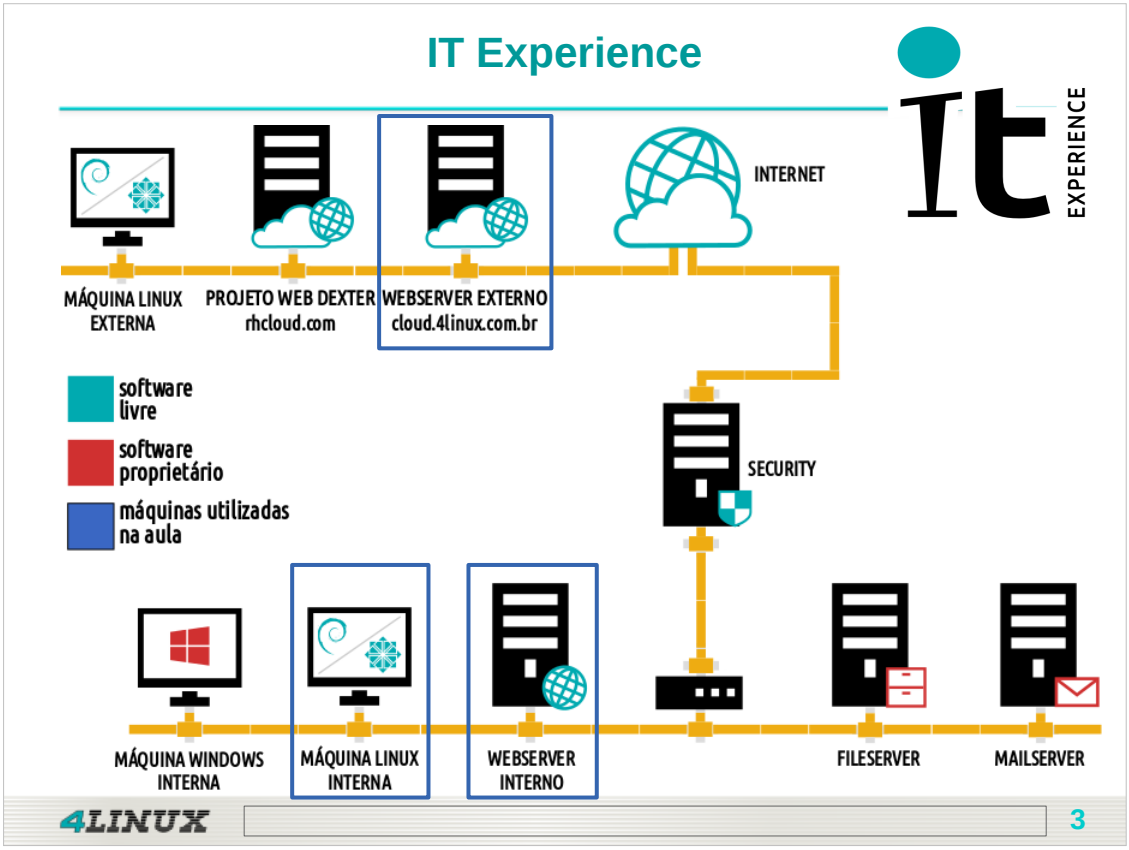
Curso 450

Linux Fundamentals in Cloud

Versão 2015_3.0

Servidor SSH, TCPWrapper e Migração do Site

Anotações:


IT Experience
IT EXPERIENCE
INTERNET
MÁQUINA LINUX EXTERNA
PROJETO WEB DEXTER rhcloud.com
WEBSERVER EXTERNO cloud.4linux.com.br
software livre
software proprietário
máquinas utilizadas na aula
SECURITY
MÁQUINA WINDOWS INTERNA
MÁQUINA LINUX INTERNA
WEBSERVER INTERNO
FILESERVER
MAILSERVER
4LINUX
3

Anotações:

Objetivos da Aula

Aula 06 (parte 1/2)

- Realizar acesso e cópias através do SSH;
- Ajustar configurações do servidor SSH;
- Configurar acesso com uso de chaves entre os servidores **Webserver Interno** e **Webserver Externo**;

Anotações:

Objetivos da Aula

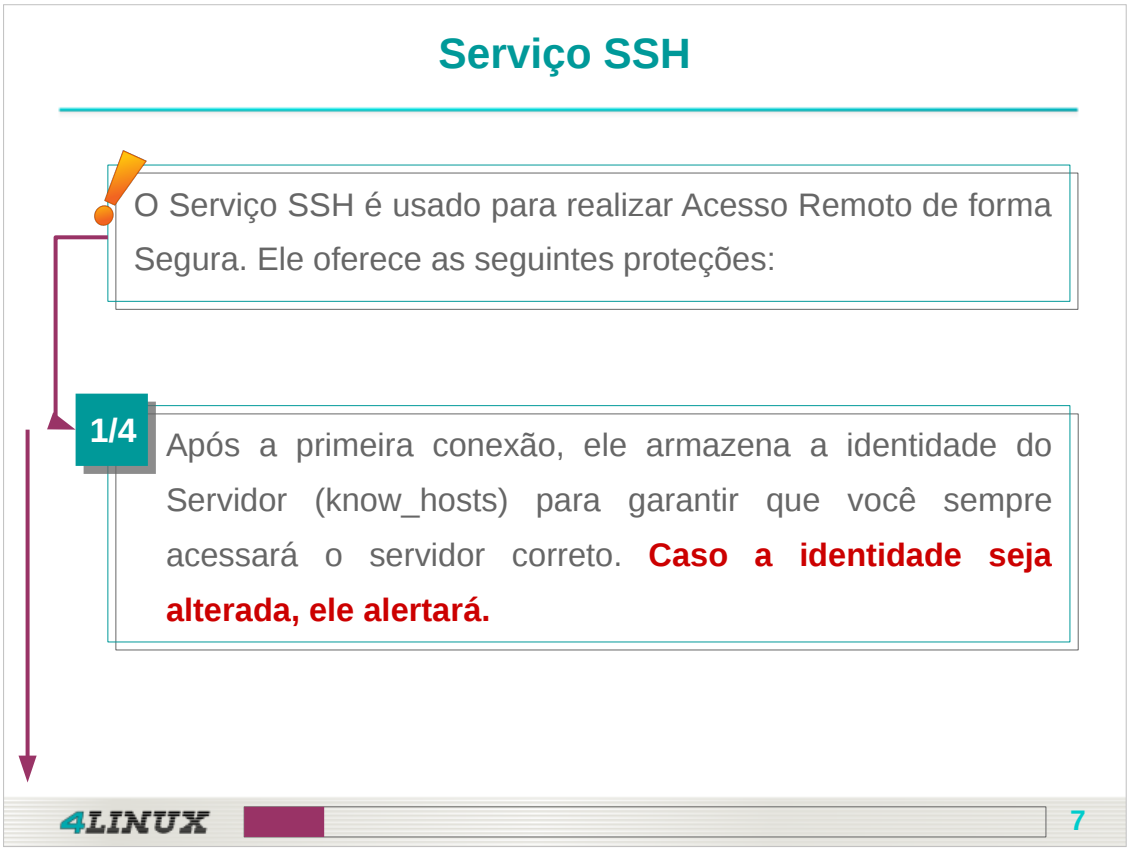
Aula 06 (parte 2/2)

- Entender o serviço TCPWrappers;
- Definir acessos ao servidor **Webserver Interno**:
  - Configurar o arquivo /etc/hosts.allow;
  - Configurar o arquivo /etc/hosts.deny.

