"can't find \"nobody\" in the passwd
database");
          got_error = 1;
      }
      pdp->pd_uid = pwp->pw_uid;
      pdp->pd_gid = pwp->pw_gid;
    }

    if ( !got_captured_proxy ) {
      parser_warning(pf,"captured-proxy defaulted to NONE");
      pdp->pd_captured_proxy = NULL;
    }

    if ( !got_trusted_daemon ) {
      parser_warning(pf,"trusted-daemon defaulted to NONE");
      pdp->pd_trusted_daemon = NULL;
    }

    if ( !got_untrusted_daemon ) {
      parser_warning(pf,"untrusted-daemon defaulted to NONE");
      pdp->pd_untrusted_daemon = NULL;
    }

    if ( (!got_captured_proxy) && (!got_trusted_daemon) &&
(!got_untrusted_daemon) ) {
      parser_warning(pf,"no proxy programs defined in proxy
description");
    }

    if ( got_error ) {
      free_proxy_desc(pdp);
      return(NULL);
```

```
    }

    if ( pdp->pd_wildports && ntohs(pdp->pd_port) >= IPPORT_RESERVED ) {
      parser_error(pf,"wildports option may only be specified on
privileged ports (i.e. port number < %d)",IPPORT_RESERVED);
      got_error = 1;
    }

    return(pdp);
}

conf_file_t *
parse_conf_file(parser_file_t *pf)
{
    conf_file_t *cfp;
    tokenized_line_t *tlp;
    int max_proxies, next_proxy;
    int max_trusted_ifs, next_trusted_if;
    int rval;

    cfp = (conf_file_t *)calloc(sizeof(conf_file_t),1);

    max_proxies = 50;
    cfp->jc_proxies = (proxy_desc_t **)malloc(sizeof(proxy_desc_t) *
max_proxies);
    next_proxy = 0;
    cfp->jc_proxies[next_proxy] = NULL;

    max_trusted_ifs = 50;
    cfp->jc_trusted_ifs = (char **)malloc(sizeof(char *) *
max_trusted_ifs);
    next_trusted_if = 0;
    cfp->jc_trusted_ifs[next_trusted_if] = NULL;

    /*
     * Start parsing lines
     */

    for ( ;
    (rval = tokenize_line(pf)) == 0;
    pf->pf_goterror = 0 ) {

      tlp = pf->pf_tokenized_line;

      if ( strcmp(tlp->tl_tokens[0],"trusted-if") == 0 ) {
          char *trusted_if;

          trusted_if = parse_trusted_if(pf);

          if ( trusted_if != NULL ) {
            char **pp;

            for ( pp = &cfp->jc_trusted_ifs[0]; *pp != NULL; pp += 1 ) {
                if ( strcmp(*pp,trusted_if) == 0 ) {
                  break;
                }
            }
            if ( *pp == NULL ) {
```

```
                cfp->jc_trusted_ifs[next_trusted_if++] = trusted_if;
                if ( next_trusted_if == max_trusted_ifs ) {
                  max_trusted_ifs += 10;
                  cfp->jc_trusted_ifs = (char **)realloc(
                      cfp->jc_trusted_ifs,
                      sizeof(char *) * max_trusted_ifs);
                }
                cfp->jc_trusted_ifs[next_trusted_if] = NULL;

            } else {

                parser_error(pf,"interface \"%s\" is already defined as a
trusted interface",trusted_if);
                free(trusted_if);

            }

          }

      } else if ( 0 /* strcmp(tlp->tl_tokens[0],"proxy-directory") == 0
*/ ) {
          /*
           * The proxy-directory keyword is being deleted.
           * Disable it for now.
           */

          char *proxy_directory;

          proxy_directory = parse_proxy_directory(pf);

          if ( cfp->jc_proxy_directory != NULL ) {

            parser_error(pf,"extra \"proxy-directory\" line ignored");

          } else if ( proxy_directory != NULL ) {

            cfp->jc_proxy_directory = proxy_directory;

