ADALBERTO DINIZ DE SOUZA

ADMINISTRAÇÃO DE SERVIÇOS COM SEGURANÇA EM SERVIDORES LINUX

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

Orientador

Prof. Luiz Henrique Andrade Correia

LAVRAS
MINAS GERAIS – BRASIL
2001

ADALBERTO DINIZ DE SOUZA

ADMINISTRAÇÃO DE SERVIÇOS COM SEGURANÇA EM SERVIDORES LINUX

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências da disciplina Projeto Orientado para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em __ de _____ de 2001

_____
Prof. Jones Oliveira Albuquerque DCC

_____
Prof. Reginaldo Ferreira de Souza DAE

_____
Prof. Luiz Henrique Andrade Correia
DCC
(Orientador)

LAVRAS
MINAS GERAIS - BRASIL

Agradecimentos

A Deus.

Ao meu filho que me ajuda a crescer em cada momento que estamos juntos.

Aos meus pais, que corajosos sempre acreditaram em mim.

A minha namorada, que mesmo quando não compreendia tentava me apoiar.

Aos meus amigos Leonardo Camargos e Lucas Bueno.

A minha avó Maria Diniz, que sempre tenho junto a meu coração.

Resumo

Este trabalho trata os procedimentos para a instalação e administração de servidores mediante a implantação de alguns serviços dos inúmeros existentes no universo das redes de computadores e internet. O trabalho advem de uma pesquisa realizada na Universidade Federal de Lavras, especificamente do Departamento de Administração e Economia. O enfoque dado ao trabalho gesta por segurança a usuários e serviços que sejam pretendidos a um servidor, principalmente se este servidor funcionar com sistema operacional Linux. São tratados aspectos desde a escolha dos serviços e suas causas e conseqüências até o usuário final, com a obtenção e gestão das contas de usuários. É abordado, quanto a usuário: a escolha de senhas, como o usuário vê uma política e como deveria ver. A descrição de serviços e ferramentas de gestão é uma constante, bem como suas implicações e algumas limitações. Conclui-se o trabalho apresentando uma avaliação das dificuldades e soluções para estas.

a Gabriel Ribeiro de Souza

Sumário

Lista de Figuras **Página**

Lista de Tabelas **Página**

1. Introdução

Tendo o objetivo de estudar a Internet, sua evolução e aplicação, na Universidade Federal de Lavras, pode-se projetar uma expansão, a curto prazo, da rede.

Fisicamente, é possível verificar que a rede da universidade está praticamente pronta, existindo somente alguns pontos a serem concluídos, o que será necessidade expandir na universidade nos próximos meses, e talvez anos, será os serviços disponíveis na universidade, pois cada um dos departamentos da universidade possui um servidor hospedado em suas dependências, que possui a função de funcionar com um *gateway*. Um *gateway* é muito bem descrito por seu próprio nome, é um portão para que os computadores ligados em rede, que estão em cada departamento, possam ter acesso a internet. É claro que este serviço é uma evolução e uma qualificação presente a universidade, porém, a vontade por expansão da prestação de serviços caminha como a evolução de tecnologia, e a real expansão esta em um patamar, ainda, muito lento.

Com exceção do Departamento de Ciência da Computação da universidade, os outros departamentos ainda não possuem serviços de e-mail local, FTP (*File Transfer Protocol* - Protocolo de Transferência de Arquivos), *homepage* e outros. Por observar esta dificuldade, no Departamento de Administração e Economia, foi elaborado este trabalho que visa dar alicerces a esta implantação, como descrito no capítulo 2.

Uma vez que a principal necessidade no departamento de Administração e Economia é a criação de correio eletrônico, com o domínio do departamento,

foi definida esta linha de pesquisa, um documento que trate os caminhos para a implantação de correio eletrônico. Porém, criar somente por criar, de certa forma, não é tão difícil, talvez um técnico bem preparado, com cursos específicos, seja perfeitamente capaz desta função, por isso foi abordado um aspecto científico, que está presente a partir do capítulo 3, uma política para implantação e gestão dos serviços, que proteja o administrador, os usuários e o próprio sistema. É descrito abordagens sobre se o melhor é deixar tudo restrito ou tudo muito fechado, como no capítulo 4. Também como administrar senhas, processo aparentemente tão simples, por sua propagação através dos meios tecnológicos, e no dia a dia da sociedade, que utiliza tecnologia, como apresentado no capítulo 5. No capítulo 6 é apresentado como foram feitas a implementações e no capítulo 7 as conclusões.

2. Proposição

O início do trabalho foi baseado na realização de uma pesquisa sobre os serviços disponibilizados em servidores, a qual a razão ficará explicita durante a leitura do texto, especificamente a partir do capítulo 4. Esta pesquisa foi realizada observando as necessidade serviços e aplicações que os usuários do Departamento de Administração e Economia da Universidade Federal de Lavras possuem, e ponderado de maneira a abranger e ser aplicado com o menor índice de restrições possíveis, determinou-se quais serviços seriam apresentados e tratados no documento.

A necessidade maior dos usuários está em possuir um serviço de correio eletrônico que os identificasse melhor com a universidade e sobretudo ao departamento, pois a maioria já possui endereço de correio eletrônico da universidade, para isso serão citadas algumas características principais dos *softwares*. Para se manter contas de correio eletrônico que possam ser confiáveis é preciso realizar ainda outros controles, como definir uma quota, de espaço de disco, para cada usuário, para que não ocorra de um usuário descuidado prejudicar outro através do recebimento e não leitura de suas mensagens, para isso será utilizado o software *Quotas*. Até aqui talvez tenha feito uma proteção mais ao administrador, que está livre de reclamações quanto ao grande espaço que um determinado usuário está usando e pouco espaço que outro usuário possui para usar, então também será tratado de uma política de contas, chamada *accounting*, que começará a determinar restrições aos usuários e com isso dar um primeiro passo a implementação de segurança. Para estes serviços é importante possuir uma autenticação e estabelecer um controle quanto ao acesso físico do servidor e acesso a conta de usuário para isso será utilizado o serviço PAM (*Pluggable Authentication Modules*). Para que possivelmente

possa escolher-se pessoas para realizar a função de sub-administrador do sistema, quando se fizer necessário, outro *software* foi escolhido, o *Sudo*.

Bem, tudo isso é muito importante, mas se não houver um ponto de apoio para implementar e determinar estes serviços, normas de condutas para os usuários e para os administradores, a possibilidade de eficiência, e até a segurança do sistema estará muito ameaçada, por isso antes de tudo é necessário uma política de segurança.

Os serviços serão disponibilizados em servidores com Sistema Operacional Linux, onde as observações e referencias serão feitas sobre este, especificamente a distribuição Red Hat 6.2 e 7.1. A escolha de um Sistema Operacional para a função de servidor não pode ser feita ao acaso, esta deve estar fundamentada em compatibilidades, funcionalidades e segurança, características encontradas no sistema em questão.

No departamento de Administração e Economia, foram feitas as instalações dos aplicativos para colher as análises que ajudarão nos fundamentos para este documento.

3. Política de Segurança

3.1. O que é uma Política de Segurança e por que ter uma?

As decisões relacionadas a segurança que se toma, ou não, como administrador em grande parte determina quanto seguro ou inseguro é a sua rede, a funcionalidade que sua rede oferece, e como fácil sua rede será de usar [6]. Porém, você não pode tomar boas decisões sobre segurança sem determinar primeiro quais são suas metas de segurança.

Até que se determinem quais são suas metas de segurança, não pode-se fazer uso efetivo de qualquer coleção de ferramentas de segurança porque simplesmente não se sabe o que conferir e que restrições impor.

Suas metas serão determinadas em grande parte pelas comparações a seguir [29]:

(1) Serviços oferecidos versus Segurança provida:

Cada serviço oferecido aos usuários leva seus próprios riscos de segurança. Para alguns serviços o risco excede o valor do benefício do serviço e o administrador pode escolher eliminar o serviço em lugar de tentar torná-lo seguro.

(2) Facilidade de uso versus Segurança:

Quanto mais simples o sistema é de usar, ele permite acesso sem restrições a qualquer usuário e não requer nenhuma senha; quer dizer, não haveria nenhuma segurança. A requisição senhas faz o sistema um pouco menos conveniente, ou seja, mais difícil de usar, porém mais seguro.

(3) Custo de segurança versus Risco de perda:

Há muitos custos envolvidos na segurança: monetário (o custo de comprar hardware de segurança e software de *firewalls*), desempenho (encriptação e desencriptação levam tempo de processamento), e facilidade de uso (como mencionado acima). Também há muitos níveis de risco: perda de privacidade (a leitura de informação por indivíduos sem autorização), perda de dados (a corrupção ou perda de informação), e a perda de serviço (o preenchimento de espaço de armazenamento de dados, uso de recursos computacionais, e restrição de acesso de rede). Cada tipo de custo deve ser pesado contra cada tipo de perda.

Suas metas devem ser comunicadas a todos os usuários, equipe de operação, e gerentes por um conjunto de regras de segurança, chamando política de segurança.

3.1.1. Definição de uma Política de Segurança

Uma política de segurança é uma declaração formal das regras pelos quais as pessoas que utilizarão determinada tecnologia de uma organização devem se submeter.

3.1.2. Propósitos de uma Política de Segurança

O propósito principal de uma política de segurança é informar os usuários, equipe e gerentes das obrigações deles para proteger tecnologia e informações. A política deve especificar os mecanismos pelos quais estas exigências podem ser satisfeitas. Outro propósito é prover uma base pela qual possa adquirir, configurar e examinar sistemas de computador e redes para

submissão à política. Então uma tentativa para usar um conjunto de ferramentas de segurança, como softwares adequados ou mesmo medidas administrativas ponderadas, na ausência de pelo menos uma política de segurança é incluída sem sentido.

Uma Política de Uso Apropriada (*Appropriate Use Policy* - AUP) [12] também pode ser parte de uma política de segurança. Deve mostrar aos usuários o que podem ou não podem fazer nos vários componentes do sistema, inclusive o tipo de tráfego permitido nas redes. O AUP deveria ser tão explícito quanto possível para evitar ambigüidade ou mal entendido.

3.1.3. Quem deve ser envolvido quando cria-se uma política de segurança?

Para que uma política de segurança seja apropriada e efetiva, precisa-se ter a aceitação e apoio de todos os níveis de funcionários dentro da organização. É especialmente importante que a administração apóie completamente o processo de política de segurança caso contrário haverá pouca chance que elas tenhão o impacto planejado ou esperado. Um outro aspecto é uma lista de pessoas que devem ser envolvidas na criação e revisão de documentos de política de segurança [2][5]:

(1) Administrador responsável.

(2) Equipe técnica de informática (por exemplo, a equipe do centro de informática).

(3) Os administradores de grandes grupos de usuário dentro da organização (por exemplo, divisões empresariais, departamento de informática dentro de uma universidade, etc.).

(4) Os representantes dos grupos de usuário afetados pela política de segurança.

(5) Pessoal responsável pela segurança física da infra-estrutura.

(6) O administrador de segurança do sistema.

(7) Aconselhamento jurídico (se apropriado).

A lista acima é representativo em muitas organizações, mas não é necessariamente inclusivo, ou seja, nem todas as empresas possuem esta estrutura tão bem definida ou mesmo existente. A idéia é trazer representação para riscos fundamentais, administração, equipe técnica que sabe o que pode e não pode ser apoiado, e aconselhamento legal que conhece as ramificações legais de várias escolhas de política.

3.2. O que faz uma Política de Segurança ser boa?

As características de uma boa política de segurança são:

(1) deve ser implementável por procedimentos de administração de sistemas, publicando por diretrizes de uso aceitáveis, ou outros métodos apropriados.

(2) deve ser verificada através de ferramentas de segurança, quando apropriadas, e com sanções aos usuários e administradores, quando prevenção não é tecnicamente possível.

(3) tem que definir claramente as áreas de responsabilidade para os usuários, administradores.

Os componentes de uma boa política de segurança incluem [29]:

(1) Uma Política de Privacidade que define expectativas razoáveis de sigilo que considerando acesso aos arquivos de usuários, e monitoramento de correio eletrônico.

(2) Uma Política de Acesso que define direitos de acesso e privilégios para proteger a perda, especificando diretrizes de uso aceitáveis para usuários, equipe de operação e administração. Deve prover diretrizes para conexões externas, comunicações de dados, dispositivos conectados a uma rede, e acrescentar novos softwares ao sistema. Também deve especificar qualquer mensagem de notificação (por exemplo, mensagens de conexão deve prover advertências sobre autorização de uso e linha de monitoramento, e não simplesmente dizer "*Welcome*").

(3) Uma Política de Responsabilidade que define as responsabilidades de usuários, equipe de operação, e administração, Deve especificar uma capacidade de auditoria, e prover diretrizes de manipulação de incidentes, por exemplo, o que fazer e quem contactar se uma possível invasão é descoberta.

(4) Uma Política de Autenticação que estabelece confiança por uma política de senha efetiva, e estabelecendo diretrizes para autenticação de local remoto e o uso de dispositivos de autenticação.

(5) Uma Declaração de Disponibilidade que fixa as expectativas de usuários para a disponibilidade de recursos. Também especificar horas operacionais e períodos de manutenção.

(6) Um Sistema de Informática e Política de Manutenção de Rede que descreve como são permitidas para as pessoas de manutenção internas e externas controlar e tecnologia de acesso. Um tópico importante a ser analisado aqui é se existirá manutenção remota, se será permitida e como este acesso será controlado. Outra área a ser considerada é como será administrado quando se contratar uma empresa para prestar determinados serviços para a empresa.

(7) Uma Política que Informa Violações que indica quais os tipos de violações (por exemplo, privacidade e segurança, interno e externo) deve ser informado e a quem são feitos os relatórios. Uma atmosfera não ameaçadora e a possibilidade de informar anônimo resultarão em uma maior probabilidade que uma violação seja informada se for descoberta.

(8) Informação de Apoio que provê os usuários, equipe e administração informações de contato para cada tipo de violação de política; diretrizes em como controlar estas questões sobre um incidente de segurança, ou informação que pode ser considerada confidencial; e cruzar referências para procedimentos de segurança e informações relacionadas, como políticas de companhia e leis governamentais e regulamentos.

Pode haver exigências reguladoras que afetam alguns aspectos de sua política de segurança. Os criadores da política de segurança devem considerar ajuda jurídica na criação da política. No mínimo, a política deve ser revisada através de deliberação legal.

Uma vez que sua política de segurança foi estabelecida deve ser comunicado claramente aos usuários, equipe e administração. Tendo todo o registro de usuários através de uma declaração que indica que eles leram,

compreenderam, e concordaram em cumprir a política, que é uma parte importante do processo. Finalmente, sua política deve ser revisada em uma base regular para ver se está apropriada em suas necessidades de segurança.

3.3. O que mantêm a Política Flexível

Para que uma política de segurança seja viável por um longo tempo, flexível fundamentada em um forte conceito de segurança. Uma política de segurança deve ser (em grande parte) independente de hardware específico e situações de software. Os mecanismos para atualizar a política devem ser escritos claramente. Isto inclui o processo, as pessoas envolvidas e as pessoas que devem sinalizar as mudanças.