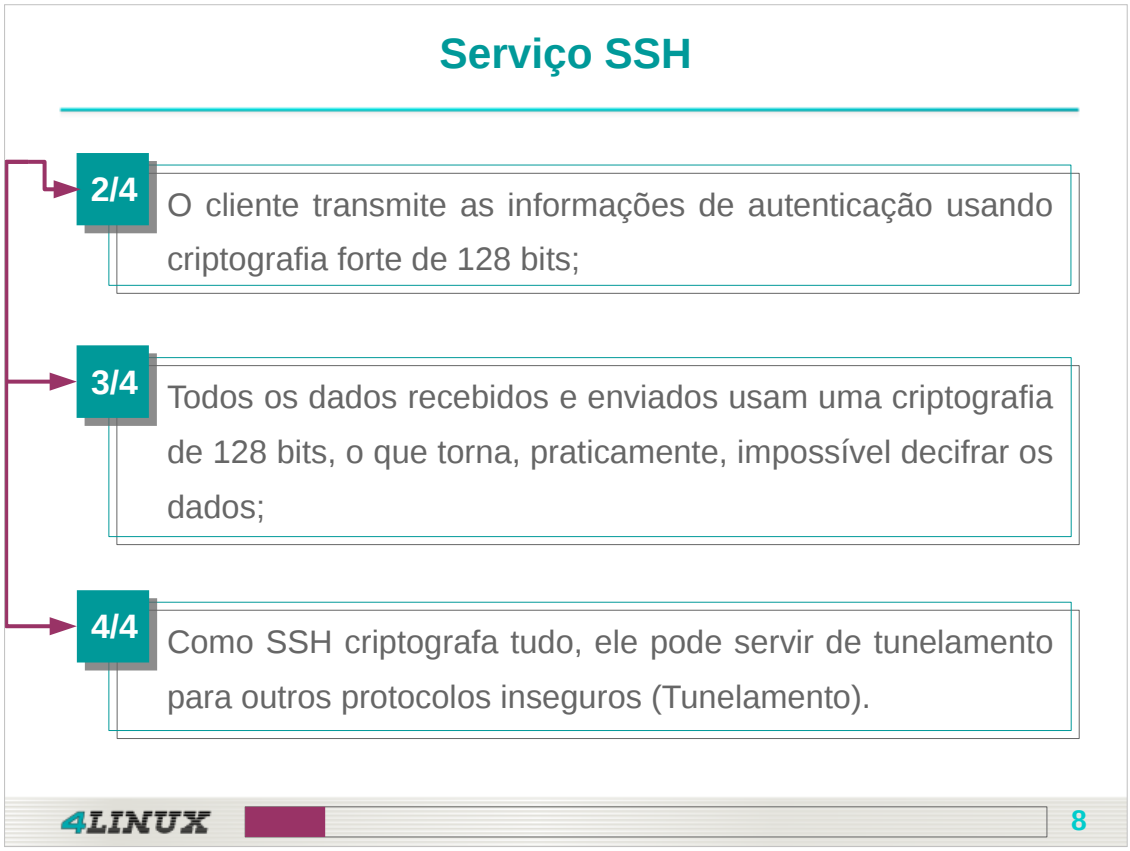
Anotações:

Realizar a Administração de Servidores Remotamente é uma das principais qualidades em sistemas Unix/Linux devido a flexibilidade e rapidez que a mesma pode proporcionar.

Nessa aula, aprenderemos o Serviço SSH (Secure Shell) que nos possibilita acessar máquinas remotas com segurança para realizar tarefas administrativas em sistemas Linux.


Serviço SSH
O Serviço SSH é usado para realizar Acesso Remoto de forma Segura. Ele oferece as seguintes proteções:
1/4
Após a primeira conexão, ele armazena a identidade do Servidor (know_hosts) para garantir que você sempre acessará o servidor correto. Caso a identidade seja alterada, ele alertará.
4LINUX
7

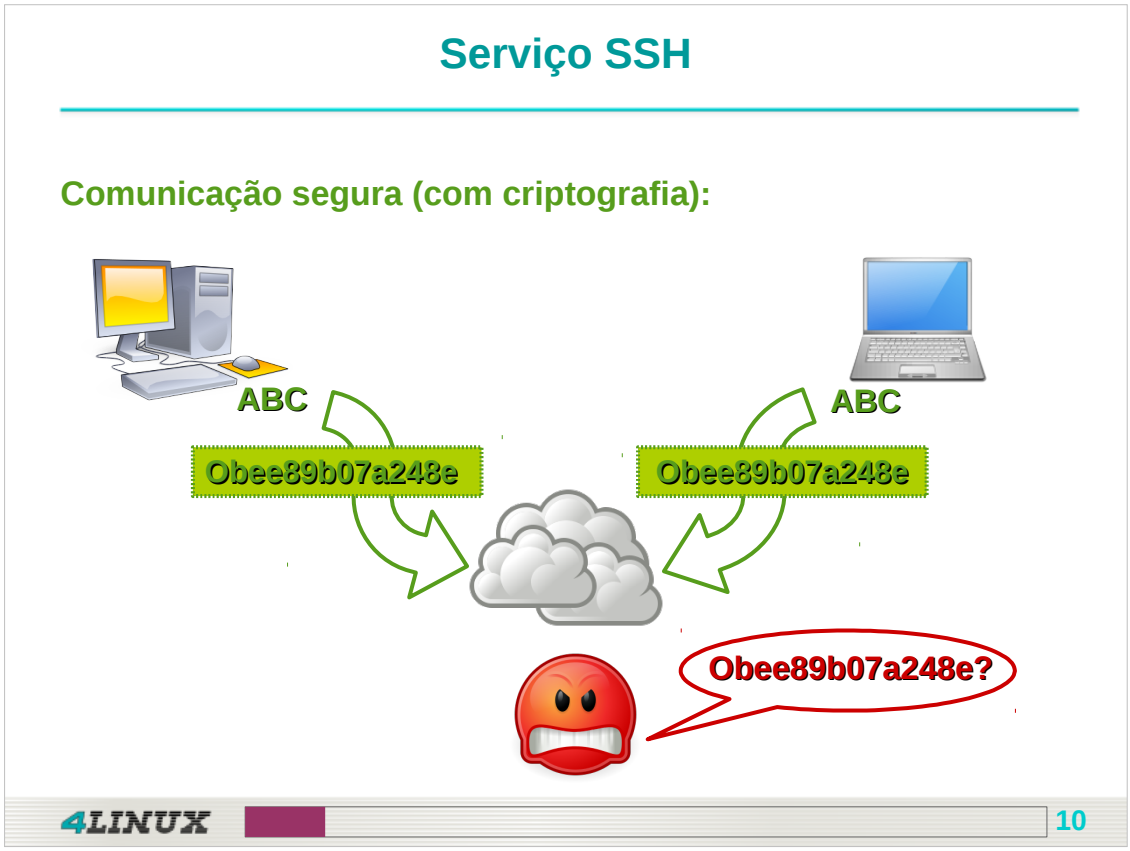
Anotações:


Serviço SSH
2/4
O cliente transmite as informações de autenticação usando criptografia forte de 128 bits;
3/4
Todos os dados recebidos e enviados usam uma criptografia de 128 bits, o que torna, praticamente, impossível decifrar os dados;
4/4
Como SSH criptografa tudo, ele pode servir de tunelamento para outros protocolos inseguros (Tunelamento).
4LINUX
8

Anotações:

Sistemas não criptografados

Existem sistemas de comunicação que não utilizam criptografia como por exemplo o FTP e o TELNET, o SSH surgiu como uma alternativa a este tipo de sistema



Por que o SSH é um protocolo tão seguro?

O SSH é considerado seguro por utilizar algumas funcionalidades de transmissão e autenticação de usuários, provavelmente a principal dessas funcionalidades é o sistema de chaves criptográficas, onde uma chave criptográfica pública é utilizada para autenticar um computador remoto, Esse sistema de chaves também pode ser utilizado para autenticar usuários no sistema.

Por que SSH?

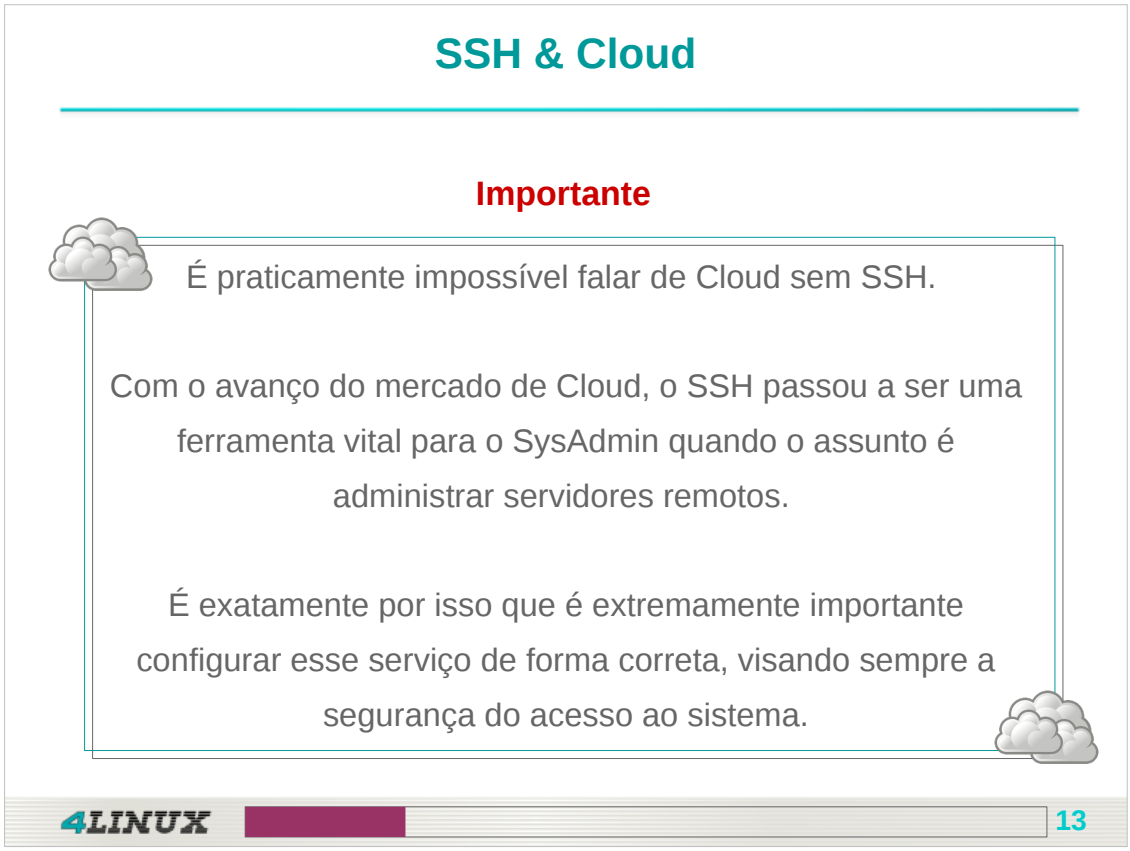
Existe uma variedade de ferramentas que podem ser usadas para romper ou interceptar dados de uma comunicação com o objetivo de conseguir acesso a um sistema como, por exemplo, usar um sniffer para capturar dados que estão trafegando na rede;

