CENTROATLÂNTICO PT

RENATO MARTINI

MANUAL de SEGURANÇA em REDES LINUX

Renato Martini

Manual de Segurança em Redes Linux

Edições Centro Atlântico
Portugal/2000

Copyright (C) 2000 Renato Martini & Frédéric Couchet.
Outorga-se a permissão de copiar, distribuir e/ou modificar este documento sob os termos da Licença de documentação Livre GNU. Versão 1.1 ou qualquer outra versão posterior publicada pela Free Software Foundation; com as Secções Invariantes sendo apenas “Preâmbulo à edição brasileira” e “Preâmbulo à edição francesa”, sem Textos da Página de Rosto, e sem textos da Contra-capa.
Uma cópia da Licença é incluída na secção intitulada “GNU Free Documentation License”.

Manual de Segurança em Redes Linux
Colecção: Tecnologias
Autor: Renato Martini
Direcção gráfica: Centro Atlântico
Tradução: António Cardoso (POLI)
Capa: Paulo Buchinho

© Centro Atlântico, Lda., 2000
Av. D. Afonso Henriques, 1462 - 4450 Matosinhos
Tel. 22 - 938 56 28/9 Fax. 22 - 938 56 30
Rua da Misericórdia, 76 - 1200 Lisboa
Tel. 21 - 321 01 95 Fax 21 - 321 01 85
Portugal
geral@centroatlantico.pt
www.centroatlantico.pt

Fotolitos: Centro Atlântico
Impressão e acabamento: Inova
1ª edição: Novembro de 2000

ISBN: 972-8426-30-5
Depósito legal: 157191/00

Marcas registadas: todos os termos mencionados neste livro conhecidos como sendo marcas registadas de produtos e serviços, foram apropriadamente capitalizados. A utilização de um termo neste livro não deve ser encarada como afectando a validade de alguma marca registada de produto ou serviço.
O Editor e os Autores não se responsabilizam por possíveis danos morais ou físicos causados pelas instruções contidas no livro nem por endereços Internet que não correspondam às *Home-Pages* pretendidas.

Índice

Prefácio 5

1. Fundamentos de Segurança 11
- 1.1 Introdução 11
- 1.2 Segurança: o conceito 11
- 1.3 Segurança e as ferramentas de rede 13
- 1.4 Objectivo deste livro 18

2. 0 *Firewall*: duas soluções no ambiente Linux 19
- 2.1 Uma palavra inicial sobre os firewalls 19
- 2.2 Firewalls e acesso remoto: o Secure Shell 21
- 2.3 Firewalls: solução Linux 27
- 2.4 A filtragem de pacotes 29
- 2.5 IPCHAINS (The Enhanced IP Firewalling Chains Software for Linux) 30
- 2.6 O Firewall SINUS - um filtro de pacotes TCP/IP para o Linux 47

3. Monitorização de Rede 75
- 3.1 Os scanners de rede 75
- 3.2 Hunt: um sniffer de ligações diversas para o Linux 96
- 3.3 SAINT: Security Administrator's Integrated Network Tool 104

ANEXOS 113
- Anexo A: O ficheiro '/etc/protocols' 114
- Anexo B: O ficheiro '/etc/services' 115
- Anexo C: Um 'script' para IPCHAINS 125
- Anexo D: Os cabeçalhos de pacotes IP, ICMP, TCP, UDP e ARP 133
- Anexo E: Como é que um pacote viaja através da pilha TCP/IP? 136
- Bibliografia 137
- Anexo F: GNU Free Documentation License 138

1. Fundamentos de Segurança

1.1 Introdução

O objectivo deste documento é demonstrar a todos os utilizadores como o sistema operativo GNU/Linux oferece uma solução *completa* para o problema da segurança. Deveremos ter em atenção, que quando falamos em "solução", não significa a existência de um sistema *absolutamente* seguro. O GNU/Linux oferece, isso sim, as ferramentas necessárias para a gestão da segurança numa máquina isolada ou numa Intranet, e, então, tal Intranet ligada à Grande Rede. Este documento descreve introdutoriamente essas ferramentas e os problemas de segurança.