Também é importante reconhecer que há exceções a toda regra. Sempre que possível, a política deve ressaltar quais exceções existem à política geral. Por exemplo, até que condições é permitido um administrador de sistema passar pelos arquivos de um usuário. Também, pode haver alguns casos quando múltiplos usuários terão acesso ao mesmo *UserId* (Identificador de Usuário). Por exemplo, em sistemas com um usuário *root* (administrador), múltiplos administradores de sistemas podem saber a senha e podem usar a conta de root, ou, neste caso, procurar métodos que evitem tal ocorrência.

Outra consideração é chamada a "Síndrome do Caminhão de Lixo" [12]. Isto mostra o que aconteceria em uma empresa, se, por exemplo, uma pessoa fundamental a um serviço repentinamente estivesse indisponível para trabalhar, como por exemplo, ficou repentinamente doente ou saiu da companhia inesperadamente. Enquanto a maior segurança residir na disseminação mínima de informação, o risco de perder informações críticas aumenta quando aquela

informação não é compartilhada. É importante determinar o próprio equilíbrio para sua empresa.

3.4. Planos de Segurança completamente definidos

Todos os locais devem definir um plano de segurança que inclua todos funcionários. Este plano deve estar a um nível mais alto que as políticas específicas discutidas neste capítulo, e deve ser feito sobre uma base de diretrizes amplas nas quais políticas específicas se ajustarão.

É importante ter uma base em lugar de políticas individuais precisa ser consistente com a arquitetura geral de segurança. Por exemplo, tendo uma política forte com respeito a acesso de Internet e restrições fracas ao permitir o uso de modem são incompatíveis com uma filosofia global de restrições de segurança fortes em acesso externo.

Um plano de segurança deve definir: a lista de serviços que a rede proverá; quais áreas da organização proverão os serviços; quem terá acesso a esses serviços; como acesso será provido; quem administrará esses serviços; etc.

O plano também deve se dirigir como um incidente deverá ser controlado. É importante que cada local defina classes de incidentes e respostas correspondentes. Por exemplo, locais com *firewalls* devem fixar um limite no número de tentativas para anular (segurar) o *firewall* antes de ativar uma resposta? Devem ser definidos níveis de escalonameto para ataques e respostas. Locais sem *firewalls* terão que determinar se uma única tentativa para conectar a um servidor constituir um incidente?

Para locais conectados à Internet, o excessivo aumento na mídia de relatos sobre incidentes de segurança na Internet podem obscurecer um problema potencial de segurança interno mais sério. Igualmente, companhias que nunca foram conectadas à Internet podem ter forte políticas internas, mas falha para uma política de conexão externa.

4. Separação de Serviços

Existem muitos serviços que uma empresa pode desejar prover aos seus usuários, alguns dos quais podem ser externos. Há uma variedade de razões de segurança para tentar isolar serviços sobre computadores dedicados a servidores. Também há razões de desempenho na maioria dos casos, mas uma discussão detalhada está fora do escopo deste documento.

Os serviços que um local pode prover tem, na maioria dos casos, níveis diferentes de necessidades de acesso e modelos de confiança. Serviços que são essenciais à segurança ou operação de um local seriam melhores se colocados em uma máquina dedicada com acesso muito limitado [veja Seção 4.1, o modelo "negar tudo"], em lugar de em uma máquina que provê um serviço(s) que esteve tradicionalmente menos seguro, ou requer maior acessibilidade por usuários que podem comprometer segurança acidentalmente.

Também é importante distinguir entre *hosts* que operam dentro de modelos diferentes de confiança (por exemplo, todos os *hosts* dentro de um *firewall* e qualquer *host* em uma rede exposta).

Alguns dos serviços que devem ser examinados para separação potencial são esboçados na seção 4.3.2 é importante se lembrar que segurança só é tão forte a medida que existe uma ligação mais fraca na cadeia.

Várias das invasões publicadas atualmente foram pela exploração de vulnerabilidade em sistemas de correio eletrônico. Os intrusos não estavam tentando roubar correio eletrônico, mas eles usaram a vulnerabilidade naquele serviço para ganhar acesso a outros sistemas e arquivos.

Se possível, cada serviço deveria estar rodando em uma máquina diferente cujo seu dever é prover um serviço específico. Isto ajuda isolar os intrusos e limitar dano potencial.

4.1. Negar tudo / Permitir tudo

Há duas filosofias subjacentes e diametralmente opostas que podem ser adotadas ao definir um plano de segurança. Ambas as alternativas são modelos legais para se adotar, e a escolha entre eles dependerá do local e suas necessidades por segurança.

A primeira opção é tirar todos os serviços e então selecionar e habilitar serviços caso a caso quando são necessários. Isto pode ser feito no servidor ou no nível de rede como apropriado, chamado "negar tudo".

O outro modelo, que aqui é chamado o modelo "permitir tudo", é muito mais fácil de implementar, mas geralmente é menos seguro que o modelo "negar tudo". Simplesmente ative todos os serviços, normalmente a falta no nível de servidor, e permitir que todos os protocolos viagem pelos limites da rede, até o roteador. Como buracos de segurança ficam aparentes, eles são restringidos ou consertados em outros servidores ou no nível de rede.

O modelo "negar tudo", é geralmente mais seguro que o outro modelo descrito no parágrafo anterior. Mais trabalho é exigido para implementar eficientemente uma configuração "negar tudo" como também entender melhor de serviços. Permitindo provê só serviços conhecidos dentro de uma melhor

análise do serviço em particular e o desenvolvimento de um mecanismo de segurança servida no nível de segurança do local.

Cada um destes modelos podem ser aplicados a porções diferentes setores, enquanto dependendo de exigências de funcionalidade, controle administrativo, política de local, etc. Por exemplo, a política pode usar o modelo "permitir tudo" ao montar *workstations* para uso geral, mas adota um modelo "negar tudo" ao montar servidores de informação, como o de e-mail. Igualmente, uma política "permitir tudo" pode ser adotada para tráfego entre LAN (*Local Área Network*) interno ao local, mas uma política "negar tudo" pode ser adotada entre o local e a Internet.

Deve-se ter cuidado ao misturar filosofias como nos exemplos anteriores. Muitos locais adotam a teoria de política rígida em um meio flexível. Eles estão dispostos a pagar o custo de segurança pelo tráfego externo e requererem medidas de segurança forte, mas estão pouco dispostos ou incapazes de proverem proteções semelhantes interiormente. Isto trabalha com pressuposto de que as defesas exteriores nunca sejam quebradas e que os usuários internos podem ser confiáveis. Uma vez a passagem pelo *firewall* é rompida, atingir os servidores de rede é trivial.

4.2. Identificar Reais Necessidades de Serviços a Usuários

Há uma grande variedade de serviços que podem ser providos, ambos internamente ou na Internet. Administrar segurança é, em muitas formas, administrar acesso a serviços internos ao local e administrar como os usuários internos têm acesso a informação em locais remotos.

Serviços tendem a surgir como ondas na Internet. Durante os anos muitos locais estabeleceram servidores de FTP anônimos, servidores de gopher, servidores de WWW, etc. Como eles ficaram populares, mas não particularmente necessários em todos os locais. Avalie todos novos serviços que são mostrados com uma atitude cética para determinar se lhes são necessários ou são moda passageira que varre a Internet.

Tenha em mente que complexidade de segurança pode crescer exponencialmente com o número de serviços providos. Filtros em roteadores precisam ser modificados para apoiar os novos protocolos. Alguns protocolos são difíceis para filtrar seguramente (por exemplo, serviço RPC - Remote Procedure Control e UDP - User Datagram Protocol), provendo mais aberturas assim à rede interna. Serviços providos na mesma máquina podem interagir de modos catastróficos. Por exemplo, permitindo FTP anônimo na mesma máquina com o servidor de WWW pode permitir para um intruso colocar um arquivo na área de FTP anônimo e usar o servidor de HTTP para executar.

4.3. Rede e Configuração de Serviço

4.3.1. Protegendo a Infra-estrutura

Muitos administradores de rede vão longe para proteger os servidores de suas redes e outros administradores pouco fazem qualquer esforço para proteger suas redes. Há alguma razão para isto? Por exemplo, é fácil de proteger um servidor de uma rede. Também, é provável que os intrusos busquem dados nos servidores; danificando a rede para servir a seus propósitos [veja Apêndice C]. Isso dito, há ainda razões para proteger as redes. Por exemplo, um intruso poderia desviar tráfico de rede por um servidor externo para examinar os dados

(procurar senhas). Também, infra-estrutura inclui mais que as redes e os roteadores que os interligam. Infra-estrutura também inclui administração de rede (por exemplo, SNMP), serviços (por exemplo, DNS, NFS, WWW), e segurança (autenticação de usuário e restrições de acesso)[veja Apêndice D].

A infra-estrutura também precisa de proteção contra erro humano. Quando um administrador erra na configuração de um servidor, aquele servidor pode oferecer serviço com defeito. Isto só afeta usuários que requerem aquele servidor e, a menos que aquele servidor seja um servidor primário, o número de usuários afetados será limitado. Porém, se um roteador for configurado errado, serão afetados todos os usuários que requisitarem a rede. Obviamente, este é um número muito maior de usuários que dependem de qualquer um servidor.

4.3.2. Protegendo os Serviços

Há muitos tipos de serviços e cada um tem suas próprias exigências de segurança. Estas exigências variarão baseado no uso planejado do serviço. Por exemplo, um serviço que só deveria ser utilizável dentro de um local (por exemplo, NFS) pode requerer mecanismos de proteção diferentes que um serviço provido para uso externo. Pode ser suficiente proteger o servidor interno de acesso externo. Porém, um servidor de WWW que provê um *homepage* pode ser acessado em qualquer lugar por usuários na Internet, requer proteção embutida. Quer dizer, o serviço/protocolo/servidor têm que prover qualquer segurança para prevenir acesso sem autorização e modificação do banco de dados de *Web*.

Serviços internos (serviços significam que só podem ser usados por usuários dentro de um local) e serviços externos (criado deliberadamente para

usuários fora de um local), em geral, têm exigências de proteção que diferem das previamente descritas. É então conhecido que para isolar os serviços internos o administrador conecte um conjunto de computadores de servidores e os serviços externos a outro conjunto de computadores servidores. Quer dizer, não deveriam ser co-localizados servidores internos e externos no mesmo computador servidor. Na realidade, muitos locais vão tão distantes sobre ter uma pessoa conectada a *subnets* (ou até mesmo redes diferentes) que são acessíveis do exterior e outro conjunto que só podem fazer acesso localmente. Claro que, normalmente há um *firewall* que conecta estas partições. Deve ser tomado grande cuidado para assegurar que aquele *firewall* está operando corretamente.

Há um interesse crescente em usar *intranets* para conectar partes diferentes de uma organização (por exemplo, divisões de uma companhia). Enquanto este documento geralmente diferencia entre externo e interno (o público e privado), locais que usam *intranets* deveriam estar atentos que eles precisarão considerar três separações e objeto pegando ações apropriadas quando projetando e oferecendo serviços. Um serviço oferecido a uma intranet nem não seria público, nem tão completamente privado quanto um serviço a uma única sub-unidade organizacional. Então, o serviço precisaria ser executado em seu próprio sistema, separado de serviços externos e internos e redes.

Uma forma de serviço externo merece consideração especial, é o acesso anônimo, ou *guest*. Estes podem ser FTP anônimo ou *login* de convidado (não autenticado). É extremamente importante assegurar aqueles servidores de FTP anônimos e identificadores de *login* de convidados estão cuidadosamente isolado de qualquer servidor e sistemas de arquivo fora de usuários deveria ser mantido. Outra área para a qual deve ser dada atenção especial é onde são mantidos

arquivos de usuários e o acesso de escrita através de acesso anônimo. Um local pode ser legalmente responsável pelo conteúdo que disponibiliza publicamente informação, monitoramento cuidadoso da informação depositada por usuários anônimos é aconselhável.

Agora nós consideraremos alguns dos serviços mais populares: serviço de senha/chave, serviço de autenticação, correio eletrônico, WWW. Considerando que estes são os serviços freqüentemente usados, eles são os pontos mais óbvios de ataque. Também, um ataque com sucesso em um destes serviços pode produzir um desastre total, fora de proporção.

4.3.2.1. Correio eletrônico

Sistemas de correio eletrônico (e-mail) foram durante muito tempo uma fonte para grandes pontos de invasão porque protocolos de e-mail estão entre os mais velhos e amplamente desdobraram outros serviços. Também, por isto está mesma natureza, um servidor de e-mail requer acesso para o mundo externo; a maioria dos servidores de e-mail aceitam contribuição de qualquer fonte. Um servidor de e-mail geralmente consiste em duas partes: agente de *receiving/sending* e um agente de processo. Desde que e-mail é entregue a todos os usuários, e é normalmente privado, o agente de processo requer uso do sistema com privilégios (*root*) para entregar o correio. A maioria das implementações de e-mail executam ambas as porções do serviço que também significa que o agente receptor tem privilégios no sistema. Isto abre vários furos de segurança que devem ser tratados configurando o *sendmail* [veja Apêndice C]. Há algumas implementações disponíveis que permitem uma separação dos dois agentes. Tais implementações geralmente são considerados mais seguras,

mas ainda exige para instalação cuidadosa que evita criar um problema de segurança.

4.3.3. Protegendo a Proteção

Tenha certeza que freqüentemente se negligencia a segurança do próprio servidor de segurança, um dos locais de vulnerabilidade em potencial, deixa-o aberto a ataques. Baseado em considerações previamente discutidas, deveria estar claro que: o servidor de segurança não deve ser acessível externamente; deva oferecer acesso mínimo, com exceção da função de autenticação, para usuários locais; e não deve ser co-localizado com qualquer outro servidor [29]. Mais adiante, deveria ser anotado todo o acesso para o *host*, inclusive acesso para o próprio serviço, para prover um "rastro de papel" no caso de uma brecha de segurança.

5. Serviços e Procedimentos de segurança

Este capítulo mostra quais tópicos que devem ser observados ao se proteger um local. Cada seção menciona um serviço de segurança ou capacidades que podem ser exigidas para proteger as informações e sistemas em um local. Os tópicos são apresentados em alto-nível para introduzir ao leitor aos conceitos.

Ao longo deste capítulo, você achará menção significante de criptografia. Está fora da extensão deste documento mostrar em detalhes relativo a criptografia.

5.1. Autenticação

Por muitos anos, o método prescrito por autenticar os usuários foi pelo uso de padrão, senhas reutilizáveis. Originalmente, estas senhas eram usadas por usuários em terminais para se autenticar a um computador central. Na ocasião, não havia nenhuma rede (interiormente ou externamente), assim o risco de revelação da senha de texto clara era mínimo. Hoje, sistemas estão conectados junto a redes locais, e estas redes locais estão mais adiante conectadas juntas a Internet. Usuários estão logando por toda parte do mundo; as senhas reutilizáveis são transmitidas freqüentemente por essas mesmas redes em texto claro, fácil para qualquer intruso capturar. E realmente, o Centro de Coordenação CERT [29] e outras equipes de resposta estão vendo um tremendo número de incidentes que envolvem *sniffers* de pacote que está capturando as senhas de texto.