Com o SSH essa ameaça é quase nula, pois o cliente e o servidor SSH usam assinaturas digitais para verificar a sua identidade. Além disso, toda a comunicação entre eles é criptografada.

Anotações:

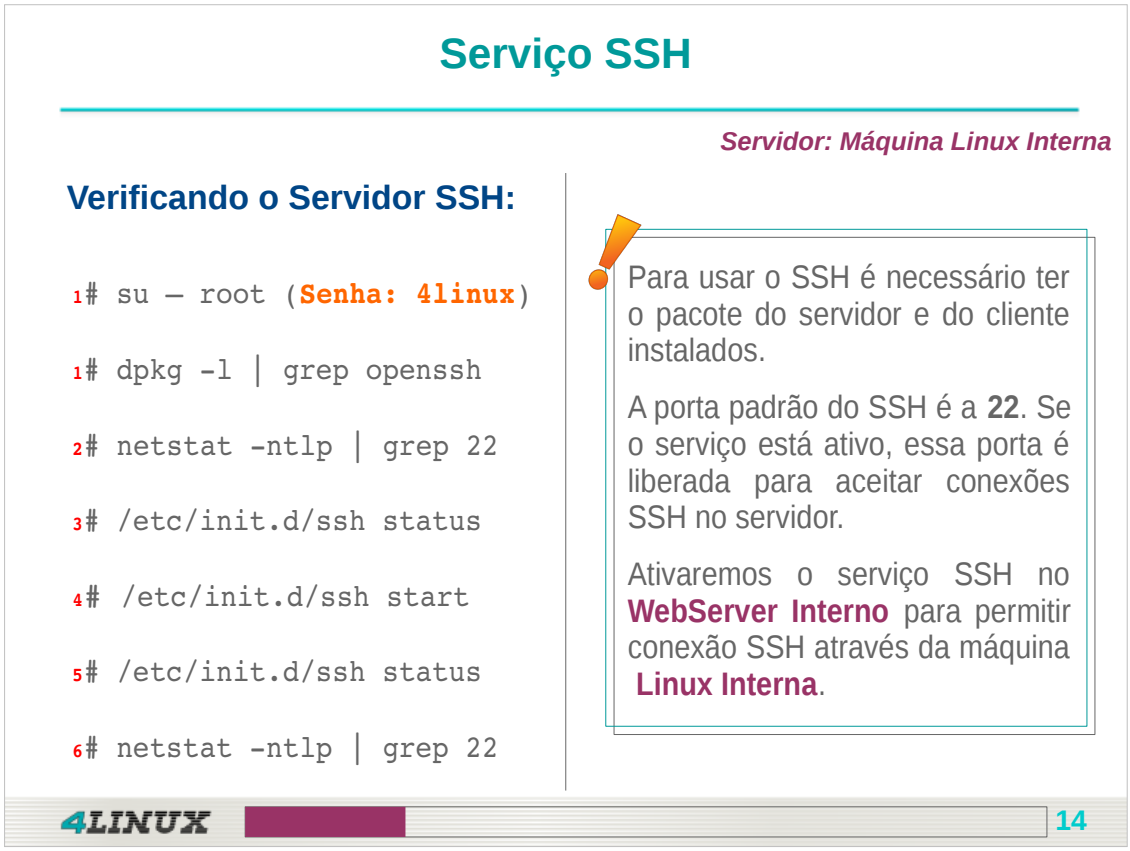
Conexão Criptografada

O SSH utiliza em sua configuração padrão o algoritmo de criptografia de dados RSA, considerado o melhor método já implementado para autenticação de chaves.



SSH & Cloud

Ao contratarmos serviços na nuvem é muito comum que o cliente receba do fornecedor do serviço uma chave com a qual será feita a conexão no servidor, esta chave é acionada através do comando **“ssh”** com o parâmetro **“-i”** seguido do caminho para a chave que será responsável pela autenticação do usuário.



Porta de conexão do SSH

Em geral o SSH utiliza a porta 22/tcp porém este padrão pode ser alterado através de seu arquivo de configuração, procedimento que veremos durante esta aula.

Serviço SSH

Servidor: Máquina Linux Interna

Primeiro acesso ao WebServerInterno a partir da máquina Linux Interna:

```
1# rm /root/.ssh/known_hosts
```

Formas de fazer um acesso remoto:

```
2# ssh -l root 192.168.200.20
```

ou

```
3# ssh root@192.168.200.20
The authenticity of host '192.168.200.20 (192.168.200.20)'
can't be established.
RSA key fingerprint is
55:02:60:1e:6c:27:cf:00:5a:73:ce:c5:47.
Are you sure you want to continue connecting (yes/no)?
yes
```

```
4# hostname
5# exit
```

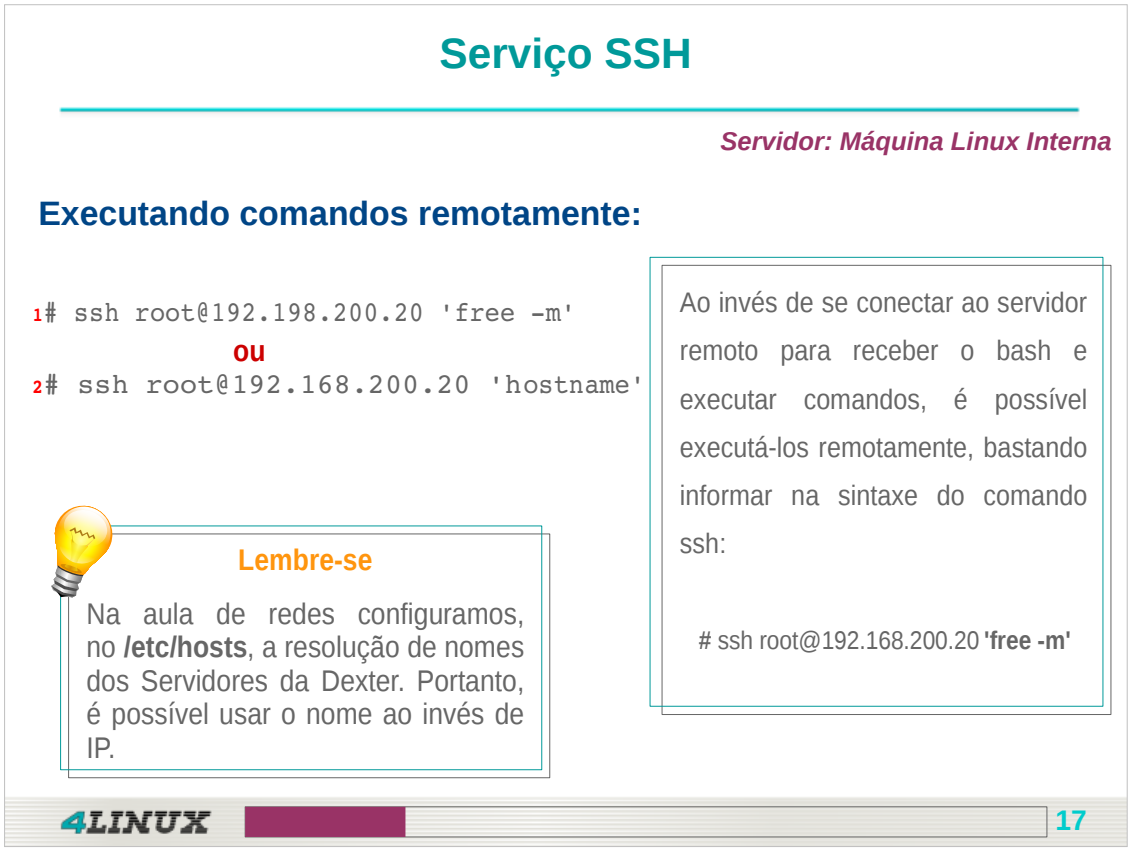
NOTA: No primeiro acesso, será solicitado que você aceite a identificação do servidor que será armazenada no arquivo **~/.ssh/known_hosts.**

Anotações:

O arquivo known_hosts

O arquivo de configuração known_hosts fica localizado dentro do diretório oculto **.ssh** que fica dentro da home de usuário.