1.2 Segurança: o conceito

Trataremos o tema segurança como sendo a restrição dos recursos de *uma* máquina, de uma rede, ou até mesmo de porções dessa rede para outros utilizadores ou computadores. A Segurança *não é mais do que a gestão de tal restrição*, – o que constitui portanto uma *política de segurança*, ou como se diz em Inglês: *security policy*. Genericamente significa que numa rede existem determinados recursos (ficheiros, dispositivos de hardware, etc.) que se encontram disponíveis para este computador ou *tipo* de utilizador, mas que por outro lado ficam restritos a outros computadores, estejam eles dentro ou fora da rede.

Mesmo um sistema como o antigo MS-DOS implementava um *tipo* de segurança, bastante singela se comparada com os Sistemas Operativos

multi-utilizadores, como o GNU/Linux ou qualquer outro sistema UNIX. No MS-DOS existem ficheiros *ocultos*, ficheiros com atributos para que se permita que estes somente sejam lidos, etc. Por conseguinte, existe a restrição de "áreas" do SO. Claro, que tudo isso num sistema MS-DOS, ou mesmo na sua sequência histórica, os Windows (9x), possuem restrições que podem de forma fácil serem "burladas". Mas quer o MS-DOS e ou o Windows, não foram idealizados a pensar em redes. O recurso à *Networking* não é embutido (*built-in*), intrínseco ao Windows e por isso mesmo todos os embaraços que os seus utilizadores possuem quando navegam na Internet, como por exemplo, no famoso *Back Orifice*.

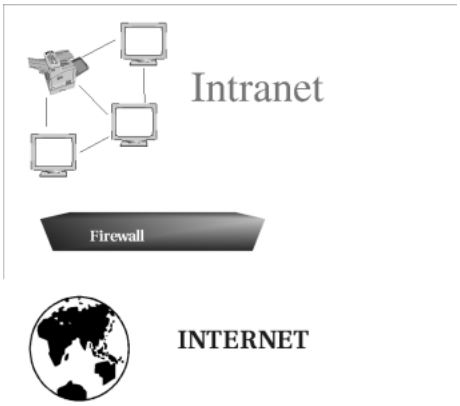
O GNU/Linux – em todas as suas distribuições – é um sistema multi-utilizador, essencialmente um SO capacitado para as Redes, não sendo por isso um recurso que lhe foi acrescido *externamente*. E como todo o SO de rede, estabelece privilégios como forma de segurança: existem utilizadores especiais, na nomenclatura correcta Super-Utilizadores ou utilizador *Root*, que podem alterar os ficheiros especiais do sistema, montar as partições, sejam elas locais ou remotas, desligar a rede, etc. O utilizador *comum*, sem privilégios, não o poderá fazer. No entanto, quando nos colocamos na óptica de um computador pensado isoladamente, isto é, uma única estação de trabalho, usado por, digamos, seis pessoas, veremos que nesta situação, já deveremos atender às questões relativas a uma implementação do tema segurança. Mas quando observamos uma rede local, a segurança é ainda um tema mais urgente. Numa Intranet, existem por vezes diversos recursos de várias máquinas que estão ou devem estar interditados de um determinado departamento, por exemplo. E, sobretudo, quando esta Intranet se liga à Internet, o que é quase inevitável, a segurança torna-se ainda mais urgente e fundamental.

Muitas empresas reagiram ao enorme problema de segurança que traz a Internet com a consequente *segmentação* das suas Intranets. Ou seja, a separação *física* da Intranet da rede mundial. Mas esta, não deve ser a "solução" adoptada, na medida em que significa a restrição de recursos de

uma rede, o que seria desta forma, rumar no caminho inverso ao do nosso tempo: a saber, radicalizar a rede mundial, como conclui Refik Molva:

"*A segmentação resultante é o maior impedimento para a realização do conceito de uma Internet global*."[2]

Poderíamos esquematizar assim:



1.3 Segurança e as ferramentas de rede

Numa única máquina ou numa rede local (de pequeno, médio ou grande porte) e, assim, quando esta rede está ligada à Grande Rede, a segurança tem que ser gerida. Assim sendo, veremos que o GNU/Linux possui todas as ferramentas e documentação necessárias para a realização desta importante tarefa.