Com o advento de tecnologias mais novas como PGP e dispositivos de autenticação baseados em símbolo, as pessoas estão usando senha equivalentes a símbolos secretos e seguros. Se não são selecionados estes símbolos secretos e seguros corretamente e são protegidos, a autenticação será descoberta facilmente.

5.1.1. Kerberos

Kerberos é um sistema de segurança de redes distribuídas que provê autenticação para redes inseguras. Se solicitado pela aplicação, também podem ser providos de integridade e encriptação. *Kerberos* foi desenvolvido originalmente no *Massachusetts Institute of Technology* (MIT) em meados dos anos oitenta. Há duas versões principais de *Kerberos*, versão 4 e 5, que são para propósitos práticos e incompatíveis, como, por exemplo, a versão 4 que pode ser usada com o *sendmail*.

Kerberos se baseia em um banco de dados simétrico que usa um centro de distribuição fundamental (KDC) que é conhecido como o servidor de *Kerberos*. Um usuário ou serviço (conhecido como *principals*) é concedido por "ingressos" eletrônicos depois de comunicar corretamente com o KDC. Estes ingressos são usados para autenticação entre *principals*. Todos os ingressos incluem um selo de tempo que limita o período de tempo para o qual o ingresso é válido. Então, os clientes de *Kerberos* e servidor têm que ter uma fonte de tempo segura, e pode manter relação de tempo com precisão.

O lado prático de *Kerberos* é sua integração com o nível de aplicação. Aplicações típicas como FTP (*File Transfer Protocol*), telnet, POP (*Post Office Protocol*), e NFS foram integrados com o sistema de *Kerberos*. Há uma

variedade de implementação que têm níveis variados de integração. Veja [21] para mais recentes informações.

5.1.2. Escolhendo e Protegendo Símbolos Secretos

Ao selecionar símbolos secretos, se preocupe em os escolher cuidadosamente. A seleção de senhas, eles devem ser robustos contra algoritmos de força de bruta para os adivinhar. Quer dizer, eles não deveriam ser palavras simples em qualquer idioma, nada comum, indústria, ou siglas culturais, etc. Idealmente, eles serão mais longos em lugar de mais curto e eles consistem em frases de passagem que combinam caracteres maiúsculo e minúsculo, dígitos numéricos, e outros caracteres e metacaracteres.

Uma vez escolhido, a proteção destes símbolos secretos é muito importante. Alguns são usados como trancas para dispositivos de hardware (como cartões simbólicos) e estes não deveriam ser escritos abaixo ou colocaram no mesmo local como o dispositivo com que eles são associados. Outros, como uma chave secreta *Pretty Good Privacy* (PGP), deve ser protegido de acesso sem autorização.

Finalmente ao usar produtos de criptografia, como PGP, deve-se preocupar em determinar o próprio comprimento mínimo e assegurar que seus usuários são treinados para fazer igualmente. Com avanços de tecnologia, o comprimento mínimo da chave de segurança continua crescendo. Tenha certeza que seu servidor seja atualizado constantemente na tecnologia, de forma que você possa assegurar que qualquer criptografia está provendo a proteção que você acredita ter.

5.1.3. Garantia de senha

Enquanto não se elimina a necessidade do uso de um padrão, não podem ser exageradas em senhas reutilizáveis, é reconhecido que algumas organizações ainda podem os estar usando. Enquanto é recomendado que esta transição em organizações para o uso de tecnologia melhor, ainda em tempo, nós temos o seguinte conselho para ajudar com a seleção e manutenção de senhas tradicionais. Mas lembre-se, nenhuma destas medidas provê proteção contra revelação devido a programas de *sniffer*.

(1) A importância de senhas robustas [veja Apêndice A] - Em muitos casos de invasão de sistema, o intruso precisa ganhar acesso a uma conta no sistema. Um modo que é tipicamente realizado é adivinhando a senha de um usuário legítimo. Isto é freqüentemente realizado escolhendo uma senha padronizada, fácil em que com um programa que quebra esta senha, pois este utiliza um dicionário muito grande contra o arquivo de senha do sistema. O único modo para vigiar senhas que são descobertas desta maneira é pela seleção cuidadosa de senhas que não podem ser adivinhadas facilmente (combinações de números, cartas, e caráter de pontuação).

(2) Mudando senhas padrões - são instalados muitos sistemas operacionais e programas com contas e senhas padrões. Estes devem ser mudados imediatamente para algo que não possa ser adivinhado ou quebrado.

(3) Restringindo acesso ao arquivo de senha [veja Apêndice D]- em particular, um local quer proteger a porção de senha codificada em arquivo de forma que intrusos que não os tenha disponível para quebra. Uma técnica efetiva é usar senhas de sombra (*shadow password*) onde o campo de senha do arquivo padrão

contém um "bobo" ou falsa senha. O arquivo que contém as senhas legítimas é protegido em outro lugar no sistema.

(4) Senha que expira [veja Apêndice A]- Quando e como expirar senhas ainda é um assunto de controvérsia entre a comunidade de segurança. Geralmente é aceito que uma senha não deveria ser mantida em uma conta que possui muito tempo de uso, mas é debatido calorosamente se um usuário deveria ser forçado a mudar uma senha boa que está em uso. Os argumentos para senhas variáveis relacionam à prevenção do uso continuado de contas invadidas. Porém, as reivindicações de oposição que a senha freqüentemente muda conduzem a usuários que escrevem as senhas em áreas visíveis (como colar a um monitor), ou para usuários que selecionam senhas muito simples que são fáceis de adivinhar. Também deveria ser visto que um intruso provavelmente usará rapidamente uma senha capturada ou adivinhada, neste caso senha não oferece qualquer proteção.

Enquanto não houver nenhuma resposta definitiva a este dilema, deveria ser tratado por uma política de senha diretamente e prover diretrizes para com que freqüência um usuário deve mudar a senha. Certamente, uma mudança anual na senha normalmente não é difícil para a maioria dos usuários, e deve-se considerar esta possibilidade. É recomendado que senhas sejam mudadas pelo menos sempre que uma conta de um administrador seja descoberta, ou quando uma conta qualquer for descoberta. Além do mais, se uma senha de conta de um administrador for descoberta, devem ser mudadas todas as senhas no sistema.

(5) Bloqueio de conta através da senha - Talvez seja útil, bloquear contas depois de um número predefinido de tentativas que falham para autenticação. Se decidir empregar este mecanismo, é recomendado que o mecanismo não se

anuncie. Depois de bloquear, até mesmo se a senha correta é apresentada, a mensagem exibida deveria permanecer como uma tentativa de falhada de *login*. Implementando este mecanismo requererão aqueles usuários legítimos contatam o administrador de sistema para pedir que a conta deles seja reativada.

(6) O *daemon* [4] de *finger* - Por padrão, o *daemon* de *finger* exibe o sistema utilizado e informação de usuário. Por exemplo, pode exibir uma lista de todos os usuários que usam um sistema atualmente, ou todos os conteúdos do arquivo de *.plan* de um usuário específico. Estas informações podem ser usadas por intrusos que pretendem identificar *usernames* e adivinhar as senhas deles. É recomendado que se modifique o *finger* para restringir a informação exibida.

5.2. Confiança

Haverá informações que sua empresa desejará proteger de revelação a entidades sem autorização. Sistemas operacionais têm freqüentemente mecanismos de proteção de arquivo embutidos que permitem para um administrador controlar quem no sistema pode ter acesso, ou ver o conteúdo de um determinado arquivo. Um modo mais eficiente para prover confiança é por encriptação [29]. Encriptação é realizada misturando-se dados de forma a tornar muito difícil a descoberta e o tempo que consome para qualquer um, diferente dos usuários autorizados ou donos, obterem um texto claro. Os usuários autorizados e o dono da informação possuirão as chaves de desencriptação correspondentes que lhes permitem ordenar o texto facilmente para uma forma legível (texto claro). Nós recomendamos que locais usem encriptação para prover confiança e proteger informação valiosa.

O uso de encriptação às vezes é controlado por regulamentos de governos, assim nós encorajamos que os administradores sejam informados de leis ou políticas que regulamentam seu uso antes de emprega-la. Está fora da extensão deste documento discutir os vários algoritmos e programas disponíveis para este propósito, mas nós acautelamos contra o uso casual do UNIX cripta programa que pode ser quebrado facilmente. Nós também encorajamos todos para gastar algum tempo para entender o poder da encriptação em qualquer algoritmo. A maioria dos produtos famosos são bem documentados, assim deve ser uma tarefa bastante fácil.

5.3. Integridade [29]

Como um administrador, você desejará ter certeza sobre uma informação, por exemplo, arquivos do sistema operacional, dados de companhia, etc., se não foi alterado sem autorização. Isto significa você desejará prover alguma garantia sobre a integridade da informação em seus sistemas. Um modo para prover isto é produzir um *checksum* do arquivo inalterado, e periodicamente (ou quando desejar) conferir para ter certeza se o *checksum* do arquivo on-line não mudou (que indicaria se dados foram modificados).

Alguns sistemas operacionais possuem programas de *checksum*, como o UNIX. Porém, estes podem não prover a proteção que você precisa de fato. Podem ser modificados arquivos de tal forma que o programa não observe e preserve o resultado! Então, pode-se usar um programa de criptografia forte, tal com o método de criptografia MD5, pois este método torna o método de desencriptação extremamente trabalhoso, para um usuário não autorizado, em virtude do tamanho da chave gerada e tamanho da informação, para mais

informações ver [32], assim para produzir *checksums* que serão usados para assegurar integridade é o ideal.

Há outras aplicações onde integridade precisará ser assegurada, como ao transmitir uma mensagem de e-mail entre duas partes. Há produtos disponíveis para prover esta capacidade. Uma vez você identifica que esta é uma capacidade que você precisa, você pode continuar a identificar tecnologias que proverão isto.

5.4. Autorização

Autorização recorre ao processo de garantir privilégios a processos por usuários. Isto difere de autenticação naquela autenticação de processos usada para identificar um usuário. Uma vez identificado o processo (confiantemente), os privilégios, direitos, propriedade e ações permitidas do usuário são determinados através de autorização.

Listar as atividades autorizadas de cada usuário explicitar (e os processos de usuário) com respeito a todos os recursos (objetos) é impossível em um sistema razoável. Em um sistema real, certas técnicas são usadas para simplificar o processo de conceder e conferir autorização.

Uma aproximação, popularizada em sistemas de UNIX, é nomear a cada objeto três classes de usuário: o dono, grupo e mundo. O dono ou é o criador do objeto ou o usuário nomeado como dono pelo super-usuário. As permissões de dono (leitura, escrita e execução) só se aplicam ao dono. Um grupo é uma coleção de usuários que compartilham direitos de acesso a um objeto. As permissões de grupo (leitura, escrita e execução) se aplicam a todos os usuários

no grupo (menos o dono). O mundo corresponde a todo o mundo com acesso ao sistema. As permissões mundiais (leitura, escrita e execução) se aplicam a todos os usuários (menos o dono e membros do grupo).

Outra aproximação é prender a um objeto uma lista que explicitamente contém a identidade de todas as permissões de usuários (ou grupos). Esta é uma Lista de Controle de Acesso (*Access Control List* - ACL). A vantagem de ACLs é que elas são facilmente mantidas (uma lista central por objeto) e é muito fácil uma checagem visual para quem tem acesso a ela. As desvantagens são os recursos extras exigidos para armazenar tal lista, como também o número vasto de tal lista requerido para sistemas grandes.

5.5. Acesso

5.5.1. Acesso físico

A restrição de acesso físico a servidores, é necessária permitindo somente o acesso as pessoas que supostamente administram os servidores. Servidores incluem terminais confiáveis (terminais que permitem uso não autenticado como consoles de sistema, terminais de operador e terminais dedicados a tarefas especiais), e microcomputadores individuais e *workstations*, especialmente quando esses se conectam a rede. Torne as áreas de trabalho de pessoas seguras restringindo o acesso de pessoas; caso contrário eles acharão modos para evitar sua segurança física (por exemplo, arrombando portas).

Mantenha original e copia dos arquivos de configuração de seu servidor. Mantenha-os atualizados para propósito de auxilio, eles devem ser protegidos de

roubo. É importante manter as cópias em um local separado dos originais, não só considerando danos, mas também para controlar contra roubos.

Computadores portáteis são um risco em particular. Tenha certeza que não causará problemas se um computador portátil de seus usuários for roubado. Considere diretrizes para os tipos de dados que deveriam ser permitidos estar nos discos de computadores portáteis como também como os dados deveriam ser protegidos (por exemplo, encriptação) quando está em um computador portátil.

Outras áreas onde acesso físico deveria ser restrito são os armários de instalação elétrica e elementos de rede importantes como servidores de arquivo, servidor de nome, roteadores, e centro de cabeamento (*hubs* e *switches*).

6. Métodos

Comparando o custo de segurança versus o risco de perda pode-se determinar vários procedimentos de proteção. Quando o Departamento teve problemas elétricos criou-se um agravante ao custo monetário, pois a queima de equipamentos tornou-se uma constante, para isso foi gasto mais dinheiro, que a longo prazo já foi retornado e compensado. A aquisição de *no-break's* foi de fundamental importância, pois a rede elétrica pertence a universidade e o departamento não possui autonomia para altera-la como é necessário.

Um aspecto comum aos departamentos da Universidade Federal de Lavras, e encontrado no Departamento de Administração e Economia também, é a falta de interesse em aprender novas técnicas e ferramentas para proteção pessoal e proteção do serviço que o usuário tem a seu dispor, o acesso a Internet. Essa característica pode ser evitada nos departamentos que agora possuem Internet, como o Departamento de Medicina Veterinária, onde as pessoas responsáveis pela informatização querem implantar uma cultura de utilização dos recursos de maneira correta. Por isso, adotar ferramentas que requerem alteração drástica na maneira de receber uma informação, ou fazer uma operação, não é realizado de maneira trivial.

A evolução ocorrida no Departamento de Ciência da Computação, com alteração do interfaceamento do correio eletrônico de *pop3* para *imap*, não seria facilmente aceita em outros departamentos, e ainda penalizaria os administradores, em função de atendimento sobre dúvidas que chegarão desordenadas, ou seja, cada usuário de uma vez. Por isso a facilidade de uso é importante para a escolha das ferramentas utilizadas. Para evitar esse problema

específico ao correio eletrônico foi adotado o *pop3* no servidor do departamento, porque:

1- Os usuários já possuem correio eletrônico com esse interfaceamento.
2- Existe uma versão que possui transmissão através de tunelamento.
3- Há necessidade de manter contas com somente correio eletrônico, sem espaço em disco.

Tunelamento é a criação de um modelo semelhante a um túnel virtual, por onde é realizada a transmissão de dados, assim pode-se atingir um ótimo nível de segurança para o usuário e para o servidor, sem que necessite alterar o hábito do usuário. O pequeno inconveniente é o aumento do tempo de transmissão da informação e do processamento do hardware.

O risco de perda de privacidade envolvendo obtenção de informações de outros é explicita na política de uso e limitada através de software, no caso de correio eletrônico, por exemplo, que é o escopo deste trabalho. A possibilidade de criptografar todos os dados contidos nas contas dos usuários foi levada em consideração, pois assim um usuário que possui acesso ao servidor que quisesse ver o conteúdo da pasta de outro usuário não entenderia nada, caso essa pasta permitisse leitura para o usuário, porém limitações de hardware não possibilitarão está implementação, por isso não estudado.