Este arquivo possui a função de armazenar os nomes de hosts e as chaves dos equipamentos acessados via SSH, por padrão essas entradas são armazenadas em hashes, sendo que caso informações como a chave ou o endereço IP mudem o SSH emitirá um aviso durante a tentativa de conexão.



Como remover uma entrada no arquivo know_hosts?

Caso uma entrada tenha sido alterada propositalmente é possível removê-la do arquivo know_hosts através do comando ssh-keygen -R seguido do nome (ou endereço) do host que deverá ser removido.

Serviço SSH

Servidor: Máquina Linux Interna

Copiando o arquivo /etc/hostname do servidor Web para o servidor local em /tmp:

```
1# scp root@192.168.200.20:/etc/hostname  /tmp/

2# cat /tmp/hostname
```

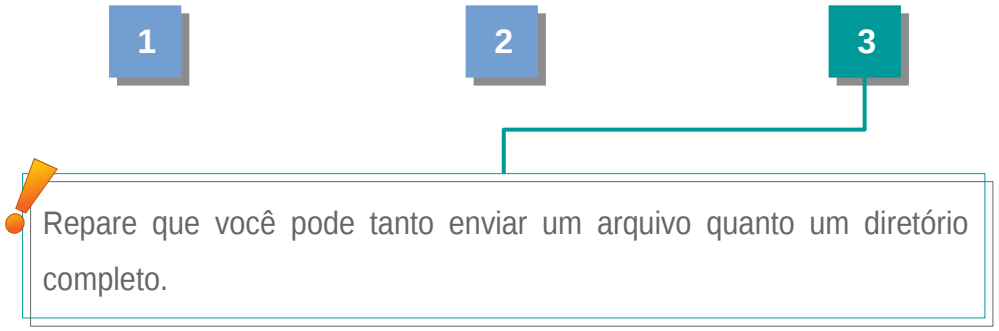

Anotações:


Serviço SSH
Servidor: Máquina Linux Interna
1
2
3
Copiando o diretório /etc/home do servidor Web para o servidor local em /tmp:
1# scp -P 22 -r root@192.168.200.20:/home /tmp/
2# ls /tmp/home
NOTA: A sintaxe do scp é semelhante ao cp, primeiro origem e depois destino.
-r → Copiar recursivamente. Usado para enviar ou copiar diretórios completos.
-P → Usado para especificar a porta, diferente do ssh, o -P precisa ser MAÍUSCULO.
4LINUX
19

Anotações:

Serviço SSH

Servidor: Máquina Linux Interna

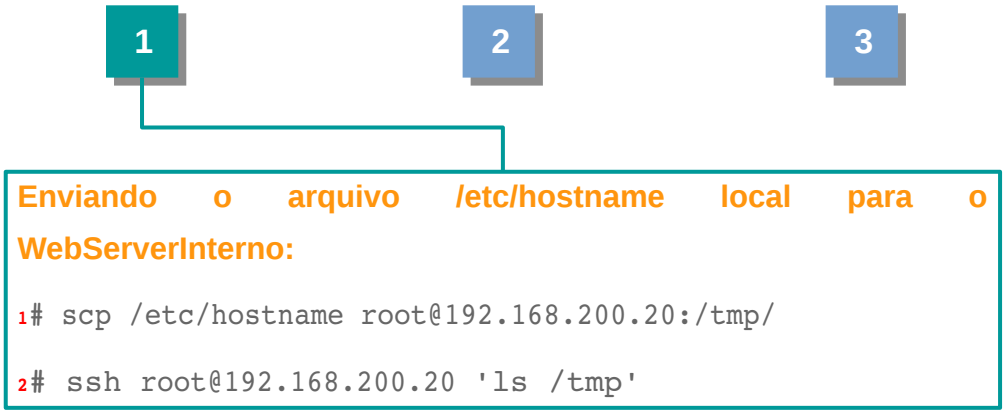


Repare que você pode tanto enviar um arquivo quanto um diretório completo.

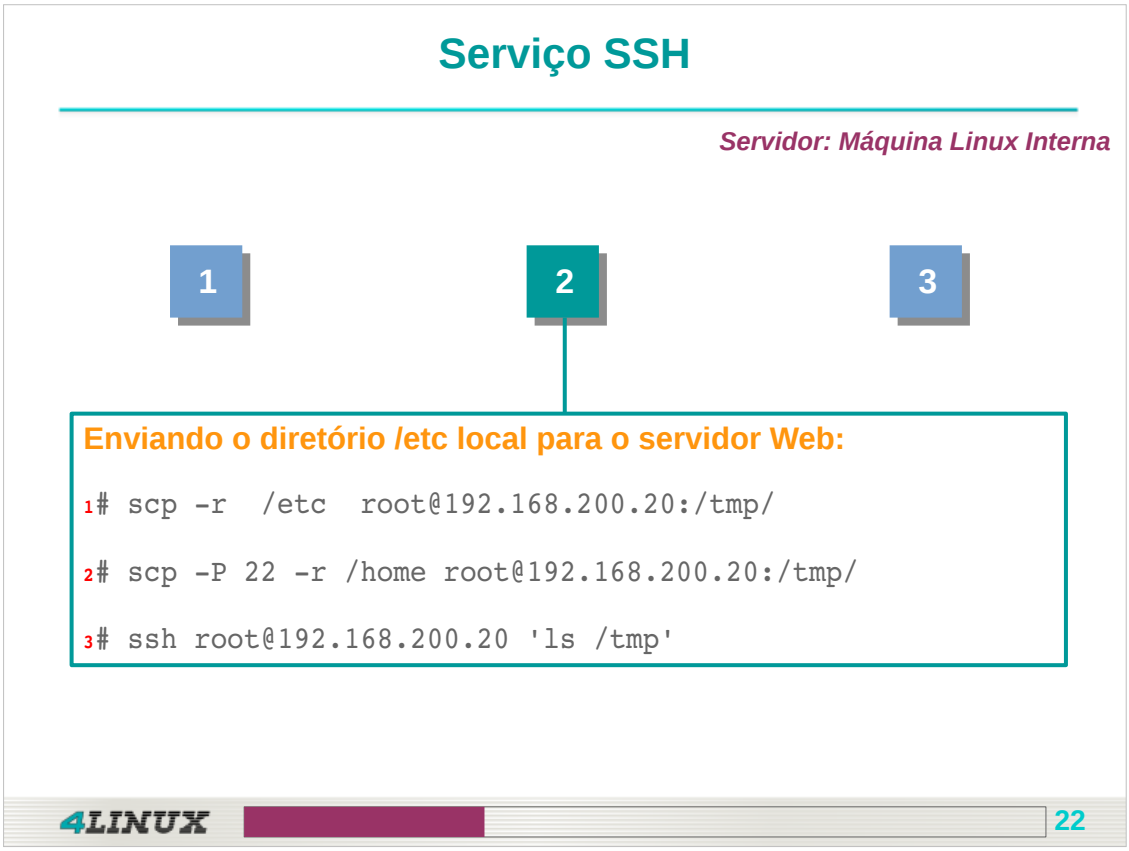
Anotações:

Serviço SSH

Servidor: Máquina Linux Interna



Anotações:


Serviço SSH
Servidor: Máquina Linux Interna
1
2
3
Enviando o diretório /etc local para o servidor Web:
1# scp -r /etc root@192.168.200.20:/tmp/
2# scp -P 22 -r /home root@192.168.200.20:/tmp/
3# ssh root@192.168.200.20 'ls /tmp'
4LINUX
22