Deveremos ter sempre presente que tais ferramentas são inúteis, se o administrador não tiver uma política de segurança. Não constitui objectivo deste documento tratar explicitamente do tema. Existem no entanto, leituras introdutórias que são obrigatórias, tais como: o *RFC*[3] *2196* intitulado de **Site Security Handbook** (*http://www.rfc-editor.org*) escrito por B. Fraser (Setembro de 1997); e o *Linux Security HOWTO* (*ftp://metalab.unc.edu/pub/Linux/docs/HOWTO*) de K. Fenzi e D. Wreski (Maio de 1998); e o livro de Paul Sery *Ferramentas Poderosas para Redes em Linux* (ed. Ciência Moderna[4]). Em

todos estes documentos, o leitor certamente poderá encontrar todos os ensinamentos iniciais para a construção da *política de segurança da sua rede*, - e que o *RFC* 2196 propriamente define como uma declaração formal de *regras* que concedem acesso aos recursos de informação e tecnologia, existindo por isso o dever de serem cumpridas.

Devemos lembrar ainda que o leitor poderá encontrar um conjunto bastante abrangente de ferramentas, para todos os "sabores" de sistemas UNIX, no *site http://www.securityfocus.com*, visita obrigatória para todos aqueles que se ocupam com a segurança. Há dois portais no idioma português sobre segurança, que no nosso entender são obrigatórios:

http://www.securenet.com.br
http://www.linuxsecurity.com.br

O administrador Renato Murilo Langona, faz a mediação de uma lista de discussão, a "security-l", no site *Linuxsecurity*, onde o leitor pode obter mais informações. No caso de ser um estudioso do tema e administrar uma rede, deve procurar esta lista.

Pesquise as ferramentas de procura disponíveis na Internet, para assim se encontrar constantemente actualizado. Como a Internet é muito *mutável,* não faz sentido citar dezenas e dezenas de *sites*, que agora estão *on-line*, mas que muito brevemente podem deixar de estar, ou mudam de nome, etc.

Outro ponto que deve ser tido em conta pelo leitor é a *criptografia*. Neste momento, não é mais um tema puramente técnico, que se encontra voltado simplesmente para a Matemática ou a Lógica. Já sabemos que a nova geração de protocolos TCP/IP que estão a ser desenvolvidos, o IPv6, vai colocar definitivamente protocolos criptográficos na *suite* TCP/IP. É necessário que o administrador que lê este manual, esteja preparado para a

tarefa da montagem de um *criptosistema*. O que é isso? É todo um sistema criptográfico que se baseia em três pontos fundamentais:

- **AUTENTICIDADE**
- **PRIVACIDADE**
- **INTEGRIDADE**

Resumidamente: um software criptográfico pode garantir que a minha mensagem ou os meus dados, são efectivamente *autênticos*, não podendo por isso serem *repudiados* pelo autor; permitindo total segurança na medida em que ninguém que eu não desejo irá fazer a leitura das minhas mensagens; e, por fim, que ninguém poderá *alterar* a minha mensagem.
No CIPSGA, pode fazer o download do GnuPG (versão 1.0.2) em pacotes RPM, já prontos para instalação e uso. Também traduzimos o Mini How To para o idioma português, onde pode ser obtido um primeiro contacto com o programa.

GnuPG Mini How To:

http://www.cipsga.org.br/gnupg.html

O GnuPG foi desenvolvido pelo alemão Werner Koch. É uma implementação *livre* do padrão OpenPGP. Como um software livre, ele pode ser usado em privado ou comercialmente, diferentemente do PGP distribuído pela NAI.