A perda de um serviço também foi tratada durante a pesquisa, pois os usuários têm a sua disposição uma área para armazenagem de arquivos, ou seja, um disco virtual. Este disco virtual é atrelado a conta de correio eletrônico e por isso foi protegido.

Primeiro foi feito o levantamento do disco disponível, sua capacidade de armazenamento. Verificou-se que os tamanhos das partições existentes eram inadequadas a prestação do serviço de correio eletrônico, então foi projetado para quantas pessoas seria disponibilizado o serviço. Foi definido que os professores, os alunos de pós-graduação, os alunos de graduação, as secretarias e foi reservado algum espaço para futuras contas, se necessário. Como não pode-se fechar os olhos diante a possibilidade de futuramente estes possuírem *homepage*, o dimensionamento foi estabelecido assim em winchester de oito *gibabytes* (8GB):

Partições	Tamanhos
swap	131MB
/boot	31MB
/	1066MB
/home	2455MB
/home/ftp	462MB
/var	2251MB
/usr	2172MB
/tmp	101MB

Tabela 6.1 - Partições

As partições que exigem um planejamento maior são o */home*, onde será armazenado os arquivos do disco virtual, e o */var*, onde são armazenadas as mensagens de correio eletrônico, quando estas chegam, porque o diretório fica em */var/spool/mail/USER*. Como o diretório padrão do Linux para recebimento de correio é este preferiu-se não alterar por não saber os riscos quanto a interpretação que o software *sendmail* [ver Apêndice C] daria a esta atitude, mesmo sendo o *sendmail* um software robusto, aplicado com eficiência em pequenas e grandes aplicações, se configurado no mínimo o necessário [veja Apêndice C].

O tamanho para cada usuário ficou assim:

Usuários	/home	/var	Quantidade Reservada	Quantidade Existente
Professores	12MB	5MB	35 pessoas	31 pessoas
Pós-graduação	8MB	5MB	70 pessoas	60 pessoas
Alunos Graduação	5MB	3MB	260 pessoas	250 pessoas
Secretarias e Anexos	8MB	5MB	6 pessoas	5 pessoas
Somente e-mail	0MB	3MB	11 pessoas	7 pessoas
Futuros	135MB	350MB	pessoas	pessoas

Tabela 6.2 – Espaço para cada grupo de usuários

Para que um usuário não interferisse no espaço destinado a outro, fazendo com isso que o serviço fosse suspenso a outro usuário foi realizado a limitação de espaço em disco para cada grupo e usuário com a adoção do software *Quotas* (ver Apêndice B), que realiza o controle de quanto de espaço pode ser consumido por um usuário ou um grupo. Uma vantagem do *Quotas* é que este software realiza o controle de espaço baseado na propriedade do arquivo, independendo de onde esteja, é seguro e não muito difícil de configurar.

Uma política que supervisione como os usuários e os administradores se comportam, suas responsabilidades e seus deveres tem que ser apresentada junto a disponibilidade dos serviços. Esta política deve proteger os usuários de responsabilidades que não conseguirão cumprir, e não tem interesse em cumprir, e o administrador, que não pode ser penalizado se um usuário resolve manter materiais proibidos, ou inapropriados, em seu disco virtual, ou utilize seu endereço de correio eletrônico para incomodar ou mandar informações impróprias. Então esta política é sobretudo um acordo de conduta diante o serviço e as atribuições e penalidades jurídicas que podem ser aplicadas somente a quem não estiver se comportando corretamente.

Não só foram utilizadas ferramentas que auxiliam a criação dos serviços, como também foram escolhidas ferramentas que auxiliam o controle e manutenção das contas, para que o administrador não tenha que se preocupar com uma senha muito boba, por exemplo.

Baseada na política de segurança, um gerenciamento de conta, conhecido e referenciado no meio por *accountig manegement* foi estabelecido. Neste módulo do Sistema Operacional Linux, foi utilizado a interface "*linuxconf*", para configuração de:

1- Qual o tamanho mínimo da senha.
2- Quantos caracteres numéricos é exigido na senha, se é que algum é exigido.
3- Configuração de uma quota padrão, caso o administrador esqueça de atribuir uma, ou um grupo à nova conta.

Foi estabelecido que a senha deve ter 10 (dez) caracteres no mínimo, sendo no mínimo um (1) numérico. O tamanho do *login* restrito a no máximo 10 caracteres, preferencialmente, letras.

A quota padrão ficou sendo como a definida para os professores, tanto na partição */home* quanto na */var*. Assim ninguém seria prejudicado se uma desatenção fosse cometida na hora de criar a conta.

A outra ferramenta adotada foi o serviço PAM (*Pluggable Authentication Modules*), que provê um auxílio ao gerenciamento visando a autenticação [veja Apêndice D], com isso este serviço não impede que o sistema seja invadido, mais constroem algumas barreiras, como não permitir que o

usuário execute o comando *reboot* e reinicie o servidor. Ou ainda limitar os dias e horários que se possa fazer FTP.

O mais importante, diante o escopo deste trabalho, é impedir que o usuário desligue ou reinicie a máquina, limitar recursos de hardware disponíveis a grupos e/ou usuários, definir horários que determinados usuários podem se logar, avisar quando o usuário tem mensagem nova. Como descrito no [veja Apêndice D], existem muitos outros módulos que são importantes a servidores, como FTP, por exemplo.

Uma última ferramenta diretamente utilizada, foi o *Sudo*, que casa com segurança ao servidor, e é importante para a criação de sub-administradores para o sistema, pois assim pode-se delegar responsabilidades, aos interessados, lembrando que não é necessário ficar fazendo uso da senha do administrador, senha esta que quanto menos for utilizada melhor.

O que é função do administrador de rede, quanto a aquisição de software, instalação e configuração está concluído. Sendo a política de segurança a etapa que requer atenção e necessita mobilização dos membros [veja seção em 3.1.3]. Reunir o supervisor de informática, o responsável técnico, o chefe do departamento, o coordenador de curso e neste caso, esperar o aval da procuradoria da Universidade Federal de Lavras.

7. Conclusão

Atualização é uma palavra muito forte e presente a nomenclatura do profissional da área de informática. Estar atualizado requer evolução de pensamento e atitude pois a velocidade que surgem novos problemas é enorme.

O profissional que mantiver a iniciativa em suas diretrizes de atuação profissional estará se mantendo útil ao mercado. Sobre este ponto de dista é que este documento foi concebido e deste pode-se retirar várias interpretações para conclusões.

A utilização de ferramentas para auxiliar o administrador é fundamental para a permanência de um serviço em funcionamento. Porém, ouvir falar em uma ferramenta e não fazer um estudo sobre esta antes de fazer uso no servidor é inocência do administrador, além disso, pode estar levando a um erro de proteção, uma vez que em redes de computadores algo mal configurado é caminho para possíveis invasões.

Segundo este documento, o gerenciamento de contas, que é uma atitude preventiva e necessária, deve ser casado ao serviço de PAM. Um aplicativo é com certeza o complemento do outro.

A facilidade de usar um serviço é um aspecto importante quanto as técnicas de usabilidade de um sistema. Clareza, eficiência, atalhos bem projetados, mensagens bem escritas são fundamentais para que o usuário fique satisfeito quanto ao sistema que utiliza, recomendado-o e se sentindo seguro.

Um usuário seguro está mais disposto a colaborar com o sistema, faça este usuário satisfeito que seu empenho em melhorar como o sistema pode ser estimulado, e não focará só dizem: Porque vou querer aprender isso? Esta atitude deve estar fundamentada na política escolhida. Uma política de segurança deve ser baseada no usuário, como a escolha de programas que sejam amigáveis, por isso a opção por escolher o *Sendmail*, um sistema robusto, integrável aos principais sistemas operacionais e ferramentas de acesso a rede e internet, rico em documentação e difícil de configurar.

Apesar de pouco citado neste texto, o protocolo que casa a interface do programa utilizado para leitura de correio eletrônico com o servidor, o *POP3*, é muito importante, novamente, devido aspectos de usabilidade, a capacidade de retenção de informações, o hábito de operação do usuário não precisa ser mudado quando há segurança para as partes envolvidas no serviço, administrador e usuário.

Uma política que foi projetada verificando os benefícios do usuário estará pronta para protege-lo, e com isso o sistema. A política deve ser amiga do usuário, deve cercear suas tentações, não suas atitudes, deve com certeza limita-lo no que excede a proposição do serviço, usuário não pode ser tratado como problema. Por exemplo, em uma universidade, como pesquisado neste documento, é necessário obter informações de todos *sites* contidos na Internet? É preciso receber correio eletrônico de todos os domínios? Porque o *sendmail* pode bloquear domínios indesejáveis, ou mesmo contas individuais. Até quando deve-se tolerar uma série de correios eletrônicos inconvenientes. Estas especificações por mais minuciosas que possam parecer são aspectos de segurança e podem alterar o grau de satisfação e confiança do usuário.

No aspecto recursos de hardware, em alguns casos, como em universidades, por exemplo, a disponibilidade por comprar equipamentos nem sempre é possível, para isso distribuir os recursos de maneira equivalente é necessário, por isso, foi abordado o *Quotas*, que se mostrou eficiente e completo.

A política deve tratar o comportamento dos administradores também. Para isso o serviço *Sudo* é muito importante, conforme a topologia da rede é extremamente útil, além de implementar mais um grau a segurança do sistema.

Por fim, a idéia de política de segurança em todas as partes é saudável. Esta política pode se transformar em gestão, uso, segurança física, e conforme a necessidade da empresa. Como dito anteriormente, deve prover o usuário, então deve escuta-lo se este tiver sugestões e comunicar a um conselho de informática e este retornar a possibilidade de viabilizar o proposto, ou saber explicar porque não será adotado.

Este trabalho trata superficialmente inúmeros programas e serviços. Envolver mais detalhes, quanto a *Sendmail*, ao *POP3* e *IMAP*, por exemplo, é importante para que este se torne mais completo, mesmo com relação à política, se realizar uma breve discussão vai se encontrar omissões que não foram imaginadas pelo escritor e que devem ser muito importantes.

A descrição de outros serviços também é aspecto importante para complemento de trabalho, como FTP ou hospedagem de página. Um estudo mais profundo sobre as possibilidades das ferramentas adotadas deve ser realizado aplicando a política apresentada aqui. Um estudo sobre os programas contidos no *Red Hat* 7.1 é necessário, pois é bem mais difícil de configurar que

na versão *Red Hat* 6.2, a implementação de *firewall* trouxe algumas mudanças quanto a postura permitir tudo.

Apêndice A

Accounting

Quanto deve-se confiar em um usuário?

O primeiro passo na configuração de um servidor é a proteção quanto ao esquecimento, relacionando a segurança, a que os usuários estão sujeitos. A criação de boas senhas, a restrição de acesso a diretórios e arquivos, a limitação aos *shells* que um usuário pode usar são algumas das ocupações do gerenciamento de contas, ou *account management*. O sistema operacional Linux provê várias maneiras de realizar este gerenciamento, das quais serão apresentadas as mais importantes para o escopo deste documento.

Para realizar o gerenciamento pode-se estabelecer algumas diretivas como [2]:

- Escolha de boas senhas.
- Assegurar que todas as contas de usuários possui senha ou estão bloqueadas.
- Usar senhas que ficam ativas por determinado tempo e arquivos *shadow*, que mascaram a senha.[veja Seção 5.1.3]
- Manter permissões de arquivos restritas.
- Usar módulos de autenticação, como o PAM.

Proteger o sistema de usuários curiosos, ou mesmo irresponsáveis, é necessário. Por exemplo, se um usuário curioso resolve copiar um arquivo importante para a arquitetura do sistema, como o */etc/sudoers*, que, se configurado, mostra quem são os sub-administradores e suas funções [Apêndice

E], este pode ser usado por uma pessoa que pretenda invadir o sistema. Ou ainda um usuário que escreve sua senha no monitor de seu micro.

Para evitar que os próprios usuários sejam um problema em potencial é necessário um trabalho de conscientização para explicar, por exemplo o que é uma boa senha para o usuário e o que é uma boa senha para o sistema. Porém, em minha experiência, os usuários não estão necessariamente preocupados em aprender a se proteger e a proteger o sistema, em sua maioria a preocupação está em usar se está funcionando e reclamar se não está funcionando. Por isso a importância de uma política de responsabilidades, que é a política de segurança. O valor do documento escrito é extremamente importante, pois força o usuário a ler, e talvez questionar, antes de ler, e por isso tomando consciência de termos, necessidades e qual perfil este usuário deve estabelecer diante o servidor de seu departamento. Uma política que só os administradores conhecem não tem qualquer propósito de existir, uma vez que o pessoal que realizou a elaboração desta, a pessoa que mais opinou, provavelmente, foi o próprio.

Escolher uma boa senha não é tarefa fácil [veja Seção 5.1], pois conciliar uma organização de caracteres, que é a senha, eficiente com uma possível facilidade de lembrar deve ser considerada neste momento. Senhas como nomes pessoais, cidade, cachorro, planta são péssimas senhas, pois qualquer pessoa mal intencionada utilizará um programa para quebrar senha com um dicionário de palavras, onde com certeza terá esta senha, ou uma pequena variação desta. A melhor atitude a ser tomada consiste em forçar o usuário a escolher senhas que não possuam possibilidade de estarem contidas em um dicionário, fazendo uso, então, de palavras que não existem, e ainda misturando caracteres numéricos, no Linux é possível a adoção deste procedimento, usando a interface do programa "*linuxconf*", como mostrado na figura

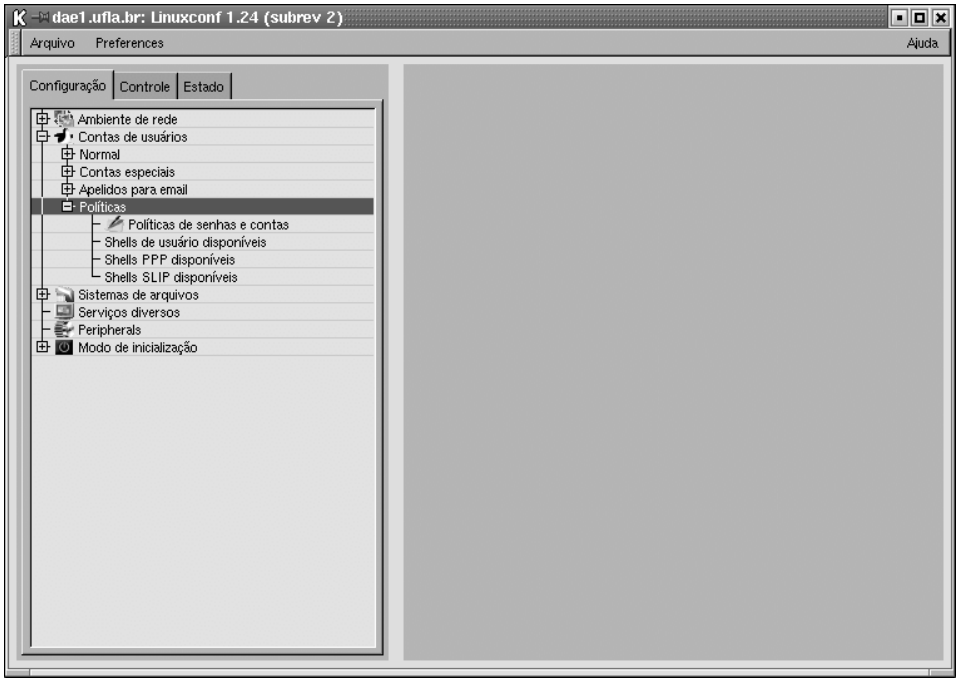

Figura A1 – Linuxconf na seção Accounting.