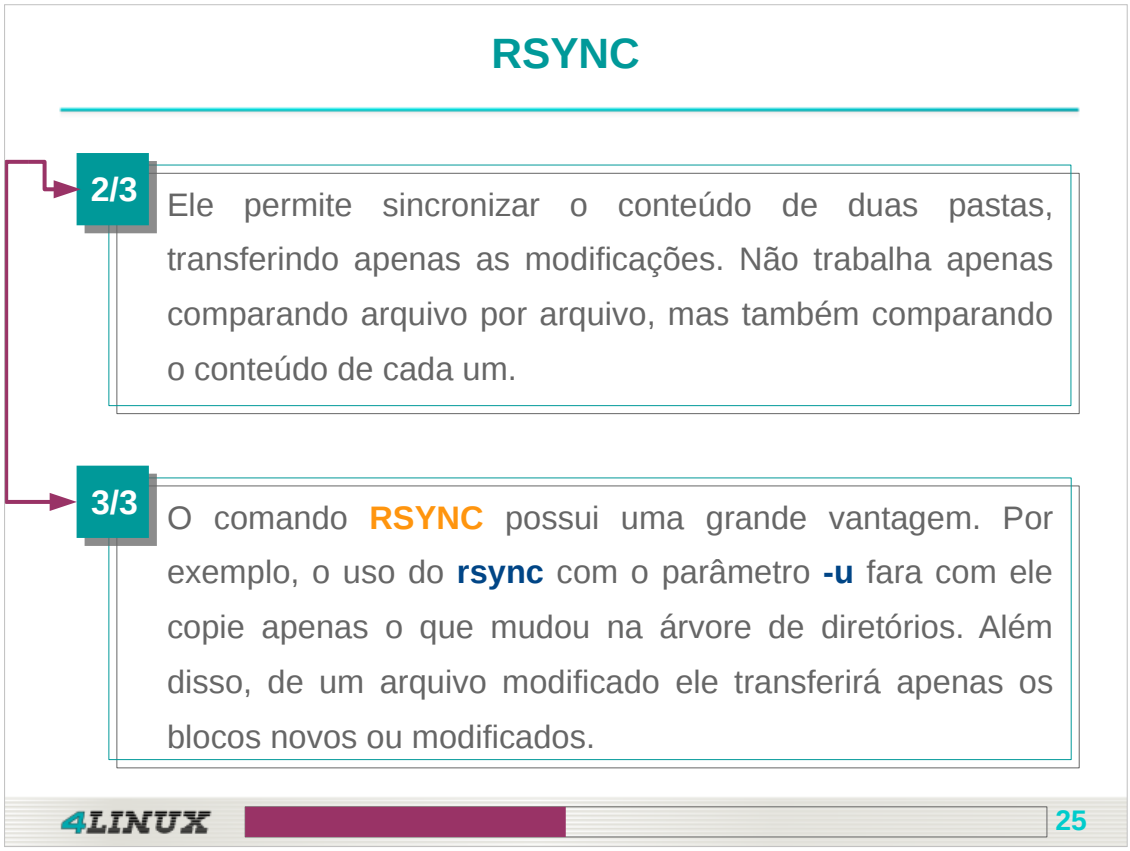
Anotações:


Serviço SSH
Servidor: Máquina Linux Interna
1
2
3
Repare que o comando SCP é uma via de mão dupla: você pode tanto enviar arquivos locais para um servidor como copiar de um servidor para a sua máquina local.
4LINUX
23

Anotações:


RSYNC
O RSYNC sincroniza arquivos e diretórios localmente ou remotamente, fazendo uma cópia exata dos arquivos.
1/3
O RSYNC (Remote Sync) é um utilitário para realizar cópias e sincronismo de arquivos (ou diretórios) localmente ou remotamente.
Uma forma simples de fazer back-ups completos de grandes quantidades de arquivos, ou mesmo partições inteiras, mantendo uma única cópia atualizada de tudo em um HD externo ou num servidor remoto.
4LINUX
24

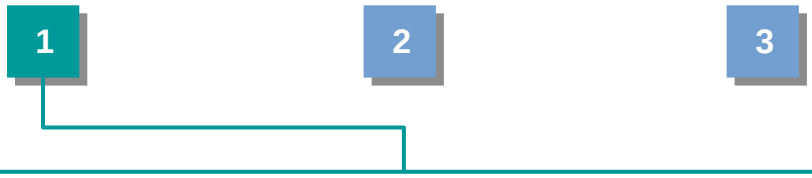
Anotações:


RSYNC
2/3
Ele permite sincronizar o conteúdo de duas pastas, transferindo apenas as modificações. Não trabalha apenas comparando arquivo por arquivo, mas também comparando o conteúdo de cada um.
3/3
O comando RSYNC possui uma grande vantagem. Por exemplo, o uso do rsync com o parâmetro -u fara com ele copie apenas o que mudou na árvore de diretórios. Além disso, de um arquivo modificado ele transferirá apenas os blocos novos ou modificados.
4LINUX
25

Anotações:

RSYNC

Servidor: Máquina Linux Interna



Enviando o arquivo /etc/hostname local para o WebserverInterno:

```
1# apt-get install rsync
2# rsync /etc/hostname root@192.168.200.20:/tmp/
3# ssh root@192.168.200.20 'cat /tmp/hostname'
```


Anotações:


RSYNC
Servidor: Máquina Linux Interna
1
2
3
Enviando o diretório /etc local para o Webserver Interno:
2# rsync root@192.168.200.20:/etc/hostname /tmp
3# cat /tmp/hostname
4LINUX
27

Anotações:


Serviço SSH
Servidor: Máquina Linux Interna
1
2
3
Ao realizar a cópia de arquivos, utilize a opção -p que preserva a permissão de dono, grupo, e datas de criação e modificação iguais ao arquivo de origem.
4LINUX
28

Anotações:

Laboratório Dexter

Servidor: Máquina Webserver Interno

Comando Wall:

- O comando **Wall** tem como característica mandar uma mensagem para todos os usuários que estiverem logados no servidor, constituindo-se em uma ferramenta muito útil e ágil para um administrador de rede que tem a intenção de fazer uma manutenção ou algum reparo no servidor e deseja avisar os usuários.
- Da máquina Linux Interna conecte-se à máquina **Webserver Interno**:

```
1# ssh suporte@192.168.200.20(senha @123Mudar)
```


Anotações:

Laboratório Dexter

Servidor: Webserver Interno

- O comando **w** é utilizado para verificar quais usuários e em quais terminais eles estão logados na sua máquina:
- Execute a partir da máquina Webserver Interno o seguinte comando:

```
1# w
```

```
 14:13:16 up  4:02,  2 users,  load average: 0,00, 0,01, 0,05
USER     TTY      LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1     10:30              24:28  6.42s  6.42s  w
suporte  pts/0    13:49     4.00s  0.01s  0.00s  -bash
```

- A partir da máquina **Webserver Interno** mande uma mensagem a todos que estão conectados no servidor:

```
2#echo "Dia 26/06 uma manutenção no server será realizada" | wall
```

- Visualize se a mensagem foi enviada para o terminal da máquina **Linux Interna**

Anotações:

Laboratório Dexter

Servidor: Webserver Interno

Comando Screen

1

Como um Sysadmin, é muito comum você ter a necessidade de rodar vários comandos de uma vez.

Você provavelmente abrirá várias sessões do **ssh** para isto, porém, existe uma maneira melhor, mais segura e recomendada.

2

Você pode, por exemplo, ter a necessidade de executar os comandos e querer que outra pessoa veja exatamente o que você está fazendo, como no caso de algum cliente ou algo do tipo.

Anotações:

Laboratório Dexter

Servidor: Webserver Interno/ Máquina Linux Interna

- Instale no Web Server Interno o screen e abra uma nova sessão para compartilhar a tela com outros servidores:

```
1# yum install screen
2# su - suporte
3# screen
```

- Agora acesse remotamente o servidor Webserver Interno através da máquina Linux Interna:

```
1# ssh suporte@192.168.200.20 (senha @123Mudar)
2# screen -x
```

NOTA: Repare que agora, tudo o que é feito no servidor Web Interno é possível você visualizar no terminal da máquina Linux Interna, podendo também enviar comandos ou escrever uma frase.