O GnuPG usa a denominada criptografia assimétrica, por isso usa duas chaves criptográficas: uma para criptografar ou cifrar – chave pública –, e outra – chave privada ou secreta – para descriptografar ou decifrar. A primeira é divulgada publicamente, a outra evidentemente guardada com todo o zelo possível. Assim se quero criptografar uma mensagem para o *José*, uso a chave pública deste utilizador para criptografar, e *somente ele*, que possui a chave privada correspondente poderá ler a mensagem decifrando-a. Da mesma forma, posso assinar digitalmente uma mensagem assegurando a sua autenticidade e integridade. Assim sendo, a criptografia assimétrica é perfeita para a Era da Internet, para a época dos canais de comunicação abertos. No entanto, a criptografia clássica ou simétrica trabalha apenas com uma única chave, ela cifra e decifra, – se a envio num canal de comunicação e alguém obtém a sua posse, pode ler e assinar qualquer documento usando o nome de outrem. Se administra uma rede, use o GnuPG. Ensine as pessoas a fazê-lo; este tipo de prática vai com toda a certeza intensificar-se com o tempo, e o "cibercidadão" precisa de estar acostumado com estas práticas. Como o pode fazer? Como um cidadão pode e *deve* ter vários pares de chaves – um para cada tipo de actividade realizada. Imagine, na sua empresa, a forma como se encontram divididos os departamentos ou sectores, atribua assim uma chave para cada um, entregue a um responsável, depois crie ou incentive que cada funcionário tenha o seu par de chaves *enquanto* membro deste ou daquele departamento.

Gostaríamos de resumir aqui alguns pontos cruciais existentes no Linux Security HOWTO (2.3 "*What are you trying to protect?"* E o 2.4 "*Developing a security policy"*), os quais devem ser levados em consideração, antes mesmo de ser criada uma política de segurança para qualquer rede. Antes

de proteger o sistema, deve saber qual o tipo de ameaça que está a tentar proteger, e se atacado o que poderá estar em jogo... Vejamos então:

- *O risco é a possibilidade que um intruso possa ter sucesso ao tentar invadir os seus computadores. Um intruso pode, ao aceder aos seus ficheiros, danificar dados críticos? Não se esqueça, também, que ao possuir uma conta da sua rede, o intruso pode passar por si.*

- *As ameaças serão sempre no sentido de se obter acesso não autorizado na sua rede ou computador. Há portanto vários tipos de intrusos e, assim sendo, diferentes tipos de ameaça à sua rede.*

- *Há o curioso: esse tipo de intruso tem o interesse pelo tipo de dados e pelo sistema que possui.*

- *Há o malicioso: esse quer em síntese derrubar o sistema, destruir dados, destruir os documentos publicados no seu Web server, etc. É o chamado cracker.*

- *Há o intruso de "alto nível" (High-Profile): ele quer obter popularidade, mostrando todas as suas habilidades ao invadir o sistema.*

- *Há o competidor: esse quer conhecer os seus dados para poder obter algum ganho com isso.*

- *Por fim, "vulnerabilidade" descreve o quanto bem protegido é ou está o computador, e o que se perderá se alguém obtiver acesso não autorizado a algum(ns) computador(res).*

Portanto, crie uma *política de segurança* para a rede, que seja simples e genérica e que todos os utilizadores possam prontamente compreender e seguir. Pode proteger dados tanto quanto respeitar a **privacidade** dos utilizadores.

1.4 Objectivo deste livro

Em suma, a nossa intenção é mostrar – como já dissemos – as ferramentas necessárias para a construção e gestão de uma política de segurança em ambiente GNU/Linux. Todos são softwares livres e cobertos pela licença GPL (*General Public License* da GNU), assim como a documentação disponível. Para tanto, o nosso caminho não irá começar da segurança local (ou seja, uma rede interna, segundo a nomenclatura do *Site Security Handbook*) em direcção à segurança Extranet (rede externa). Contrariamente, iremos da segurança externa – passando obviamente pelas questões de segurança na rede Intranet, na medida em que o *firewall* também se ocupa com este ponto –, para a segurança interna.

---

[2] "Internet Security Architecture". *Computer Networks* 33 (1999), p. 787.

[3] RFC é o acrónimo de *Request for Comments*, um conjunto de documentos organizados pela INTERNIC que reune as informações sobre o TCP/IP e outros protocolos, assim como redes, segurança, correio eletrónico, etc.