Pode-se determinar que só serão aceitas senhas com um determinado número mínimo de caracteres, ou ainda pode-se determinar que está senha contenha pelo menos um número específico de caracteres numéricos.

A senha ideal, com no mínimo 10 caracteres, sendo alternados de maneira aleatória números e letras, é a melhor opção para a segurança do sistema, porém deixa o usuário sujeito a esquecer esta senha mirabolante. Para isso deve ser estimulado que o usuário escolha uma frase como senha, talvez uma frase sem muitas pretensões pode ser uma ótima possibilidade de senha. Primeiro, se está frase não for algo que o usuário fale com freqüência, está estará combinando vários caracteres, geralmente acima de dez, com algo mais familiar

a quem escolheu, e fácil de lembrar, não como algo que foi gerado por um programa de senhas automáticas, que fica do tipo:

T2fjotbf00
i%8TQ#1rl1
p9D34Wv.z;

eliminando também a necessidade que alguns usuários possuem de escrever a senha no monitor, para que todos a conheçam. Para este tipo de usuário é melhor uma conta sem senha, assim não esquece jamais a senha e todos que desejarem podem tomar possuem e usar indiscriminadamente, sem fazer que para todos os efeitos é o próprio usuário e não quem descobriu a senha, cabendo cerceamentos ao dono da conta.

O administrador do sistema deve implementar a senha do tipo *shadow*. Antigamente o Linux armazenava as senhas em um arquivo chamado *passwd*, localizado em */etc*, este arquivo poderia ser algo de sucessivas tentativas de leitura até que fossem descobertas todas as senhas de todos os usuários. Depois disto deveria ser trocadas todas as senhas, por medida de segurança, e a pessoa mal intencionada iria poder ter estas senhas quantas vezes desejasse. Os atuais pacotes do *Red Hat* (6.2 até 7.1), por exemplo, já possuem as senhas em forma de *shadow*, como uma sombra, o usuário até pode ler este arquivo, mas não obterá senha alguma. E nem por este motivo este arquivo pode ser esquecido, pois contem os *login's* do sistema, ou seja, o nome de todas as contas, de todos os usuários, uma brecha de segurança.

Para a proteção quanto a permissões de leitura, escrita e execução não podem ser deixadas de lado, deve ser permitido ao usuário ter acesso ao que este

pode fazer, e reservado quanto ao que este não pode [veja 5.2]. Os atributos de arquivos devem ser observados, principalmente, se criados pelo administrados, pois podem possuir atributo "s", que permite a um usuário executar com permissão de *root*.

Em *accounting manegement* deve ser observado e configurado o arquivo */etc/login.defs*:

```
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#
#QMAIL_DIR  Maildir
MAIL_DIR      /var/spool/mail
#MAIL_FILE   .mail

# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be
used.
#       PASS_MIN_DAYS   Minimum number of days allowed between
password changes.
#       PASS_MIN_LEN    Minimum acceptable password length.
#       PASS_WARN_AGE   Number of days warning given before a password
expires.
#
PASS_MAX_DAYS   99999
PASS_MIN_DAYS   0
PASS_MIN_LEN    5
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN                  500
UID_MAX                60000
```

```
#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                  500
GID_MAX                60000

#
# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#
#USERDEL_CMD      /usr/sbin/userdel_local

#
# If useradd should create home directories for users by default
# On RH systems, we do. This option is ORed with the -m flag on
# useradd command line.
#
CREATE_HOME      yes
```

que configura características consideradas pelo sistema ao ser criada uma conta, ou usar a interface "*linuxconf*", como mostrado na figura A2.

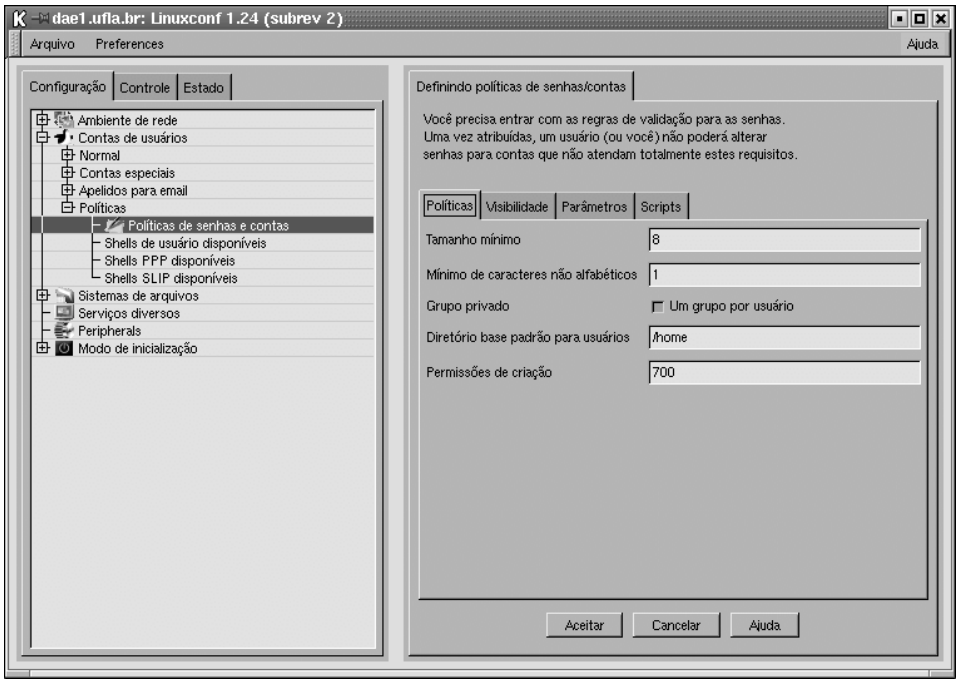

Figura A2 – Linuxconf sobre Accounting. Configuração de contas de usuários.

Quando é necessário criar uma conta temporária, como para um curso,por exemplo, não é interessante que este convidado tenha acesso a características e configurações do sistema, para isso pode-se tentar bloquear tudo, de todas as formas que se conhece, ou utilizar um *shell* que não permite que o convidado execute o comando cd, ou criar arquivos, ou executar comandos não encontrados em seu diretório padrão, para isso é necessário compilar um novo *shell*, chamado *rksh* [2], usando:

```
gcc -o rksh rksh.c
```

e copiar para */bin/rksh*, colocando como proprietário o *root* e permissão 755, com certeza para uma conta temporária é uma estrutura de permissão bastante útil.

A configuração de um controle de quota [veja Apêndice B] padrão dever ser definido neste momento, também. Definir uma quota padrão pode ser útil a uma falha do administrador ao criar uma conta, caso esqueça de atribuir um grupo de trabalho a nova conta esta irá ficar sem limite de espaço, ocasionando risco para o bom funcionamento e manutenção do sistema. A quota padrão de limite de espaço em disco seria aplicada até que o equívoco do administrador possa ser corrigido.

O gerenciamento não deve ser aplicado somente aos usuários, e também ao administrador (*root*), principalmente quando existe mais de um administrador [veja Seção 3.2 e Apêndice E].

Apêndice B

Quotas

O programa *Quotas* foi desenvolvido para permitir que o administrador do sistema limite espaço ocupado por um grupo de usuários ou apenas um usuário, este espaço pode ser verificando, e limitando-se, o número de arquivos ou o espaço em disco que pode ser utilizado. A maneira mais simples, e eficiente, de se configurar o *Quotas*, por experiência que obtive no departamento, é através o *linuxconf*.

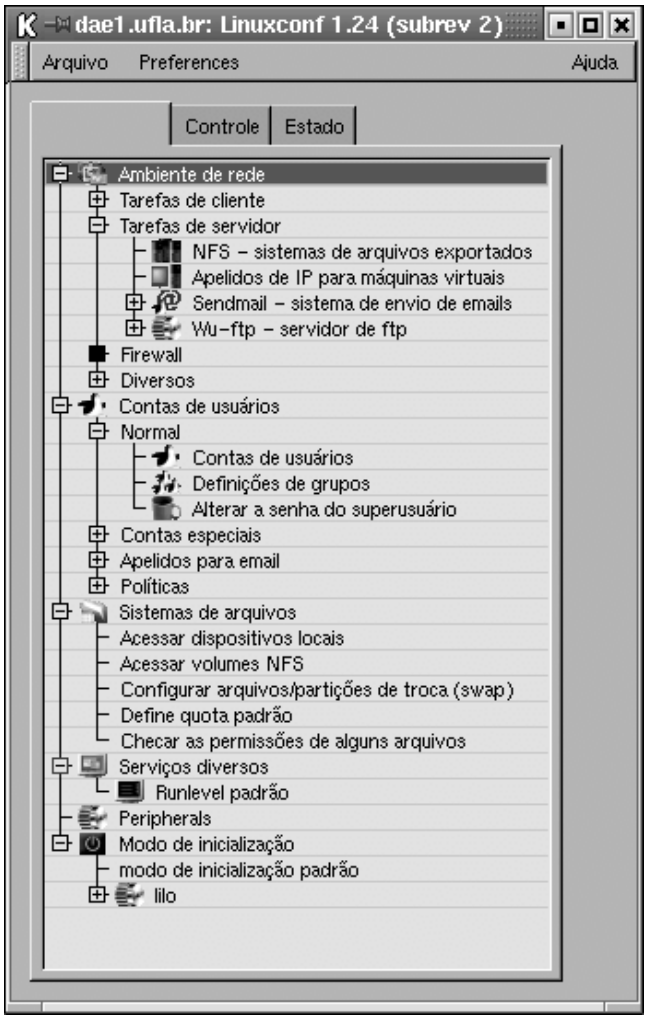

Figura B1 – Linuxconf.

A primeira etapa para implantação do *Quotas* é observar se o *Kernel* do sistema operacional o suporta, problema que não será apresentado caso esteja-se usando um *kernel* a partir da versão 2.0, se não estiver nessa versão pode-se ainda recompilar o *kernel* habilitando esta opção, mas o recomendável é realizar a atualização do *kernel*, principalmente se considerar o aspecto segurança. O próximo passo é habilitar o *Quotas* nas partições que seu sistema possui, e as

quais deve-se realizar um controle de espaço, deve-se executar o *linuxconf* e localizar o indicativo Sistema de Arquivos (*File System*) veja Figura B2, onde será apresentado todas as partições presentes no sistema, neste local é preciso localizar a partição onde vai habilitar o controle de quota e clicar sobre ela, veja Figura B3 e B4, o que fará surgir uma janela e no final da primeira aba a indicação sobre quota *user* e quota *group*, ver Figura B5.

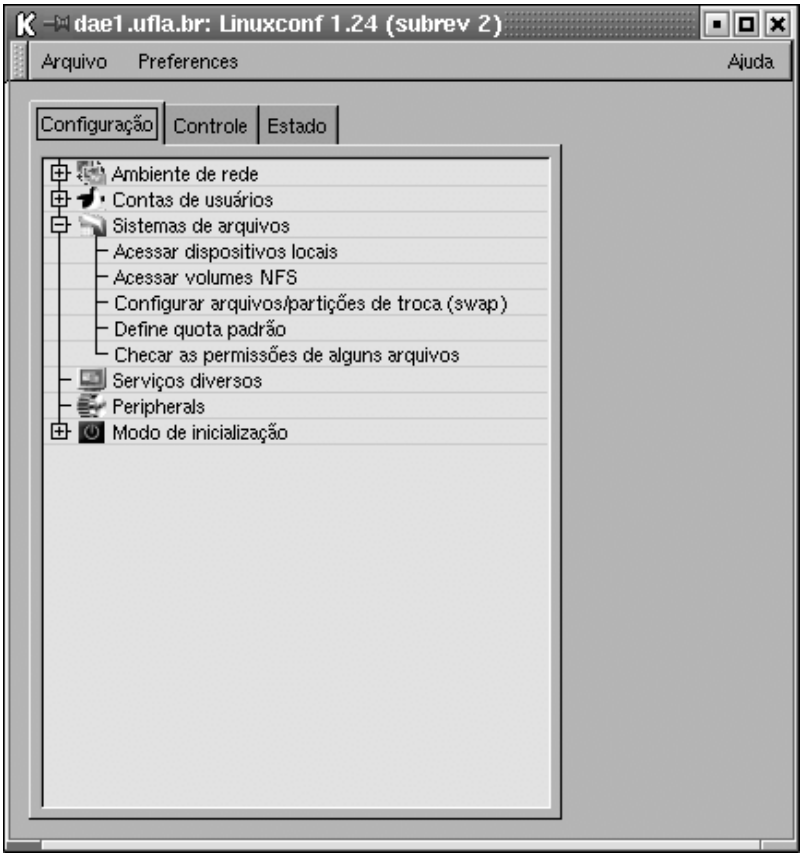

Figura B2 – Linuxconf no Sistema de arquivos.

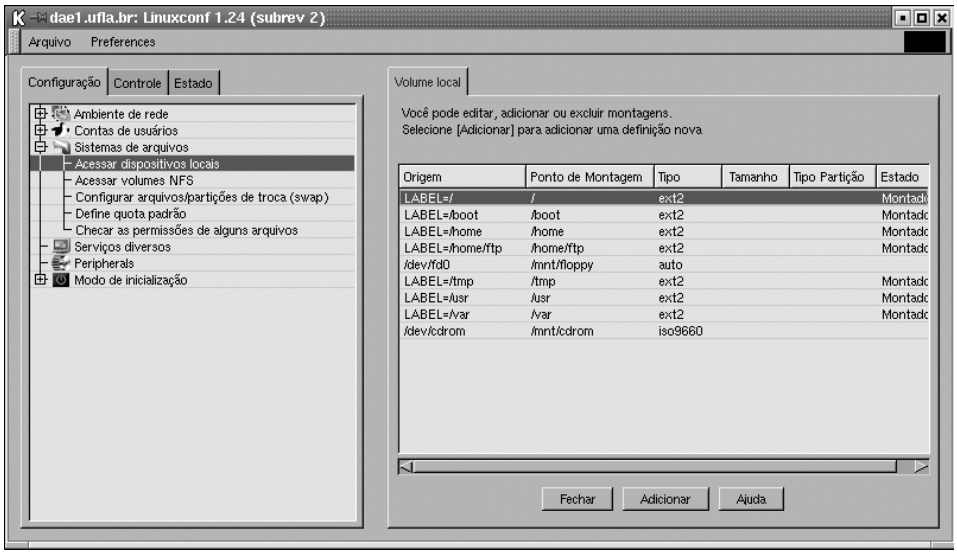

Figura B3 – Linuxconf no Sistema de arquivos. Acessar dispositivos locais.