          }

      } else     if (strcmp(tlp->tl_tokens[0], "untrusted-acl") == 0) {
        if ((tlp->tl_ntokens) != 2) {
          parser_error(pf,"untrusted-acl must be followed by one
filename");
          pf->pf_goterror = 1;
        }
        else {
          if (untrusted_acl_file != NULL) {
            /* shouldn't happen since we only get called once, but
             * what the heck..
             */
            free(untrusted_acl_file);
          }
          untrusted_acl_file = strdup(tlp->tl_tokens[1]);
        }
      } else if (strcmp(tlp->tl_tokens[0], "trusted-acl") == 0) {
        if ((tlp->tl_ntokens) != 2) {
          parser_error(pf,"trusted-acl must be followed by one
filename");
```

```
          pf->pf_goterror = 1;
        }
        else {
          if (trusted_acl_file != NULL) {
            free(trusted_acl_file);
          }
          trusted_acl_file = strdup(tlp->tl_tokens[1]);
        }
      } else if (strcmp(tlp->tl_tokens[0], "captured-acl") == 0) {
        if ((tlp->tl_ntokens) != 2) {
          parser_error(pf,"captured-acl must be followed by one
filename");
          pf->pf_goterror = 1;
        }
        else {
          if (captured_acl_file != NULL) {
            free(captured_acl_file);
          }
          captured_acl_file = strdup(tlp->tl_tokens[1]);
        }
      } else if ( strcmp(tlp->tl_tokens[0],"proxy") == 0 ) {
          proxy_desc_t *pdp;

          pdp = parse_proxy_description(pf);

          if ( pdp != NULL ) {
            proxy_desc_t **pdpp;

            for ( pdpp = cfp->jc_proxies; *pdpp != NULL; pdpp += 1 ) {
                if ( strcmp((*pdpp)->pd_name,pdp->pd_name) == 0 ) {
                  parser_error(pf,"proxy \"%s\" already defined on line
%d",pdp->pd_name,(*pdpp)->pd_lnum);
                  break;
                }
                if ( (*pdpp)->pd_port == pdp->pd_port ) {
                  parser_error(pf,"proxy port %d already used by proxy
\"%s\" defined on line %d",ntohs(pdp->pd_port),(*pdpp)->pd_name,(*pdpp)-
>pd_lnum);
                  break;
                }
            }

            if ( *pdpp != NULL ) {
                free_proxy_desc(pdp);
                pdp = NULL;
            } else {

                cfp->jc_proxies[next_proxy++] = pdp;
                if ( next_proxy == max_proxies ) {
                  max_proxies += 10;
                  cfp->jc_proxies = (proxy_desc_t **)realloc(
                      cfp->jc_proxies,
                      sizeof(proxy_desc_t *) * max_proxies);
                }
                cfp->jc_proxies[next_proxy] = NULL;
            }

          }
```

```
      } else {

          parser_error(pf,"unknown keyword \"%s\"",tlp->tl_tokens[0]);

      }

    }

#if 0
    if ( next_trusted_if == 0 ) {
      parser_error(pf,"there must be at least one trusted interface
defined");
    }
#endif

    if ( next_proxy == 0 ) {
      parser_error(pf,"there must be at least one proxy defined");
    }

    if ( pf->pf_error_count > 0 ) {
      free_conf_file(cfp);
      cfp = NULL;
    }

    return(cfp);
}

void
dump_proxy_cmd( proxy_cmd_desc_t *pcdp, const char *name )
{
    if ( pcdp == NULL ) {
      printf("\t%s NONE\n",name);
    } else {
      char **pp;

      printf("\t%s %s",name,pcdp->pcd_binary);
      for ( pp = pcdp->pcd_parms; *pp != NULL; pp += 1 ) {
          printf(" %s",*pp);
      }
      printf("\n");
    }
}