[4] É nosso dever advertir para a má qualidade da tradução do livro **Sery**, Se puder, deve procurar o original do livro, até que uma revisão do texto seja feita.

2. 0 *Firewall*: duas soluções no ambiente Linux

2.1 Uma palavra inicial sobre os *firewalls*

O *Firewall* é uma “parede” que separa qualquer Intranet do *mundo exterior*, ou seja, da Internet. Ao mesmo tempo, o *firewall* também pode ser usado de forma eficiente numa Intranet que não tenha acesso à Internet, na medida em que permite a filtragem de *pacotes* que passam de uma máquina para outra numa Intranet.

Definição:
Um firewall, é um numeroso conjunto de *hardware* e *software,* que é elaborado para a protecção, de uma Intranet, de utilizadores potencialmente perigosos, na medida em que estes não possuem a devida autorização para acederem a estes serviços.

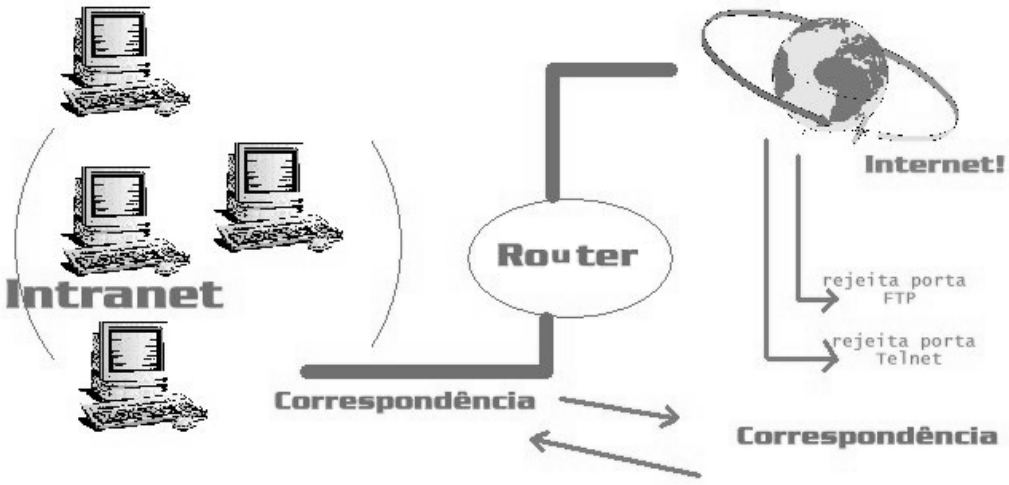
É importante desde já referir que a capacidade efectiva de controlo de um *firewall,* é implementada quando este é colocado entre uma rede local e a Internet, evitando dessa forma que o mundo possa ter acesso a dados particulares/privados.

É conveniente neste momento salientar que não nos encontramos na presença de um “programa” de execução fácil e simples, na medida em que um *firewall* possui inquestionavelmente uma natureza bastante mais abrangente.

Teoricamente não existiriam problemas de segurança, se os computadores ou a rede em questão estivesse isolada do resto do mundo. No entanto, uma Intranet é fundamentalmente "uma rede interna que utiliza as tecnologias da World-Wide-Web (WWW) para a partilha de tarefas e de informações, entre os vários departamentos e/ou locais remotos".[5]

Não é possível instalar, por exemplo, um servidor Apache numa determinada máquina e fechar o acesso aos seus dados nesse *firewall*, nem tão pouco poderemos fechar o acesso à totalidade dos computadores que constituem a rede mundial.

Este esquema, exemplifica a implementação simples de um *firewall*:



Deveremos ter presente que no esquema apresentado acima, a rede deve possuir obrigatoriamente um *host* (*bastion host*), o qual permite realizar o contacto com o mundo exterior, desde que se encontre ligado ao *router*.

[5]Definição recolhida de *SCO OpenServer© Internet Services* (v.5.0.4 may 1997), p. 11.

É bastante consensual que tal máquina deve ter a totalidade dos seus serviços inactivos (desligados), devendo ainda possuir o número mínimo de portas activas. Para tanto, o leitor deve seguir as orientações básicas para desligar os serviços no ficheiro *inetd*.