Anotações:

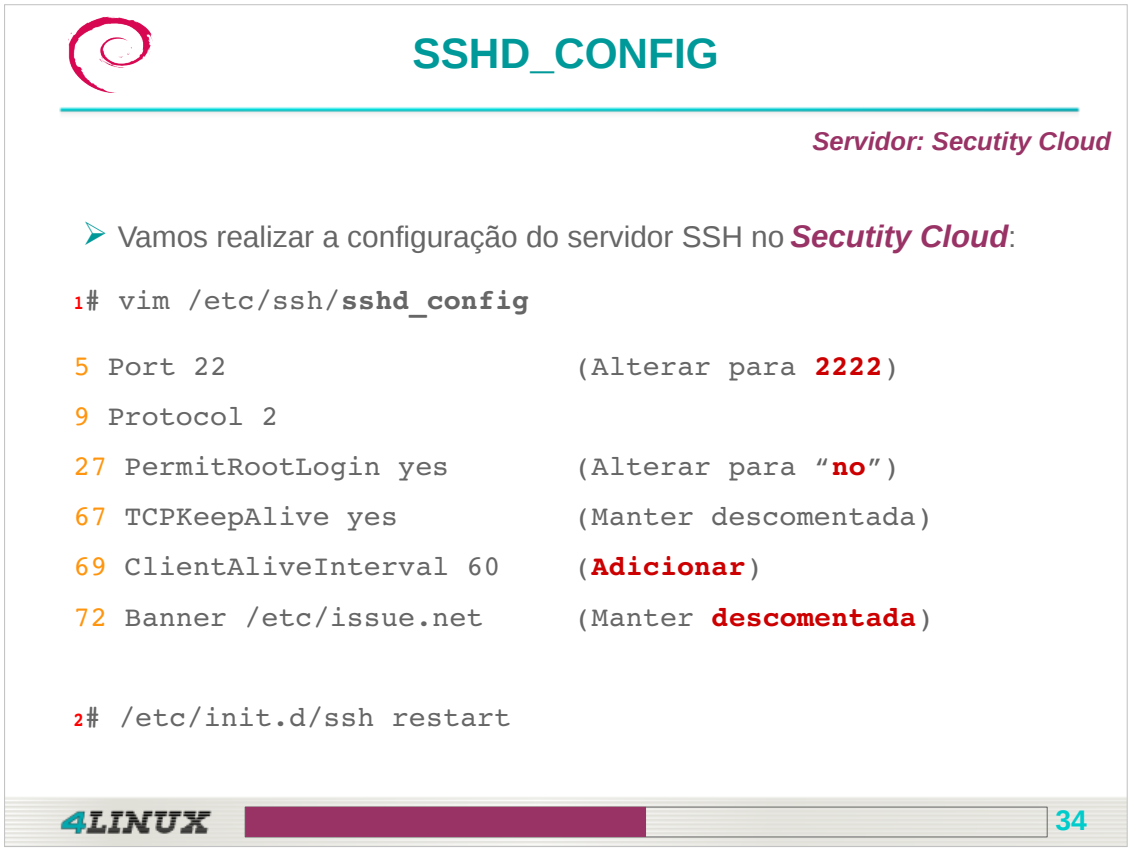
Configurando o SSH

- O **serviço SSH** possui 2 arquivos de configuração:
  - **sshd_config** → Configurações do Servidor SSH;
  - **ssh_config** → Configurações do Cliente SSH.

Servidor → Máquina que recebe um acesso remoto
Cliente → Máquina que realiza um acesso remoto

Todos os Servidores podem ser Cliente e Servidor!

Anotações:



Configurando o Servidor SSH:

Há diversos parâmetros de configuração que podem ser alterados de forma a ajustar seus parâmetros de funcionamento:

Opção	Descrição	Padrão
AllowGroups	Habilita acesso apenas para grupos especificados	*
AllowUsers	Habilita acesso apenas para usuários especificados	*
DenyGroups	Nega acesso apenas para grupos especificados	none
DenyUsers	Nega acesso apenas para usuários especificados	none
Port	Porta de acesso ao ssh	22
PermitRootLogin	Habilita/nega acesso do usuário root por ssh	Yes
X11Forwading	Habilita/nega acesso ao X (SSH com modo gráfico)	Yes
Banner	Habilita banner do issue.net	/etc/issue.net
LoginGraceTime	Tempo para se logar no servidor	120

Acesso Remoto Seguro

Servidor: Webserver Interno

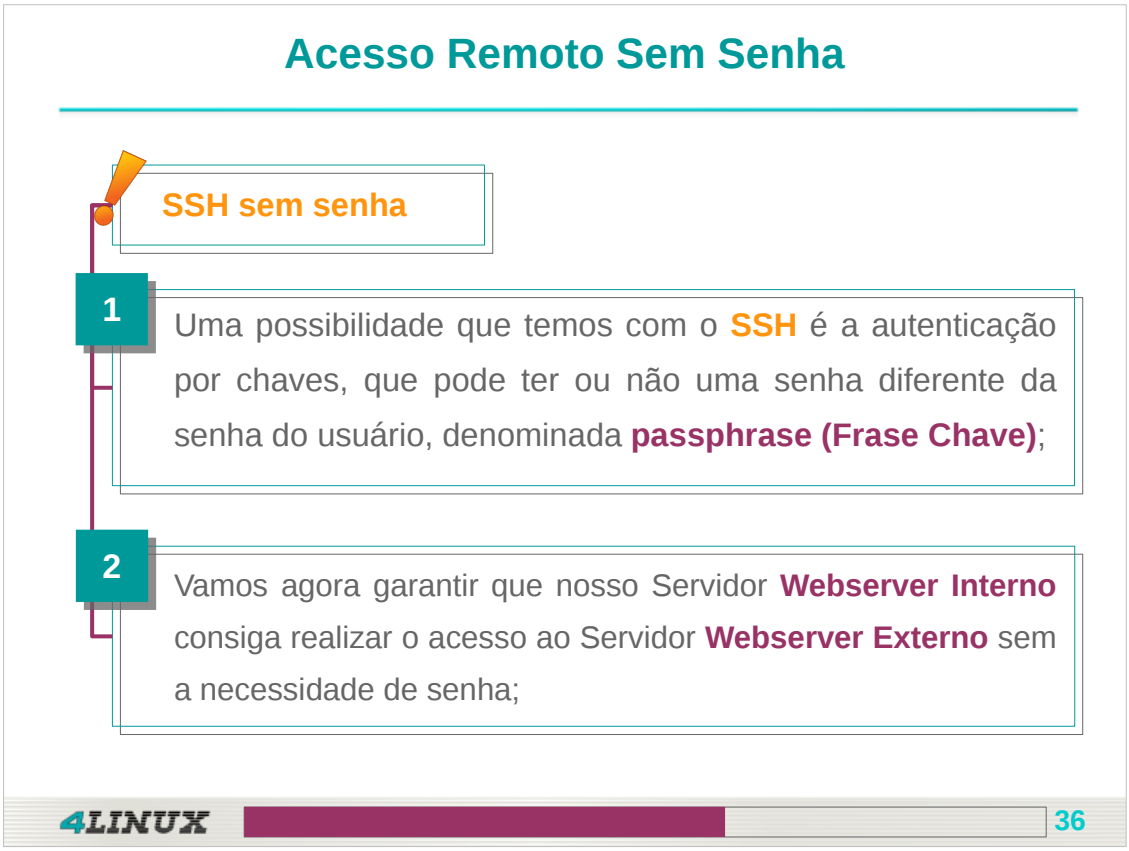
Acesse a máquina **Webserver Interno** e tente acessar remotamente a máquina Security Cloud.

```
1# ssh root@192.168.200.1
```

Uma vez que a porta padrão foi alterada e o acesso do root como medidas de segurança bloqueado, não será possível realizar acesso da maneira anterior!

```
2# ssh suporte@192.168.200.1 -p 2222
3$ su  (Senha: 4linux)
4# whoami
```