void
dump_proxy_desc( proxy_desc_t *pdp )
{
    printf("proxy %s",pdp->pd_name);
    if ( pdp->pd_protocol == IPPROTO_TCP ) {
      printf(" tcp");
    } else {
      fprintf(stderr,"proxy has illegal protocol value %d\n",pdp-
>pd_protocol);
      abort();
    }
    printf("\n");
    printf("\tport %d\n",ntohs(pdp->pd_port));
    printf("\tusername %s # uid %d gid %d\n",pdp->pd_username,
    (int)pdp->pd_uid,(int)pdp->pd_gid);
    dump_proxy_cmd(pdp->pd_captured_proxy,"captured-proxy");
```

```
    dump_proxy_cmd(pdp->pd_trusted_daemon,"trusted-daemon");
    dump_proxy_cmd(pdp->pd_untrusted_daemon,"untrusted-daemon");
    printf("end-proxy\n");
}

/*
 * Dump out the parsed configuration file structure
 */

void
dump_conf_file( conf_file_t *cfp )
{
    proxy_desc_t **pdpp;
    char **tifpp;

    if ( cfp->jc_proxy_directory != 0 ) {
      printf("proxy-directory %s\n",cfp->jc_proxy_directory);
    }

    for ( pdpp = cfp->jc_proxies; *pdpp != NULL; pdpp += 1 ) {
      dump_proxy_desc(*pdpp);
    }