A distribuição Red Hat Linux possui uma ferramenta de *setup* em modo texto. Esta pode ser usada como uma interface valiosa para desligar (inactivar) serviços como *ftp*, *tftp*, *telnet*, etc.

Tanto pode ser executada através do *console*, por intermédio do comando **setup,** como directamente pelo comando **netsysv**, na medida em que a ferramenta **setup** é apenas e só, uma interface para os vários programas de configuração existentes no sistema operativo. É importante, neste momento também referir, que não é necessária a instalação do sistema gráfico *XFree*.

O *host* responsável pelo *firewall* da rede, deve ser uma máquina que possua o software mínimo necessário, devidamente instalado e com a prioridade óbvia de cumprir todas as funções de *firewall*.

2.2 Firewalls e acesso remoto: o Secure Shell

O acesso remoto ao *firewall* deve ser feito com a utilização do *software SSH Secure Shell* (*http://www.ssh.org*), e nunca com o *telnet*, o *rsh*, ou o *rlogin*, atendendo a que estes programas já foram desligados previamente.

O *ssh* é um conjunto completo de aplicações altamente necessárias, para que seja possível efectuar o *login* remoto, possuindo a capacidade de usar a criptografia, utilizando para a execução dessa tarefa uma autenticação forte, a comunicação entre redes e *hosts não autorizados.* Assim sendo, todas as comunicações são criptografadas.

A criptografia de chave pública ou assimétrica é usada para o intercâmbio ou partilha de "chaves" (num esquema de chaves públicas e privadas[6]) sendo ao mesmo tempo, um método de criptografia (IDEA, Blowfish, etc.) que também é usado para criptografar a sessão remota que vai ser aberta posteriormente.

O facto mais importante no *ssh,* é que o software inicia a criptografia ainda *antes* de ser iniciado o processo de autenticação, ou seja, *antes* do processo de verificação das *passwords*. Por isso, nenhuma *password* será enviada pela rede *em aberto* sem a utilização da criptografia.

A maior parte das distribuições actuais de GNU/Linux, incluem a versão 1.2.27 deste programa (a última versão é a 2.0.13, no momento em que se escreve este manual, chamada *ssh2*), ou mais recente.
No caso de obter o *ssh* através de um pacote RPM, este pode ser instalado com o comando "**rpm -ivh"**. Estes são os ficheiros instalados pelo comando anteriormente referenciado:

Ficheiros	Descrição
sshd	*Daemon que é executado no servidor, espera o pedido do cliente ssh, autentica a ligação e inicia a sessão.*
ssh ou slogin	Cliente: p*rograma usado para login e execução de outros comandos,*
scp	*Usado para copiar ficheiros de um computador para outro em segurança*
ssh-keygen	*Usado para criar chaves RSA*
ssh-agent	*Agente para a autenticação das chaves*
ssh-add	*Usado para registar novas chaves*
make-ssh-known-hosts	*Script perl usado para criar o ficheiro /etc/ssh_known_hosts a ser usado pelo DNS*

[6] Veja a explicação dada no Capítulo 1. Recordando: uma chave para criptografar e a outra para descriptografar. A chave pública é usada para criptografar, e a chave para descriptografar por sua vez é privada, mas *jamais poderemos derivar esta chave de descriptografar da outra*.

Realizada com êxito a instalação, iremos proceder à criação de uma chave RSA para o utilizador *root*, num *host* denominado *alpha,* usando o comando **ssh-keygen.**

Encontraríamos, por exemplo, a seguinte saída:

```
[root@alpha /]# ssh-keygen
Initializing random number generator...
Generating p: .................++ (distance 250)
Generating q: ......................++ (distance 314)
Computing the keys...
Testing the keys...
Key generation complete.
Enter file in which to save the key (/root/.ssh/identity):
Enter passphrase: (escreva a password - nada será ecoado)
Enter the same passphrase again: (idem)
Your identification has been saved in /root/.ssh/identity.
Your public key is:
1024 27 797979797979372973273927755665868302838974864836463468364
8
979479274937493749797397492794729749348793475154551542455251454514
686486348638463826427864863861525415199494879757808883282083082038
121122882018208201812802810803746836585979384 08 root@cipsga.org.br
Your public key has been saved in /root/.ssh/identity.pub
```