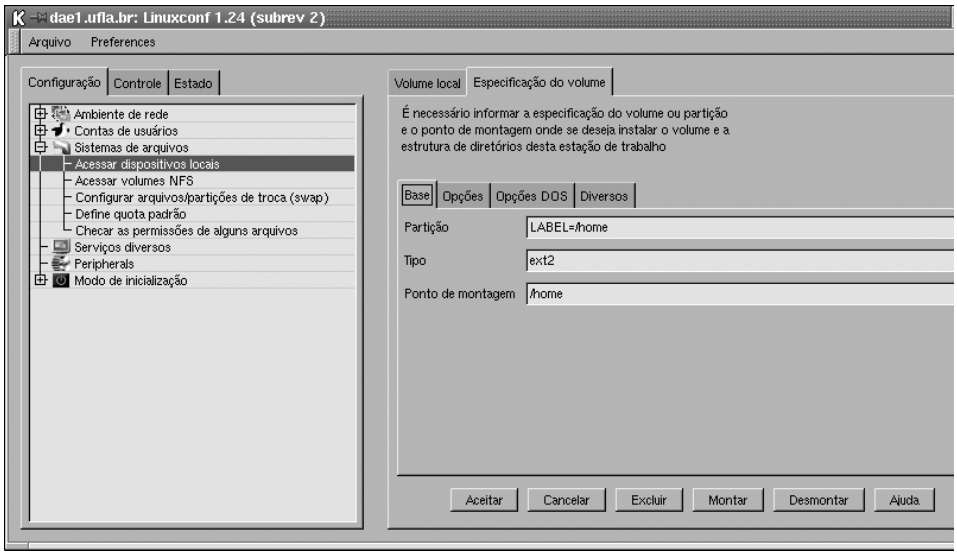

Figura B4 - *Linuxconf* no Sistema de arquivos. Dispositivo */home.*

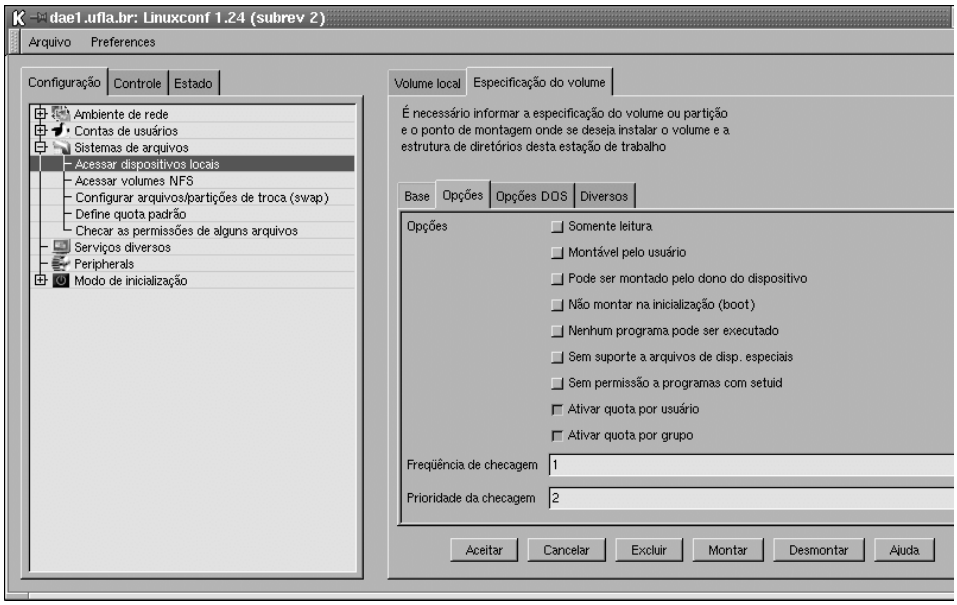

Figura B5 – Linuxconf no Sistema de arquivos. Dispositivo */home* e quotas.

O q*uota group* é o controle de quota (ou espaço) para todo um grupo de trabalho, e o *quota user* é o controle dos indivíduos do grupo, normalmente as duas opções são habilitadas. Quando o administrador determina a quota para o usuário neste local, todos os usuários estarão submetidos a aquelas restrições. O que não impede que o administrador realize uma configuração personalizada para um usuário em especial, caso este usuário esteja trabalhando em algo importante, que necessite mais espaço, por exemplo.

Agora é necessário remontar as partições com suporte para o *Quotas*. Se, o administrador, fizer a opção por usar o *linuxconf* este logo perguntará se deseja remontar a(s) partição(ões), caso não pergunte é necessário remonta-lá(s) antes de qualquer outra alteração e configuração no *Quotas*. Caso ocorra algum erro neste momento é necessário observar se as partições são do tipo ext2, pois outro tipo de partição não suporta o gerenciamento do *Quotas*. Agora é só ir ao

indicativo contas é verificar a presença de mais uma aba em seus atributos, tanto diretivas de grupos, quanto na de usuários, ver Figura B6. Pode-se então aplicar os limites para quota, é importante salientar que o número é dado em bits, ver Figura B7.

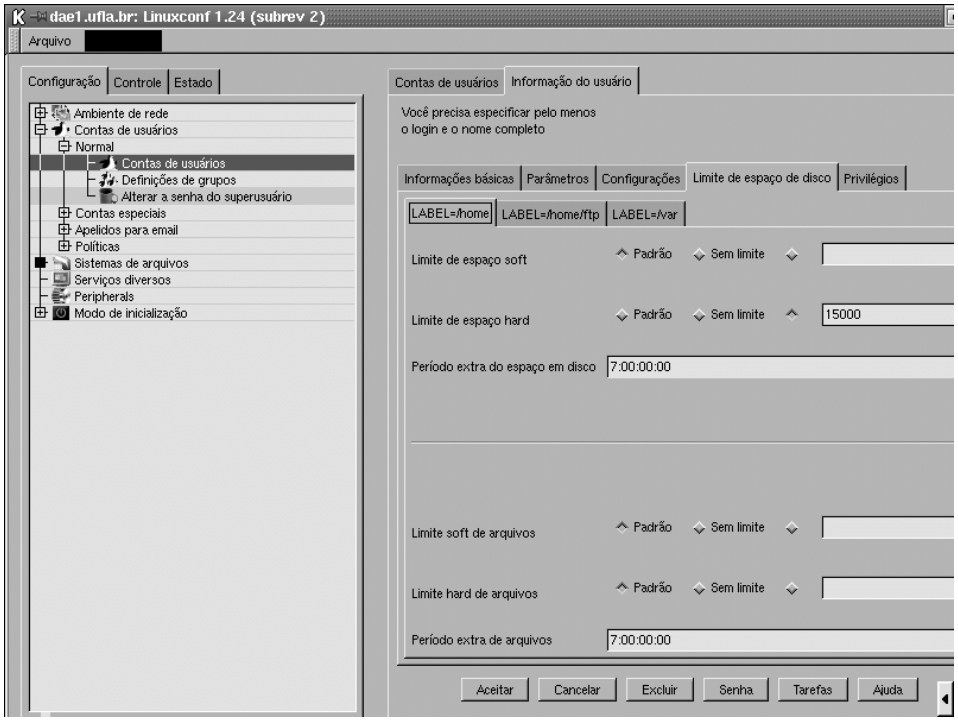

Figura B6 – *Linuxconf*. Contas de usuário.

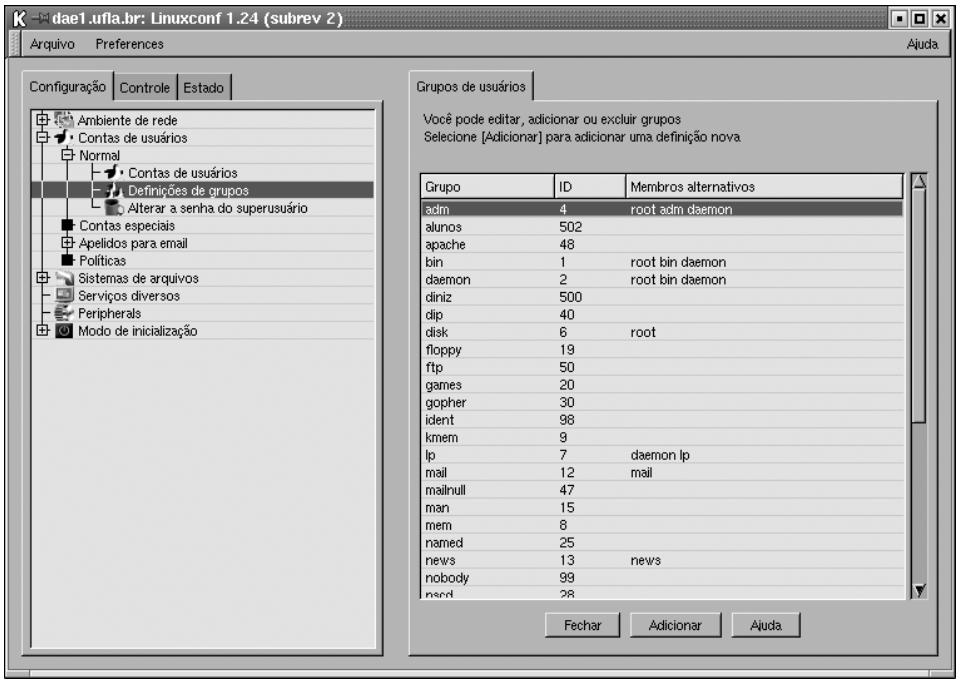

Figura B7 - *Linuxconf* Contas de usuário. Definindo de grupo.

O administrador do servidor pode então direcionar o programa para garantir que nenhum usuário vai usar mais que o espaço permitido, sem precisar estar verificando constantemente e pedindo para que esse usuário apague arquivos, simplesmente, se este vier a completar sua quota não poderá mais gravar nada em seu espaço, ver Figura B8. É nesse momento que é realizado o planejamento de serviço de e-mail e de qualquer outro serviço, como *homepage*, por exemplo, que reserva espaço em disco.

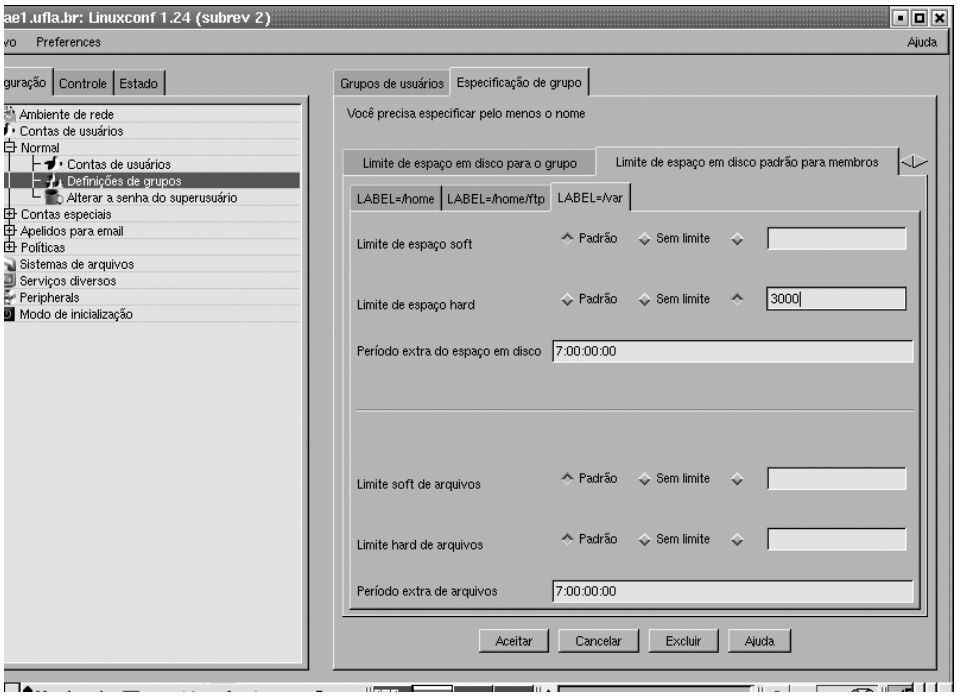

Figura B8 - *Linuxconf* Contas de usuário. Definições de grupo com quotas.

No caso do e-mail, a partição que deve permitir o *Quotas* é a */var*, caso o administrador não tenha modificado o *sendmail* [veja Apêndice C] e direcionado para outro local. O espaço de quota deve ser condizente com o tamanho máximo de uma mensagem que o usuário pode receber, pois no /var estará sendo estabelecido o tamanho da caixa postal que um usuário, ou um grupo poderá usar na partição */var*.

Se for observado o outro exemplo, a *homepage*, ou simplesmente uma área para ftp, ou ainda um diretório virtual, é necessário determinar espaço para este local também, que deve ser habilitado no */home*, caso o diretório de contas do usuário seja o */home*.

Um aspecto interessante quanto ao *Quotas*, é que independente de onde estejam os arquivos, se juntos em um mesmo computador, ou em uma mesma pasta, este programa verifica a que usuário pertencem os arquivos, e não sua localização. Por exemplo, se um usuário desejar enganar o sistema de armazenamento de dados espalhando seus arquivos por várias contas, este não irá conseguir, pois o *Quotas* realiza a soma do espaço em disco ocupado por estes arquivos, independentes de onde estejam até o permitido para aquele usuário ou grupo, então o que é realmente gerenciado pelo Quotas é o proprietário do arquivo.

A configuração do *Quotas* para limitar a quantidade de arquivos deve ser muito bem dimensionada, pois caso o sistema armazene, por exemplo, arquivos de um aluno de Ciência da Computação, pode acontecer de atingir a quantidade máxima de arquivos permitidos, caso este aluno esteja fazendo um programa, sem no entanto atingir seu espaço de armazenamento.

Apêndice C

Sendmail

O programa *sendmail* foi desenvolvido na Universidade de Berkeley, por Eric Allman, este programa foi projetado para transporte de mensagens em sistemas Unix, e logo no Linux [25].

O *sendmail* possui um conjunto de arquivos de configuração, aos quais devem ser adaptados (configurados) conforme a aplicação e dimensão do serviço ao qual será empregado. Porém a configuração destes arquivos não é nada trivial, existem livros com mais de 700 páginas só especificando sobre *sendmail*, por isso, no escopo deste trabalho só será citado as configurações para o funcionamento dos serviços aqui descritos, informações que permitirão que o servidor funcione com segurança até um aprofundamento em configurações avançadas para o programa.

O que é uma mensagem? Como funciona? Bem uma mensagem, ou correio eletrônico, ou e-mail, pode-se imaginar sendo as velhas cartas, ou seja, um papel com informações envolto por um envelope. As informações são a mensagem contida na carta, no nosso caso, no correio eletrônico, e o envelope é o mecanismo que fará a carta chegar, e também o correio eletrônico. O envelope do correio eletrônico possui um cabeçalho como qualquer informação que trafega na internet, que provê condições para que esta mensagem possua uma origem e um destino. Neste cabeçalho encontramos, primeiramente [1]:

FROM:, que contém o endereço de correio do remetente e possivelmente o seu nome completo, porém os mais diversos formatos podem ser utilizados.

TO:, onde contém o endereço do destinatário da mensagem.

SUBJECT:, é o assunto da mensagem, caracterizando o seu conteúdo por poucas palavras.

DATE:, a data em que a mensagem foi enviada.