    for ( tifpp = cfp->jc_trusted_ifs; *tifpp != NULL; tifpp += 1 ) {
      printf("trusted-if %s\n",*tifpp);
    }
}
```

LISTA DE PLACAS DE REDE COMPATÍVEIS

Abaixo, segue a lista de placas de rede compatíveis para Linux:

Placas PCI

DEC 21x4x

ENI/Adaptec ATM

3Com 3c59x

3Com 90x[B]

SMC EPIC/100 Ethernet

Essential HIPPI card

DEC DEFPA FDDI

Intel EtherExpress PRO PCnet-PCI Ethernet

NE2000 Compatível

SiS 900 Ethernet

ThunderLAN Ethernet

DECchip 21x4x Ethernet

VIA Rhine Fast Ethernet

Lan Media Corp SSI/HSSI/DS3Realtek 8129/8139

Placas ISA

AT1700

CS8900 Ethernet

3Com 3c503

3C505

3C501

3C509

3C507

StarLAN

FMV-180 series

EtherExpress/16

EtherExpress 10 ISA DEC EtherWORKS III

DEPCA

NE2100

BICC IsoLan

NE[12]000 ethernet

SMC91C9x Ethernet

IBM TROPIC (Token-Ring)

IBM TROPIC (Token-Ring)

3COM TROPIC (Token-Ring)

WD/SMC Ethernet

Placas PCMCIA

BayStack 650 (802.11FH)

Xircom/Netwave AirSurfer

3Com 3c589 e 3c562

MB8696x based Ethernet

NE2000-compatível Raytheon Raylink (802.11)

Megahertz Ethernet

Lucent WaveLan IEEE (802.11)

Xircom CreditCard Ethernet.

MÓDULOS DE REDE DISPONÍVEIS

Os módulos de rede disponíveis no arquivo **/etc/rc.d/rc.modules**, no Linux, são:

 # Ethernet cards based on the 8390 chip.

3com 3c503 support:

#/sbin/modprobe 3c503

 # Ansel Communications EISA 3200 support:

#/sbin/modprobe ac3200

Cabletron E21xx support:

#/sbin/modprobe e2100

 # HP PCLAN+ (27247B and 27252A) support:

#/sbin/modprobe hp-plus

 # HP PCLAN (27245 and other 27xxx series) support:

#/sbin/modprobe hp

 # NE2000/NE1000 support (non PCI):

#/sbin/modprobe ne io=0x300 # NE2000 at 0x300

#/sbin/modprobe ne io=0x280 # NE2000 at 0x280

#/sbin/modprobe ne io=0x320 # NE2000 at 0x320

#/sbin/modprobe ne io=0x340 # NE2000 at 0x340

#/sbin/modprobe ne io=0x360 # NE2000 at 0x360

 # PCI NE2000 clone support:

#/sbin/modprobe ne2k-pci

 # SMC Ultra support:

#/sbin/modprobe smc-ultra

 # SMC Ultra32 EISA support:

#/sbin/modprobe smc-ultra32

 # Western Digital WD80*3 (and clones) support:

#/sbin/modprobe wd

#

 # Other network hardware drivers:

#

 # 3com 3c501 (consider buying a new card, since the 3c501 is slow, broken, and obsolete):

#/sbin/modprobe 3c501

3com 3c503:

#/sbin/modprobe 3c503

 # 3com 3c505:

#/sbin/modprobe 3c505

 # 3com 3c507:

#/sbin/modprobe 3c507

 # 3com 3c509 and 3c579:

/sbin/modprobe 3c509

 # 3com 3c515:

#/sbin/modprobe 3c515

 # This one works for all 3com 3c590/3c592/3c595/3c597 and the

 # EtherLink XL 3c900 and 3c905 cards:

#/sbin/modprobe 3c59x

 # Apricot Xen-II on board Ethernet:

#/sbin/modprobe apricot

 # Generic ARCnet support:

#/sbin/modprobe arcnet

 # AT1700/1720 support:

#/sbin/modprobe at1700

 # AT-LAN-TEC/RealTek pocket adapter support:

#/sbin/modprobe atp

 # BPQ Ethernet driver:

#/sbin/modprobe bpqether

 # Generic DECchip & DIGITAL EtherWORKS PCI/EISA:

#/sbin/modprobe de4x5

 # D-Link DE600 pocket adapter support:

#/sbin/modprobe de600

 # D-Link DE620 pocket adapter support:

#/sbin/modprobe de620

 # DEPCA support:

#/sbin/modprobe depca

 # Digi International RightSwitch cards:

#/sbin/modprobe dgrs

 # Intel EtherExpress Pro support:

#/sbin/modprobe eepro

 # Intel EtherExpress PRO/100 PCI support:

#/sbin/modprobe eepro100

 # Intel EtherExpress16 support:

#/sbin/modprobe eexpress

 # SMC EtherPower II 9432 PCI support:

#/sbin/modprobe epic100

 # ICL EtherTeam 16i/32 support:

#/sbin/modprobe eth16i

 # DEC EtherWorks 3 support:

#/sbin/modprobe ewrk3

 # Fujitsu FMV-181/182/183/184 support:

#/sbin/modprobe fmv18x

 # HP 10/100VG PCLAN (ISA, EISA, PCI) support:

#/sbin/modprobe hp100