Depois de terem sido criadas as chaves (pública e privada) e destas terem sido previamente gravadas no directório da nossa escolha (aqui aceitaremos a melhor opção, que é precisamente a *default* do *ssh-keygen*), vamos tentar executar uma sessão segura.
No entanto, devemos ter em atenção que é necessário termos carregado anteriormente o programa servidor do *ssh*, o qual tem o nome de *sshd.*

Vamos executar o *sshd* manualmente. Para isso devemos digitar:

/etc/rc.d/init.d/sshd

É com este script, que a distribuição Red Hat Linux inicia o carregamento em toda inicialização (o *daemon*), sem que exista a necessidade deste ser executado manualmente.

Seguidamente, deve-se proceder à verificação com a ferramenta **netsysv** do Red Hat Linux, para que se ter a certeza que o *sshd* se encontra marcado para ser executado de forma automática.

Chegado a este ponto, deveremos ter o máximo de atenção, porque se o *ssh* (o cliente) não encontrar o *sshd* em execução, vai permitir a execução de uma sessão não segura por intermédio do *rsh*. Isto mesmo pode ser verificado no exemplo dado a seguir:

```
[root@beta /root]# ssh alpha
Secure connection to alpha refused; reverting to insecure method.
Using rsh. WARNING: Connection will not be encrypted.

Password:  (...)
```

Precisamente neste momento em que temos a plena certeza que o *sshd* se encontra activo no *host alpha*, vamos carregar noutra máquina (a qual vamos designar por *beta*), o cliente *ssh* para fazer a ligação com o *host alpha*. Vejamos:

```
[root@beta /]# ssh alpha.cipsga.org.br
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? Yes
Host 'beta.cipsga.org.br' added to the list of known hosts.
Creating random seed file ~/.ssh/random_seed. This may take a while.
root@alpha.cipsga.org.br's password: [escreve a password - nada é
ecoado]
Last login from: Sat Jan 29 23:12:00 2000 from alpha
[root@alpha /root]#
```

Seguidamente, vamos perceber as mensagens que são apresentadas no ecrã.

No momento em que o servidor *sshd* no *host alpha* recebe a solicitação, este adverte que não encontra a *host key* na lista de *hosts* conhecidos.

Mesmo assim, vamos continuar a aceitar a ligação. O *sshd* vai colocar a máquina *beta* na lista e continuar com o processo de autenticação.
Após o processo de autenticação ter sido executado com êxito, iremos receber o terminal virtual.

No caso de ter sido realizado o *logout* e entrarmos novamente com um pedido de sessão *ssh*, o *daemon* vai realizar a leitura do ficheiro **~/.ssh/ randon_seed** permitindo uma sessão segura e a autenticação padrão do sistema.

A importação da chave pública (armazenada no ficheiro *identity.pub*) para a máquina remota deve ser o método escolhido. Assim, é permitido ao programa a utilização da autenticação baseada em RSA.
Atendendo a que a máquina em questão se encontra preparada para *firewall*, esta não nos permite a transferência do ficheiro (ftp, NFS, etc.), por isso, iremos utilizar a disquete, usando um dos programas das ferramentas **mtools,** o **mcopy**.

Coloque uma disquete na unidade *a:* e digite**:**

"mcopy ~/.ssh/*.pub a:"

Através da execução deste comando, iremos copiar para o directório *remoto* **~/.ssh** a chave pública (Atenção: exclusivamente a nossa chave pública. A chave privada **não** deve ser divulgada), mas agora com um novo nome que vai ser utilizado nas ligações, a saber, **authorized_keys**.

Este ficheiro corresponde ao ficheiro convencional **.rhosts** . Ele possui uma *chave por cada linha*, suportando por isso diversas chaves públicas de diferentes utilizadores e *hosts*.
Atenção para não truncar o ficheiro, porque as chaves são longas e devem estar localizadas numa *única* linha.