REPLY-TO:, que especifica o endereço em que o remetente deseja receber a resposta da mensagem. Isso pode ser útil caso se utilizem diversos endereços distintos de correio, mas se deseja receber as respostas somente naquele usado mais freqüentemente, este campo é opcional.

ORGANIZATION:, leva a organização a qual a máquina pertence e na qual a mensagem foi originada, este campo é opcional.

MESSAGE-ID:, é uma expressão gerada pelo programa de transporte no sistema de origem e é único para cada mensagem.

RECEIVED:, é muito importante, pois toda máquina que receber e processar esta mensagem, incluindo as máquinas remetente e destinatário, inserem este campo no cabeçalho, fornecendo o nome do *site*, a identificação da mensagem, a hora e a data que a mensagem foi recebida, o *site* de origem e o software de transporte utilizado, sendo possível assim conhecer o caminho que a mensagem utilizou e encontrar o responsável caso algum problema tenha ocorrido.

X-ANYTHING, que é um campo que nenhum programa de mensagens pode rejeitar qualquer mensagem que tenha um cabeçalho que comece com X-. Ele é

usado para implementar funcionalidades adicionais que não estão definidas por uma RFC (*Request For Comment*), é usado, por exemplo, pela lista de discussão de ativistas Linux, onde o canal é selecionado através do campo de cabeçalho *X-Mn-Key*.

A única exceção a estrutura do cabeçalho apresentado é só a palavra chave *FROM*, que pode seguida por dois pontos, como apresentado, ou por um espaço em branco, este artifício é devido alguns programas de mensagens mais antigos.

Para uma mensagem ser enviada é necessário um programa de correio eletrônico, como o *pine*, por exemplo. Estes programas são chamados "agentes de mensagens de usuários", ou MUA (*Mail User Agents*), estes agentes recebem algumas das informações do cabeçalho manualmente, e outras retira automaticamente do sistema. O programa MUA para encaminhar a mensagem chama o agente de transporte de mensagens, ou MTA (*Mail Transport Agent*).

Em entregas locais de mensagens, o MTA somente anexa a mensagem que chega à caixa postal do destinatário. Para entregas remotas, o MTS depende da conexão, caso as mensagens sejam entregues através de uma rede que usa TCP/IP, o protocolo mais utilizado é o SMTP (*Protocolo Simples de Transferência de Mensagens* - Simple Mail Transfer Protocol), que é definido pelas [9] e [10]. O SMTP geralmente se conecta diretamente com a máquina remota (ou do destinatário) negociando a transferência da mensagem com o servidor SMTP desta máquina. Os detalhes de funcionamento do SMTP não é escopo deste documento. Porém a aquisição e execução de processos, decorrentes do envio e recebimento de e-mail é princípio fundamental na configuração do *sendmail*. É necessário impedir que um usuário não autorizado

possa executar processos em um servidor de e-mail, pois este pode vir a se danificar com este procedimento. Por exemplo, caso um usuário mal intencionado mande um e-mail para um servidor com o exclusivo intuito de danificar esta máquina, se não houver proteção ele obterá sucesso. Para isso é necessário configurar um dos arquivos do *sendmail*, especificamente o */etc/sendmail.mc*, inserindo a linha:

FEATURE('smrsh','/usr/sbin/smrsh')dnl

e gerar o */etc/sendmail.cf*, com o comando:

m4 /etc/sendmail.mc > /etc/sendmail.cf

Outro cuidado que se deve ter com o *sendmail* é a limitação do tamanho da mensagem que o servidor pode processar. Esta análise deve ser feita conforme o hardware disponível e que serviços este hardware disponibiliza a rede. Caso não seja configurado o *sendmail* não limita o tamanho da mensagem que ele pode tratar. Caso ocorra o envio de uma mensagem muito grande pode acontecer algum problema com a máquina e esta deixar de funcionar, neste caso o arquivo que dever ser configurado é o */etc/sendmail.cf*, retirando-se o símbolo # da linha

 # maximum message size
O MaxMessageSize=1000000

e colocando o valor correspondente ao planejado pelos administradores do sistema, observando além de CPU e memória, também o tamanho do disco para

ser implementado o serviço de quota em espaço de disco por usuário e grupo de usuários.

Em minhas pesquisas não conseguir determinar como deve ser alterado o */etc/sendmail.mc* para predefinir um tamanho máximo (o *SIZE*) do e-mail, por isso, depois que for gerado o *sendmail.cf*, este deve ser alterado novamente conforme o valor estipulado pelo administrador do sistema.

Ainda no arquivo */etc/sendmail.cf* deve ser observado o tempo de vida (ou *time to live*) para uma conexão e a reposta da máquina, na seção

#timeouts (many of these)

deve-se procurar a frase, que também está comentado

O Timeout.connect=5m

de onde deve ser removido o comentário, e estipulado em tempo adequado, pois se o serviço de e-mail puder ser acessado de fora da organização, é necessário verificar as possibilidades de conexão e a distância que será requerida esta informação, aumentando o tempo, que é dado em minutos, conforme a necessidade. Uma maneira de se determinar este tempo é observar as condições das centrais telefônicas da cidade (se analógicas ou digitais) e calcular a taxa de conexão média obtida por um usuário.

Estas configurações devem ser extremamente bem planejadas, e jamais esquecidas, pois podem, e serão, alvo de possíveis invasões ao servidor e ausência não planejada de um, ou vários, serviços. Outro fato a que deve-se

atentar, ao utilizar o *sendmail*, é uma correta configuração do servidor de DNS, pois o *sendmail* faz muito uso deste serviço. Um servidor de DNS mal configurado possivelmente acarretará perdas ou inconsistência no envio e recebimento de correios eletrônicos.

Uma característica do *sendmail* é a facilidade de configuração *anti-spam*. *Spam* são correios eletrônicos que são enviados para várias pessoas sem sua autorização, ou mesmo sem a pessoa ter o mínimo interesse nas informações. O *sendmail*, principalmente através do "*linuxconf*", aceita e trata bem estes correios, como visto na figura C1:

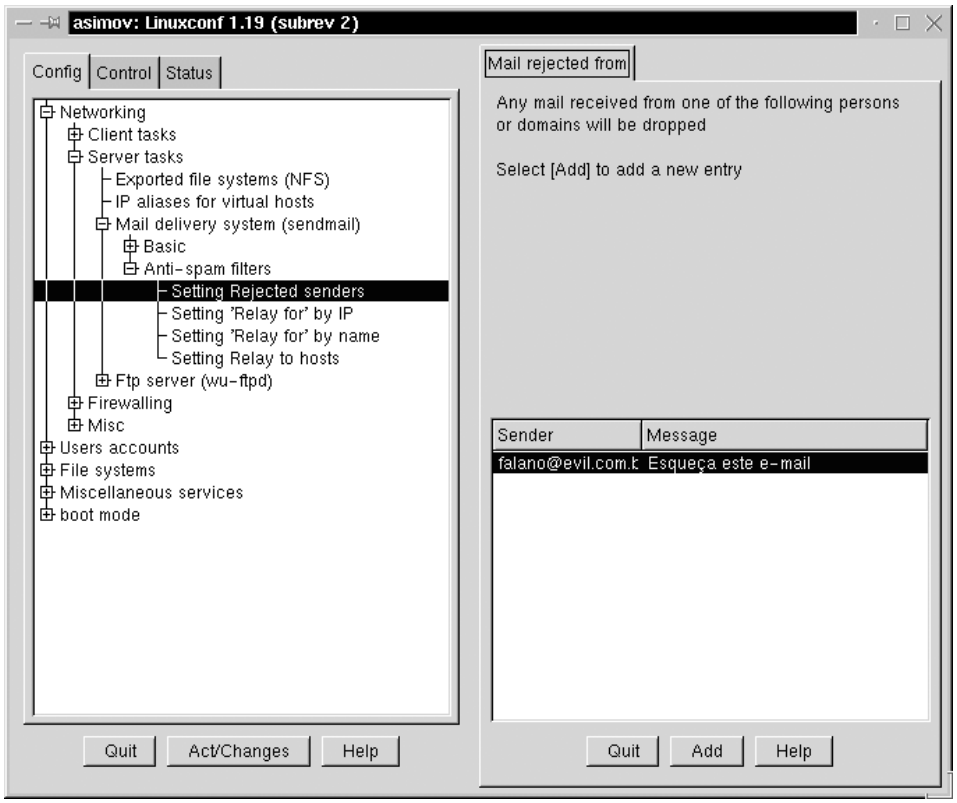

Figura C1 - Linuxconf Sendmail. Anti-spam.

A utilização do TCP *wrapper* pode ser uma solução na obtenção de segurança quando do uso do *sendmail*, pois TCP *wrapper* verifica e separa todas as entradas das conexões SMTP. Sem dúvida uma segurança a mais. Caso opte-se por não utilizar o *sendmail*, existem muitos outros programas para o serviço de correio eletrônico, a maioria mais fácil de configurar que o *sendmail*, que é de se considerar caso queira utilizar somente em uma intranet, por exemplo, contudo nenhum é tão robusto e completo quanto o *sendmail* [2].

Apêndice D

PAM

Pode-se utilizar a implementação de segurança sobre um servidor Linux utilizando PAM (*Pluggable Authentication Modules*), que integrados aos procedimentos de *login*, realizam autenticação de senha. Sendo uma ferramenta com várias opções para gerenciamento de autenticação, gerenciamento de contas, gerenciamento de seção e gerenciamento de senha.

O PAM não protege o sistema contra invasões, sua função é prevenir invasões. Ele provê um mecanismo centralizado para autenticação de todos os serviços. Como *login*, *login* remoto, ftp, *SU* e outros. Com o PAM pode ser implementado, além das citadas anteriormente, limite de acesso de usuários por determinando tempo, limite de acesso a um aplicativo e muito mais.

O diretório para que o PAM seja configurado é o */etc/pam.d*, ou existe a possibilidade de configurar o PAM através do arquivo */etc/pam.conf*, porém no momento da instalação devesse escolher qual método irá usar. O método mais comum é a instalação do */etc/pam.d*, porque pode-se configurar os serviços separadamente, pois esta instalação gera um arquivo para cada serviço suportado pelo PAM, sendo os dois métodos incompatíveis e determinados em tempo de instalação.

Os arquivos de configuração possuem duas partes, o *module-type*, que especifica que tipo de módulo possui o PAM. Podendo ser dos tipos: *auth*, *account*, *session* e *password*. A outra parte consiste no *control-flag*, que

especifica a ação que o módulo PAM irá realizar. Podendo ser dos tipos: *required*, *requisite*, *optional* e *sufficient* [2].

O *Module Type*:

- *auth*: instrue a aplicação do usuário a identificar qual a senha. Pode ser para habilitar a conexão e pode ser também para garantir privilégios.

- *account*: checa os vários aspectos da conta do usuário, tal como senhas desatualizadas, limite de acesso por tempo determinado ou para locais específicos. Pode ser usado para acesso remoto ou acesso local.

- *session*: este tipo é usado para prover funções antes e depois da seção iniciada, esta inclue um conjunto de caminhos, conexão e outros.

- *password*: é normalmente usado com o módulo *auth*, É responsável pela atualização da autenticação do usuário, geralmente uma senha.

O *Control Flag*:

- *required*: este módulo garante o sucesso do serviço. Se este é o primeiro de uma série de módulos, todos outros ainda serão executados.

- *requisite*: como anteriormente, com exceção que ocorra falha na execução deste módulo todos os outros módulos imediatamente apresentarão falha para uma aplicação.

- *optional*: como o nome indica, este módulo não é necessário. Este módulo também pode retornar falha.

- *sufficient*: se este módulo indica sucesso, todos os outros módulos são ignorados e é retornado sucesso para a aplicação. Se existir outros módulos depois deste não s/ao levados em consideração. Se este módulo falhar não necessariamente causará falha no restante, a menos que seja o único módulo.

O software PAM possui importantes módulos, baseado no escopo deste trabalho será descrito alguns, mas informações podem ser encontradas em [30] e [31].

Alguns arquivos não podem ter sua configuração esquecida no momento da montagem de um servidor, como por exemplo *reboot* e *halt*, armazenados em */etc/pam.d*. Sem a configuração destes arquivos os usuários poderão reiniciar o servidor ou mesmo desliga-lo, atitude que pode muito bem ser tomada mesmo sem a pretensão de prejudicar, simplesmente por falta de atenção. Por exemplo, usuários Linux/Unix, que estão acostumados em usar interface em linha de comando, podem por engano, quando estiver logado, via SSH no servidor, digitar *halt* ao invés de *logout* ou *exit*. Se não for aplicada esta proteção o administrador do sistema terá muitos problemas explicando porque o servidor está sempre reiniciando ou mesmo desligado.

Nos arquivos *reboot* e *halt* é importante retirar a indicação de comentário (#) da linha:

auth required /lib/security/pam_stack.so service=system-auth

pois assim antes de reiniciar ou desligar o servidor irá pedir a senha do usuário. Trocar o *control-flag sufficient* da primeira linha por *required* é muito importante, como a seguir:

auth sufficient /lib/security/pam_rootok.so
por
auth required /lib/security/pam_rootok.so

Porque assim o usuário fica impedido de reiniciar ou desligar a máquina. O sistema pede a senha por três vezes e devolve o controle de *shell* ao usuário, sem contudo que este realize a operação requerida. Para que o sistema não peça a senha é só recolocar o comentário na linha, como citado no parágrafo anterior.

Controlar o horário em que o usuário pode logar no servidor, seja remotamente ou localmente pode ser tratado como uma opção de segurança e plausível de implementação. O PAM possui o arquivo */etc/security/time.conf*, onde pode-se realizar restrição quanto a dia, hora e comandos permitidos. Por exemplo, pode-se desejar que o usuário Mane não conecte-se ao servidor nem remotamente e nem localmente, as sextas-feiras, no período de 12:00h às 18:00h, para isso é necessário acrescentar ao arquivo a seguinte linha:

login & rsh;;mane;!Fr1200-1800*

Ou ainda, que só ocorram FTP's de segunda-feira a sexta-feira de 07:00h às 24:00h, que é implementado assim:

ftp;;*;Wd0700-2400*

Controlar, ou limitar, qual capacidade de *hardware* um usuário pode exigir do sistema é importante para usuários que queiram rodar programas no servidor, pois sabe que é uma boa máquina, que possui bastante memória. Para que este usuário não faça isso o PAM possui o *pam_limits*, que é configurado no arquivo */etc/security/limits.conf*. Onde é possível controlar o usuário quanto ao tamanho máximo de um arquivo, a quantidade de memória disponível para ele, o maior número de arquivos abertos, o maior tamanho de memória que para armazenar e resgatar dados (*stack*), o tempo de CPU, o maior número de processos, o maior número de *login's*, entre outros.

Por exemplo, pode-se determinar que em todos os grupos e que para todos os usuários, podem abrir somente quatro conexões, assim:

```
*               hard    maxlogins       4
```

Ou que o grupo teste pode utilizar, aproximadamente, 64MB de memória, assim:

```
@teste hard     memlock         64000
```

e assim por diante, pois estas configurações podem ser aplicadas a usuários, grupos, em separado ou conjuntamente.