 # IBM Tropic chipset based adapter support:

#/sbin/modprobe ibmtr

 # AMD LANCE and PCnet (AT1500 and NE2100) support:

#/sbin/modprobe lance

 # NI5210 support:

#/sbin/modprobe ni52

 # NI6510 support:

#/sbin/modprobe ni65

 # AMD PCnet32 (VLB and PCI) support:

#/sbin/modprobe pcnet32

 # Red Creek Hardware Virtual Private Network (VPN) support:

#/sbin/modprobe rcpci

 # RealTek 8129/8139 (not 8019/8029!) support:

/sbin/modprobe rtl8139

 # Sangoma S502A FRAD support:

#/sbin/modprobe sdla

 # SMC 9194 support:

#/sbin/modprobe smc9194

 # DECchip Tulip (dc21x4x) PCI support:

#/sbin/modprobe tulip

 # VIA Rhine support:

#/sbin/modprobe via-rhine

 # AT&T WaveLAN & DEC RoamAbout DS support:

#/sbin/modprobe wavelan

 # Packet Engines Yellowfin Gigabit-NIC support:

#/sbin/modprobe yellowfin

QUANTOS FREEBSD *HACKERS* SÃO NECESSÁRIOS PARA TROCAR UMA LÂMPADA?

Mil cento e sessenta e nove:

Vinte e três para reclamarem no -CURRENT que estão sem luz;

Quatro para dizer que é um problema na configuração, e que essa pergunta deveria ser feita na freebsd-questions;

Três para enviar Relatório de Problemas sobre a lâmpada, dos quais ao menos um, não está completamente concluído, e consiste apenas de um breve “ta escuro”;

Um para adicionar uma lâmpada que nunca foi testada, que danifica todo o buildworld e depois de 5 minutos tem que ser retirada;

Oito para reclamarem para os autores dos Relatórios de Problemas por não ter incluído correções em seus relatórios;

Cinco para reclamar que o buildworld não está funcionando;

Trinta e um para responder que funciona para eles, e que os problemáticos devem ter feito CVSup na hora errada;

Um para enviar uma correção para a nova lâmpada na freebsd-hackers;

Um para reclamar que ele tinha correções para essa lâmpada há 3 anos, mas que quando elas foram enviadas para o -CURRENT, foram simplesmente ignoradas, e que sua experiência com o sistema de Relatório de Problemas não foram as melhores possíveis; além disso a nova lâmpada proposta

não era reflexiva;

Trinta e sete para gritarem em alto e bom som que as lâmpadas não fazem parte da base do sistema, e que os desenvolvedores não tem o direito de sair fazendo esse tipo de coisa sem antes consultar a comunidade, e O QUE O -CORE ESTA FAZENDO SOBRE ISSO!?

Duzentos para reclamar da cor do quartinho de bicicletas;

Três para dizer que a correção enviada não está de acordo com os padrões que o código do *kernel* deve ter, conforme documentado na página de manual do style(9);

Dezessete para reclamar que a nova lâmpada proposta está licenciada sob a Licença Pública Geral GNU (GPL);

Quinhentos e oitenta e seis para entrarem de corpo e alma em uma discussão sobre as vantagens comparativas entre a licença Pública Geral GNU (GPL), a licença BSD, a licença do MIT, a NPL e a higiene pessoal dos fundadores da *Free Software Foundation*;

Sete para copiar vários trechos da discussão para a lista de discussão freebsd-chat e para a freebsd-advocacy;

Um para trocar a nova lâmpada sugerida, apesar de a nova brilhar bem menos que a antiga;

Dois para retirarem a lâmpada, furiosos, dizendo que o FreeBSD está melhor no escuro do que com uma lâmpada tão fraca;

Quarenta e seis para contestarem vorazmente sobre a retirada da lâmpada fraca e escreverem um relatório para o -core;

Onze para dar a idéia de criar uma lâmpada menorzinha, que poderia caber no Tamagotchi deles, se um dia nós decidirmos portar o FreeBSD para tal plataforma;

Setenta e três para reclamar da razão sinal versus ruído na freebsd-chat e na freebsd-hackers, e se retirarem das listas em protesto;

Treze para enviarem mensagens com o conteúdo "unsubscribe", "Como eu saio da lista?", ou "Por