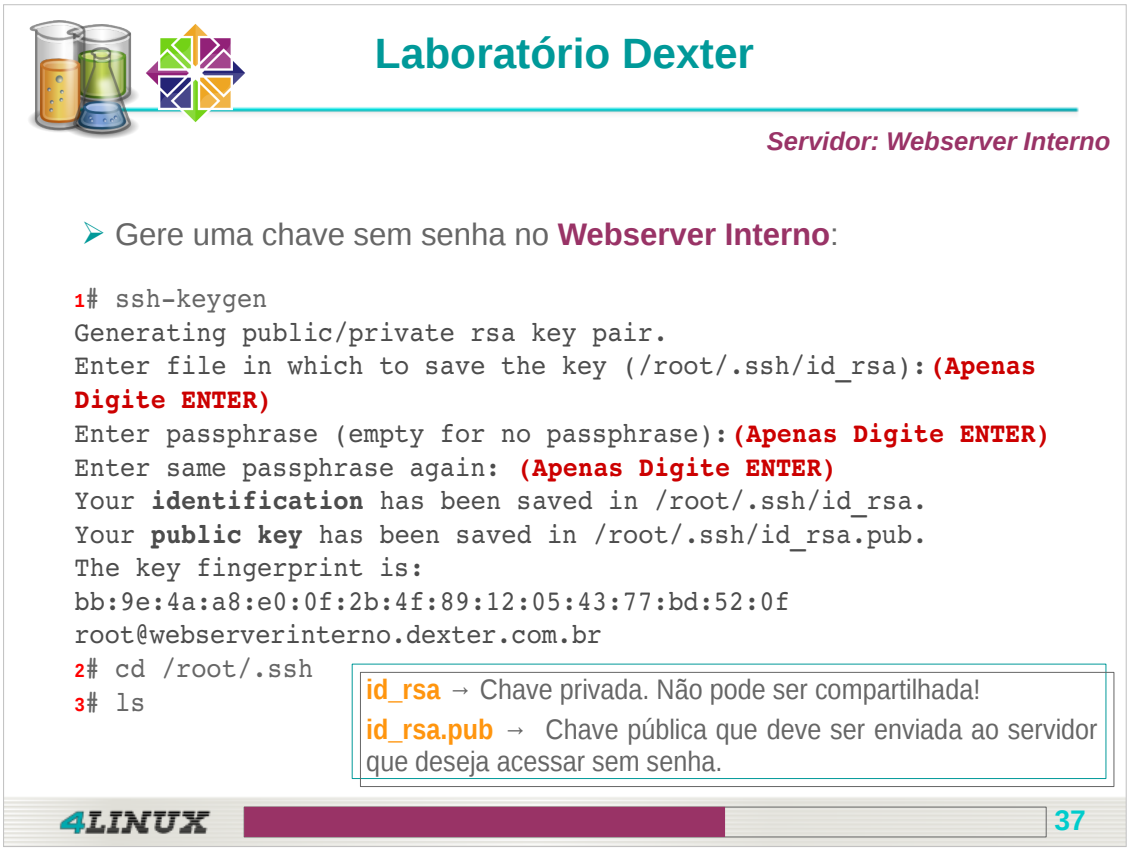

Anotações:



O Processo de criação de chaves

A criação de chaves assimétricas consiste na geração de dois arquivos que só terão funcionalidade se os dois trabalharem em conjunto. Ou seja, quando criamos um par de chaves será criada uma chave pública e uma chave privada. A chave privada é sua e absolutamente ninguém deve ter acesso a ela; a sua chave pública você coloca no servidor remoto.

Quando você tentar estabelecer uma conexão ela só será possível se a chave privada se encaixar na chave pública. Com esse sistema, existe apenas uma única chave privada que se encaixa em uma única chave pública.

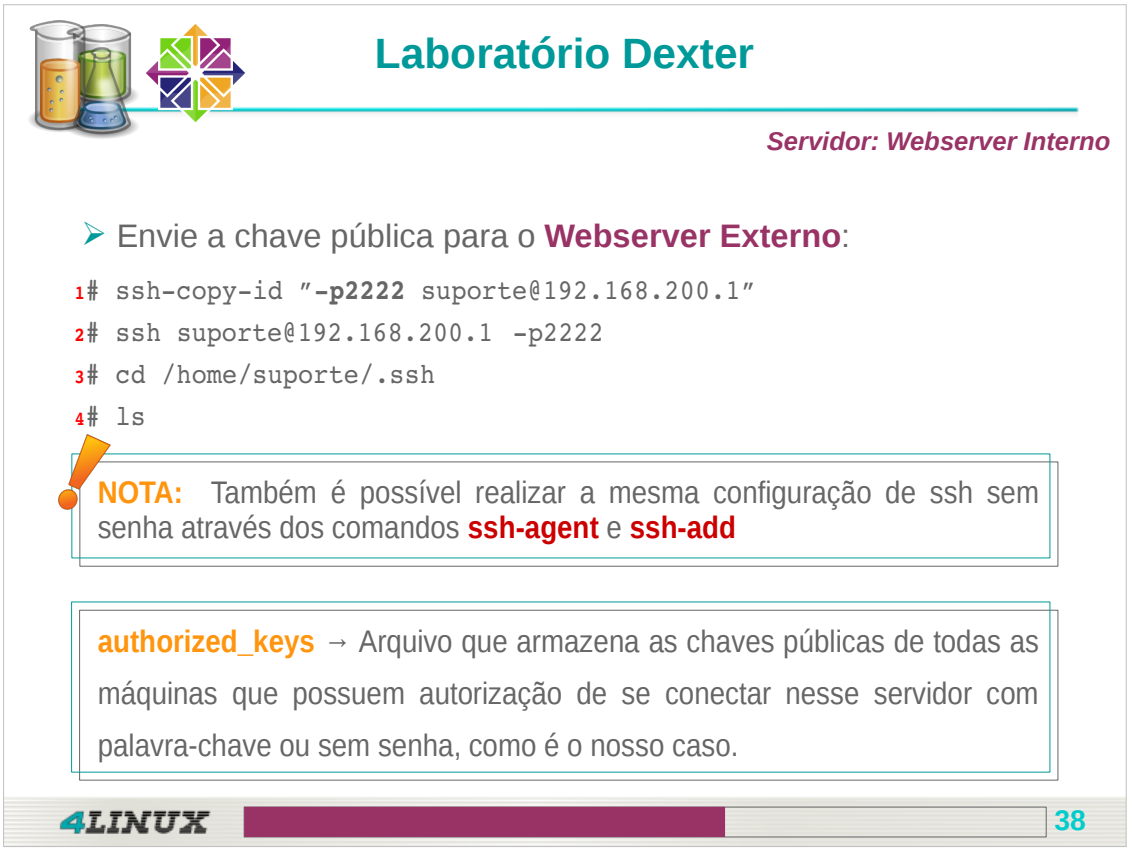


Geração de chaves

O comando acima gera os arquivos ".ssh/id_rsa" e ".ssh/id_rsa.pub" dentro do seu diretório home, que são, respectivamente, sua chave privada e sua chave pública. O ".ssh/id_rsa" é um arquivo secreto, que deve usar obrigatoriamente o modo de acesso "600", para evitar que outros usuários da máquina possam lê-lo.

Muitos servidores recusam a conexão caso os arquivos estejam com as permissões abertas.

Como só há um par que se completa, apenas quem possuir a chave privada poderá estabelecer uma conexão utilizando a respectiva chave pública.



O comando ssh-copy-id

Em resumo, o que o ssh-copy-id faz nada mais é do que copiar o conteúdo do arquivo ".ssh/id_rsa.pub", dentro do seu diretório home, para o arquivo ".ssh/authorized_keys" dentro do diretório home do servidor remoto, esta operação pode ser realizada manualmente.

Quando utilizar uma frase senha?

Criar um par de chaves assimétricas tem basicamente duas funções:

• Aumentar o nível de segurança - “Definir uma frase senha aumentará o nível de segurança de seu acesso”;

• Facilitar a execução de scripts remotamente - “Neste modelo não se deve definir uma frase senha”.

Limitando Acesso ao SSH

Servidor: Security Cloud

Alterando a opção “PermitRootLogin yes” para “no” no arquivo do servidor **Security Cloud** nós limitaremos o root de poder logar via SSH, porém todos os demais usuários do servidor ainda possuem acesso a realizar uma conexão SSH.

Vamos limitar esse acesso apenas ao usuário suporte:

```
1# vim +20 /etc/ssh/sshd_config
   AllowUsers suporte
2# /etc/init.d/ssh restart
```

Vá no **WebServer Interno** e teste o acesso:

```
3# ssh -p2222 -l helpdesk 192.168.200.1 Não Passou
4# ssh -p2222 -l suporte 192.168.200.1  Passou
```


Anotações:

Laboratório Dexter

Migração do Site – Dexter Courier:

Enfim, chegou o momento de migrarmos o site da Dexter Courier para a Cloud e essa responsabilidade está em suas mãos!

A equipe de desenvolvimento já disponibilizou o site da Dexter no ftp da empresa, realize o download do site compactado e realiza as tarefas para disponibilizar o site da Dexter Courier.

O comando ssh-copy-id

Em resumo, o que o ssh-copy-id faz nada mais é do que copiar o conteúdo do arquivo ".ssh/id_rsa.pub", dentro do seu diretório home, para o arquivo ".ssh/authorized_keys" dentro do diretório home do servidor remoto, esta operação pode ser realizada manualmente.

Quando utilizar uma frase senha?