Outra ferramenta do PAM é o *pam_access*, este módulo controla o acesso a consoles, a domínios, a grupos. O arquivo de configuração é o */etc/security/access.conf*. Com este módulo pode-se, por exemplo, caso o

servidor possua problemas com segurança física, deve-se aplicar a restrição de desabilitar todos os consoles, exceto o *tty2* para o *root*, da seguinte forma:

-:ALL EXCEPT root: tty2

Pode-se também restringir o acesso de usuários a determinados servidores. Se fosse necessário que os membros do grupo teste não acessassem o servidor laranja, e só acessassem o servidor abacaxi, então procederia assim:

+: teste: abacaxi

Outras configurações para este módulo também são possíveis, assim como existem muitos outros módulos e detalhes sobre a configuração destes apresentados. A intenção deste apêndice é de mostrar algumas funções para solucionar eventuais dúvidas durante a leitura dos textos anteriores, para mais detalhes ver os manuais do PAM e em [2] [31].

Apêndice E

SUDO (SuperUser)

Para sistemas que necessitam mais que um administrador existe um programa que se torna uma opção extremamente útil, que trata um aspecto importante de segurança, a transmissão e constante utilização da senha do *root*.

Sudo significa "*Superuser Do*". Este programa foi desenvolvido originalmente pela Universidade de *Buffalo*, no Estado de *New York*, nos EUA, em meados dos anos 80, depois o projeto foi direcionado a Universidade de *Boulder*, no Estado do *Colorado*, também nos EUA [22], este provê privilégios do administrador do sistema, como se um determinado usuário o fosse. O *Sudo* pode ser configurado para que um determinado usuário do sistema venha a ser um sub-administrador do sistema, sem que no entanto este tenha que saber qual é a senha do administrador, pode-se determinar alguns comandos que só o administrador normalmente pode executar.

O *Sudo* pode ser encontrado em versão em *RPM* e *TAR.GZ*, como na maioria dos casos a vantagem do *RPM* é a facilidade de instalação, contudo a escolha pelo *Sudo TAR.GZ* possibilita a escolha de algumas características quando da sua compilação para ser instalado, como suporte a PAM, com alerta ao correio eletrônico do administrador caso o usuário tente executar o *sudo* sem que este tenha autorização para isso, para controlar *path* que os usuários podem ter acesso, com insultos quando uma pessoa tenta fazer algo que não pode, e outros. Na versão 6.2 do *Red Hat* deve ser considerada a possibilidade de instalar o *Sudo* através do *TAR.GZ*, porque a maioria destas opções não vem como padrão, entretanto na versão 7.1 do *Red Hat* a instalação *RPM* já possui as principais opções.

Para realizar a instalação via *TAR.GZ* é necessário primeiro executar o *./configure*, mais as opções de controle do *Sudo*, que podem ser encontradas na internet [2] e [23] ou em vários *HOWTO's*, esse irá criar o *Makefile*, com as diretivas para compilação e instalação. O próximo passo é executar o *make* e depois o *make install* e está pronto.

O arquivo de configuração do *Sudo* é o */etc/sudoers*, este possui seções como:

* *Host alias specification*: nesta seção pode-se selecionar a lista de sub-administradores por número IP, *hosts*, por grupos de trabalho, e outros.

* *Command aliases specification*: nesta seção são especificados a lista de comandos, através da palavra chave *Cmnd_Alias*, que os sub-administradores poderão executar e também fornecer privilégios em diretórios.

* *User alias specification*: nesta seção estão os nomes dos grupos que terão privilégio de *root*, construídos depois da palavra chave *User_Alias*.

* *User privilege specification*: nesta seção é declarados direitos que os grupos, especificados na seção *User_Alias* tem no sistema.

Mais detalhes sobre como construir as declarações devem ser procurados em [2][22].

Diante declarações bem construídas, bem estruturadas, e uma cuidadosa escolha de quem são os sub-administradores do sistema, pode-se eliminar o uso do comando *SU*, principalmente se parte da administração é realizada

remotamente, pois assim estará fazendo o menor uso possível do uso da senha do *root*, o que é o ideal, para a segurança do sistema.

Para bloquear o uso do *SU*, basta ter o PAM instalado em seu sistema. Então ir a */etc/pam.d/su* e inserir a linha:

auth required /lib/security/pam_listfile.do onerr=fail \ item=user sense=deny file=/etc/security/sudeny

Caso o administrador queira negar completamente o *SU*, basta acrescentar a linha:

auth requisite /lib/secutity/pam_deny.so

ao invés da proposta anterior.

Apesar de todos os benefícios do uso do *Sudo*, como dito anteriormente, o cuidado com suas configurações é o mínimo necessário. Um planejamento é fundamental para que esta ferramenta de auxílio não torne-se um problema. Uma configuração desatenta pode permitir que o sub-administrador obtenha acesso indevido, ou mal configurado. Por exemplo, se um usuário, que possui permissão para executar o *sudo*, este poderá usar o editor de texto VI para copiar alguns *shells* do sistema, como */bin/bash*, */bin/ash*, */bin/ksh*, etc., para o diretório */tmp/bash*, por exemplo, iludindo o sistema e executando um *shell* em um diretório sem restrições. Para este caso vale o citado no texto [veja seção 4.2], somente deve ser permitido o que for necessário, e restrito o que não for.

Outro cuidado está em um usuário de um grupo descobrir que um outro grupo possui direitos de execução de determinado programa, e este tentar mudar de grupo para obter está permissão também, para isso arquivos de *log* são importantes e sua monitoração também. A análise de arquivos de *log* com cuidado é fundamental para inibir tal procedimento e tomar as medidas, de punição, definidas em sua política de segurança.

Glossário

CERT - Computer Emergency Response Team. Organismo criado em 1988 pela Darpa, visando tratar questões de segurança em redes, em particular na Internet.

Conexão – Ligação do seu computador a um computador remoto.

Correio eletrônico - Correio transmitido por meios eletrônicos , normalmente, redes informáticas. Uma carta eletrônica contem texto (como qualquer outra carta) e pode ter, eventualmente, anexo um ou mais arquivos.

Daemon - Programa que roda num computador e está (sempre) pronto a receber instruções/pedidos de outros programas para a execução de determinada ação.

DNS - Sigla de Domain Name Server. Designa o conjunto de regras e/ou programas que constituem um Servidor de Nomes da Internet. Um servidor de nomes faz a tradução de um nome alfanumérico (ex. microbyte.com) para um numero IP (ex. 192.190.100.57). Por exemplo, no DNS português, gerem-se todos os nomes terminados em pt. Qualquer outro nome será também traduzido pelo mesmo DNS, mas a partir de informação proveniente de outro DNS (isto se essa informação não tiver sido previamente obtida).

e-mail - Electronic Mail. Correio Eletrônico.

Finger - Programa para obter informações sobre uma determinada pessoa que tenha um endereço eletrônico na Internet. É indicado o endereço eletrônico dessa pessoa e ele procura e devolve informação relativa à mesma, apos ter inquirido o computador onde essa pessoa tem a sua caixa de correio.

Firewall - Parede de Fogo. Medida de segurança que pode ser implementada para limitar o acesso de terceiros a um determinada rede ligada à Internet. Os mecanismos de implementação são variados, percorrendo variados tipos de controlo por software ou hardware. Num caso limite, a única coisa que uma firewall poderia deixar passar de um lado (rede local) para o outro (resto da Internet) era correio eletrônico (podendo mesmo filtrar correio de/para determinado sitio).

FTP - File Transfer Protocol. Designa o principal protocolo de transferência de arquivos usado na Internet, ou então um programa que usa esse protocolo.

FTP server - Servidor de FTP. Computador que tem arquivos de software acessíveis através de programas que usem o protocolo de transferência de arquivos, FTP.

Gopher - Um espécie de parente pobrezinho do WWW. Existente há muito mais anos que este, permite a procura de informação em bases de dados existentes em todo o mundo, utilizando-se ou não algumas ferramentas próprias de pesquisa por palavras-chave.

Homepage - Pagina base do WWW de uma instituição ou particular. A pagina base é uma espécie ponto de partida para a procura de informação relativa a essa pessoa ou instituição.

Host - Computador central. Também chamado de servidor ou nó, por vezes.

How-to - Documento(s) em formato eletrônico, que acompanham o Linux (versão de domínio público do Unix) e que constituem uma espécie de manual, onde se pode procurar informação sobre quase toda a tarefa de instalação, administração e atualização do Linux.

LAN - Local Area Network. Rede Local. É uma rede com 2 ou algumas dezenas de computadores que não se estende para além dos limites físicos de um qualquer edifício. Normalmente utilizada nas empresas para interligação local dos seus computadores. Existem várias tecnologias que permitem a realização de uma rede local, sendo as mais importantes, a Ethernet e o Token-Ring.

Login - Identificação de um utilizador perante um computador. Fazer o login é o ato de dar a sua identificação de utilizador ao computador.

PGP - Pretty Good Privacy. Programa para a codificação mensagens de texto, inventado por Philip Zimmerman. Uma mensagem assim enviada é inquebrável e só o seu destinatário a pode decodificar, dando para isso uma chave que só ele conhece.

SMTP - Simple Mail Transport Protocol. Protocolo utilizado entre os programas que transferem correio eletrônico de um computador para outro.

Telnet - Protocolo/programa que permite a ligação de um computador a um outro, funcionando o primeiro como se fosse um terminal remoto do segundo. O computador que "trabalha" é o segundo enquanto que o primeiro apenas visualiza no ecran os resultados e envia os caracteres digitados no seu teclado.

UDP - User Datagram Protocol. Um dos protocolos do conjunto de protocolos da Internet (habitualmente designado por TCP/IP). Corresponde ao nível 4 do modelo OSI, pois é um protocolo de transporte, sem ligação. Em UDP, uma mensagem é enviada para o destino, sem que haja uma ligação lógica efetuada entre a origem e o destino (semelhante a uma ligação telefônica entre dois pontos). O(s) pacote(s) de mensagens podem então passar por vários nos da Internet até chegar ao destino. Menos confiável que o TCP (outro protocolo de transporte, mas com ligação), mas bastante útil quando a perda de um ou outro pacote não seja importante e se pretende velocidade na transmissão e evitar a sobrecarga de várias ligações lógicas estabelecidas.

World-Wide-Web - Conjunto dos servidores que "falam" HTTP e informação ai armazenada em formato HTML. O World-Wide-Web é uma grande teia de informação multimídia em hipertexto. O hipertexto significa que se pode escolher uma palavra destacada numa determinada página e obter assim uma outra página de informação relativa (semelhante ao Help do Windows). As paginas podem conter texto, imagens, sons, animações, etc. O World-Wide-Web é uma gigantesca base de dados distribuída acessível de uma forma muito atraente e intuitiva.

Referências Bibliográficas

[1] KIRCH, O. **Guia do Administrador de Redes**. Editora Conectiva, Curitiba, 1999. 496p.

[2] MANN, S.; MITCHELL, E.L. **Linux System Security**. Editora Pretince Hall PTR, 2000. 564p.

[3] TANENBAUM, A.S. **Redes de Computadores**, Editora Campus Ltda 1997. 923p.

[4] TANENBAUM, A.S. **Sistemas Operacionais Modernos**, Editora Prentice-Hall do Brasil Ltda, Rio de Janeiro, 1995. 493.

[5] ANCHIESCHI, O.J.G. **Segurança Total**. Editora MAKRON Books, São Paulo, 2000. 276p.

[6] ANÔNIMO. **Segurança Máxima Para Linux**. Editora Campus, 2000. 761p.

[7] SOARES, L.F.; LEMOS, G.; COLCHER, S. **Redes de Computadores**. Editora Campus, 1995. 705p.

[8] GARFINKEL, S.; SPAFFORF, G. **Practical Unix and Internet Security**. Editora O'Reilly & Associates, Inc., 1996. 971p.

[9] POSTEL, J. **Simple Mail Transfer Protocol**. RFC 0788, Novembro 1981.

[10] POSTEL, J.. **Simple Mail Transfer Protocol**. RFC 0821, Agosto 1982.

[11] CROCKER, D. **Standard for the format of ARPA Internet text messages**. RFC 0822, Agosto 1982.

[12] FRASER, B. **Site Security Handbook**. RFC 2196, Setembro 1997.

[13] ABOBA B.; ARKKO, J.; HARRINGTON, D. **Introduction to Accounting Management**. RFC 2975, Outubro 2000.

[14] McCLOGHRIE, K.; HEINANEN; J., GREENE; W.; PRASAD, A. **Accounting Information for ATM Networks**. RFC 2512, Fevereiro 1999.

[15] ELKINS, M. **MIME Security with Pretty Good Privacy (PGP)**. RFC 2015, Outubro 1996.

[16] FRYE, R.; LEVI, D.; ROUTHIER, S.; WIJNEN, B. **Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Management Framework**. RFC 2576, Março 2000.

[17] STEWART, R., et al., **Simple Control Transmission Protocol**. RFC 2960, Outubro 2000.

[18] ROSE, M.; McCLOGHRIE, K. **Structure and Identification of Management Information for TCP/IP-based Internets**, STD 16. RFC 1155, Maio 1990.

[19] GAVIN, T.; EASTLAKE 3rd, D.; HAMBRIDGE, S. **How to Advertise Responsibly Using E-Mail and Newsgroups or - how NOT to MAKE ENEMIES FAST!**. RFC 3098, Abril 2001.

[20] HOLBROOK, J.P.; REYNOLDS, J.K.. **Site Security Handbook**. RFC 1244, Julho 1991.

[21] MIT. **Kerberos: The Network Authentication Protocol**, Dezembro 2000. http://web.mit.edu/kerberos/www/. Junho de 2001.

[22] MILLER, T.C. **Sudo Main Page**, Março 2001. http://www.courtesan.com/sudo/. Maio 2001.

[23] REDHAT, Inc. **redhat.com | Mirror Sites**, 2001 http://www.redhat.com/download/mirror.html, Junho 2001.

[24] GUARDIAN DIGITAL, Inc.. **Linux Security - The Community's Center For Security**, 2000. http://www.linuxsecurity.com, Junho 2001.

[25] SENDMAIL, Inc. **Sendmail Home Page**, 2001. http://www.sendmail.org, Junho 2001.

[26] THE OPENSSL PROJECT; ENGELSCHALL, R.S. **OpenSSL: The Open Source toolkit for SSL/TLS**, 1999. http://www.openssl.org. Abril 2001.

[27] IBR, TU Braunschweig. **Network Management Research Group**, Abril

2001 por Juergen Schoenwaelder http://www.ibr.cs.tu-bs.de/projects/nmrg/. Junho 2001.

[28] CONECTIVA, Inc. **CONECTIVA S.A.**, 2001 http://conectiva.com.br, Maio 2001.

[29] SOFTWARE ENGINEERING INSTITUTE e CARNEGIE MELLON UNIVERSITY. **CERT Coordination Center**, 2001. http://www.cert.org, Junho 2001.

[30] ATOMICTANGERINE, Inc. **SecurityPortal**, 2001. http://www.securityportal.com. Junho 2001.

[31] MORGAN, A. **A Linux-PAM page**, 09/05/2001. http://www.kernel.org/pub/linux/libs/pam. Maio 2001.

[32] ARAGON, B.C. **Gerenciamento Remoto a Servidores de Redes Locais: Gerenciamento via Web X Gerenciamento via Terminal**, Em andamento (2001).