favor, me tirem da lista", seguidas do rodapé tradicional do servidor de discussão com as instruções para sair da lista;

Um, para adicionar uma nova lâmpada que funciona bem, enquanto todos os outros estão ocupados demais com a discussão para perceber que alguém já trocou a lâmpada por uma funcional;

Trinta e um para afirmar que a nova lâmpada brilha em média 0.364% mais, se comparada com as lâmpadas TenDRA (contudo, ela terá que ser refeita em formato de cubo) e que o FreeBSD deveria mudar para TenDRA ao invés do GCC;

Um para reclamar que a nova lâmpada não é honesta;

Nove (incluindo aqueles que enviaram os Relatórios de Problemas) para perguntar “o que significa MFC?”; (Merged From-CURRENT)

Cinqüenta e sete para reclamar que ficaram no escuro por duas semanas até que a lâmpada fosse trocada.

Um adendo do Nik Clayton <nik@FreeBSD.org>:

....

Aí pensei, "Peraí, não deveria ter ao menos 1 para documentar a nova lâmpada em algum lugar?”

Daí eu fui iluminado :-)

(Autor desconhecido) [PRO – 10].

7 BIBLIOGRAFIA

[ASC – 01] ASCENSO, Marina Faria e SANTOS, Pedro Miguel de Sá. **História e Desenvolvimento do Sistema Operativo UNIX**. http://www.eq.uc.pt/~pmg3/unix1.htm. 2001. consulta em 17/08/2003.

[BAR – 02] BARAHONA, Jesús M. Gonzáles [at all]. **Notes on libre software**. http://projects.openresources.com/libresoft-notes/libresoft-notes-en/. Consultado em 15/09/2003.

[CEV – 03] CEVOLI, Paul. **Embedded FreeBSD CookBook**. Burlington: Elsevier Science. 229 p. 2002.

[DOG – 04] dog@FreeBSD.org. **FreeBSD Handbook**. The FreeBSD Documentation Project. 1999. www.freebsd.org.

[DUP – 05] DUPRE, Alex. **Filtering Bridges**. http://www.freebsd.org/doc/en_US.ISO8859-1/articles/filtering-bridges/article.html. Acesso em 21/10/2003.

[FBS – 06] Introduction To FreeBSD – An Absolute Beginners Guide To FreeBSD. 1997. fbsd-book@vmunix.com.

[FIL – 07] FILHO, Claudio Ferreira. **OpenOffice.org Source Project: Projeto Brasil para o**

Brasil. 2003. http://br-pt.openoffice.org/faglic.html. Acesso em 24/09/2003.

[HEU – 08] HEUER, Dr. Konrad. **FreeBSD – Die ersten Schritte**. Göttingen: Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG). 2001.

[LEH – 09] LEHEY, Greg. **The Complete FreeBSD®**. Third Edition. Walnut Creek CDROM. 2001.

[LOR – 10] LORENZZONI, Pablo. **Pré-História da Revolução**. 2001. http://people.debian.org/~spectra/files/revolução.txt

[LUC – 11] LUCAS, Michael. **Absolute BSD – The Ultimate Guide to FreeBSD**. No Starch Press: San Francisco. 2002.

[MOR – 12] MORIMOTO, Carlos E. **Entendendo e Dominando o Linux**. http://www.guiadohardware.net/ . 2003.

[NAK - 13] NAKANISHI, Seido. **Cercado de Linux**. Revista do Linux. 2002. Ano III. N° 26. Pg. 20.

[PON – 14] PONT, Luciana Dal, CRISTIANO, Marta Adriana da Silva. **LAMSTER To Scientific Research**. Monografia. UNISUL. 2001.

[PRO – 15] Projeto de Documentação do FreeBSD. **Perguntas Mais Freqüentes Sobre FreeBSD 2.X, 3.X e 4.X**. FreeBSD: doc/pt_BR.ISO8859-1/books/faq/book.sgml, v1.1. 2002.

[SIM – 16] Simpson, Bruce M. [et all]. **Project ClosedBSD**. http://www.closedbsd.org. 2002.

[WEI – 17] WEINBERG, Bill. **Linux Inside**. Revista Linux User. 2001.

[VLE – 18] VLECK, Tom Van. **MULTICS: History**. 2003. http://ftp.stratus.com/vos/multics/tvv/history.html. Consulta em 24/09/2003.

[ZAN – 19] ZANAROTTI, Stan. **MULTICS**. http://www.mit.edu:8001/afs/net/user/srz/www/multics.html. Acesso em 24/09/2003.

[JUN-20] JUNIPER NETWORKS. **Japan's HOTnet Selects Juniper Networks for New 10-Gigabit MPLS Network**, http://www.juniper.net.company/presscenter/pr/2003/pr-031111.html. Acesso em 12/11/2003.