Criar um par de chaves assimétricas tem basicamente duas funções:

• Aumentar o nível de segurança - “Definir uma frase senha aumentará o nível de segurança de seu acesso”;

• Facilitar a execução de scripts remotamente - “Neste modelo não se deve definir uma frase senha”.

Laboratório Dexter

Servidor: Webserver Externo

Migração do Site – Dexter Courier:

- Acesse o servidor Webserver Externo e realize os seguintes procedimentos:

```
1# ssh -l email@dominio.com.br cloud.4linux.com.br
2# cd /var/www/html
3# rm -rf /var/www/html/*
4# wget http://repositorio_interno/dexter.tar.gz
5# tar -xf dexter.tar.gz
6# /etc/init.d/apache2 restart
7# /etc/init.d/mysql restart
```

NOTA: Parabéns! A migração do site para a Cloud foi realizada com sucesso!

Anotações:

Laboratório Dexter

Servidor: Security Cloud / Linux Interna

Visualizar o site – Dexter Courier:

- Para visualizar o site da Dexter Courier, execute os seguintes comandos na máquina **Security Cloud**:

```
1# startcloud
2# Enter Auth Username: email@dominio.com.br
3# Enter Auth Password: senha para se logar na cloud
```

- Abra um navegador na máquina **Linux Interna**:

```
1# http://172.16.1.X
```

NOTA: Para saber o ip da cloud, acesse o servidor e execute o comando:

```
1# ifconfig venet0:0
```

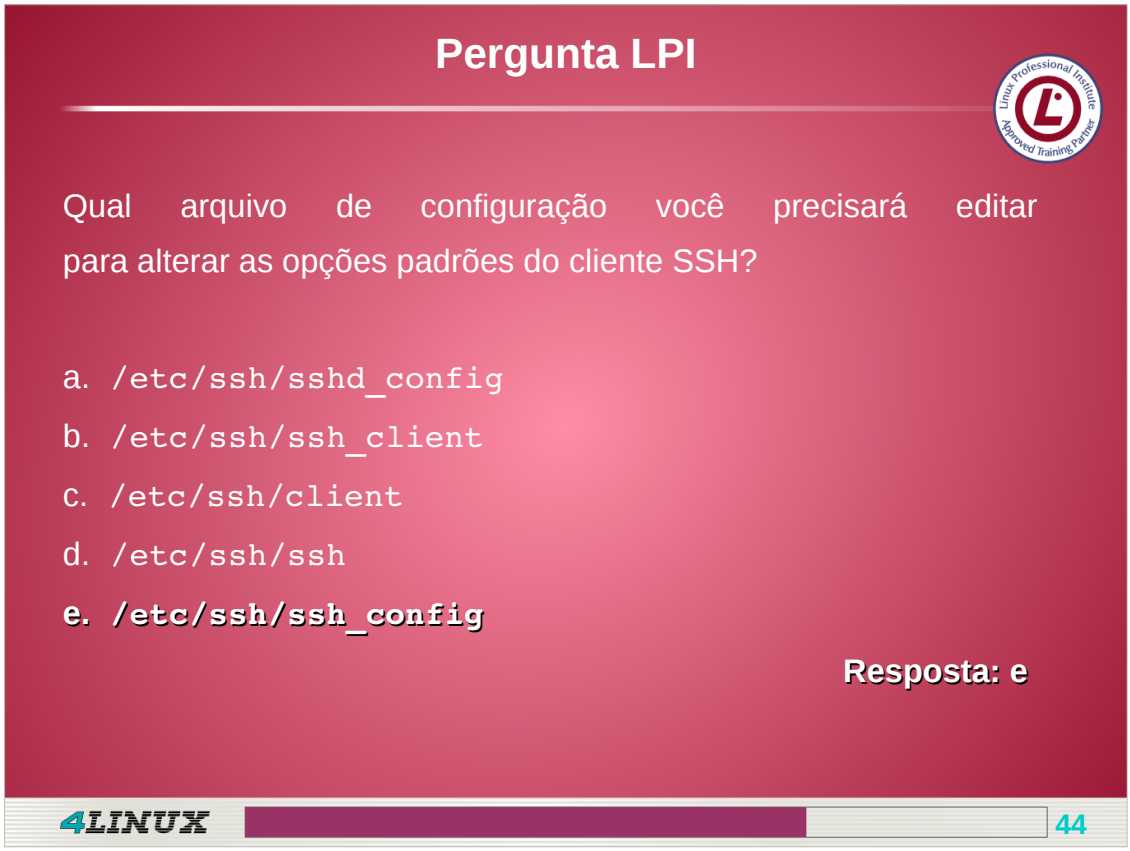

Anotações:

Pergunta LPI

Qual arquivo de configuração você precisará editar para alterar as opções padrões do cliente SSH?

a. `/etc/ssh/sshd_config`
b. `/etc/ssh/ssh_client`
c. `/etc/ssh/client`
d. `/etc/ssh/ssh`
e. `/etc/ssh/ssh_config`

Anotações:



Alternativa E: RESPOSTA CORRETA!

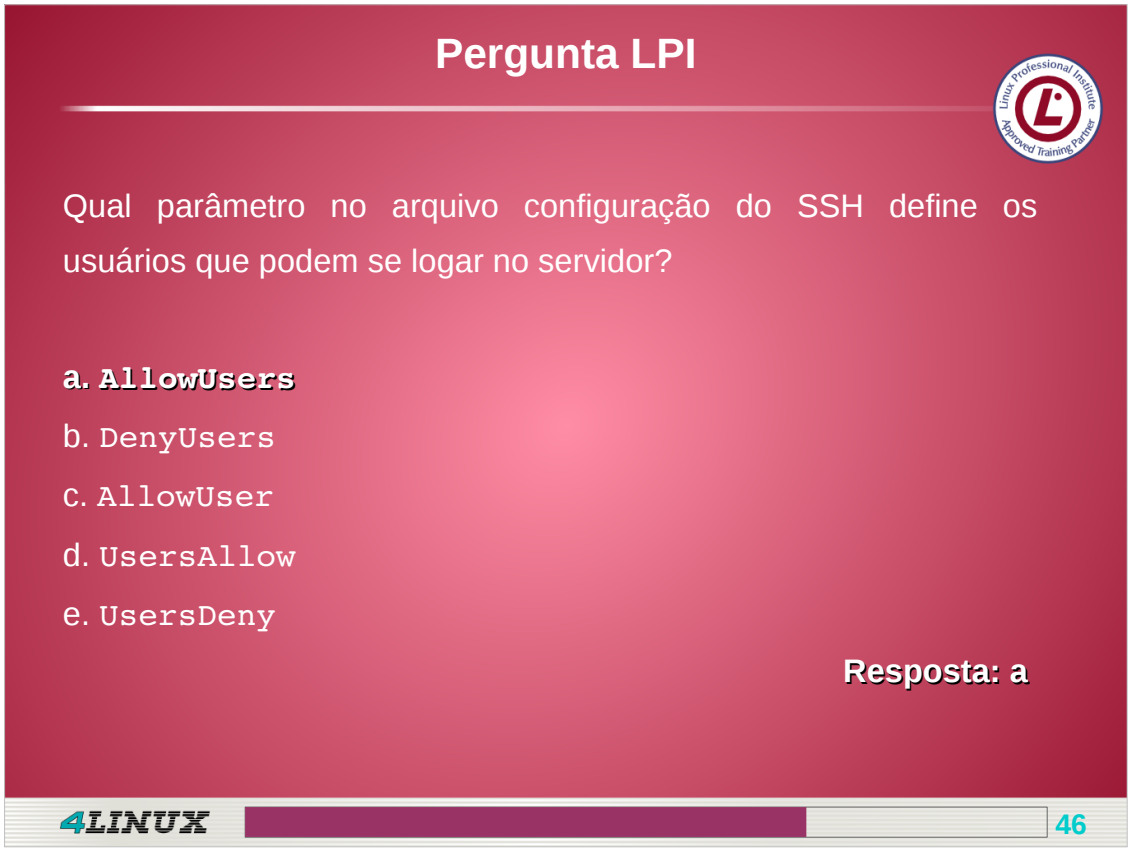
Fique atento! Muitas perguntas LPI trazem perguntas múltipla escolha com opções similares na resposta, como no exemplo acima, onde temos o arquivo sshd_config utilizado para configuração do servidor SSH e o arquivo ssh_config utilizado para configuração do cliente SSH.

Pergunta LPI

Qual parâmetro no arquivo configuração do SSH define os usuários que podem se logar no servidor?

a. `AllowUsers`
b. `DenyUsers`
c. `AllowUser`
d. `UsersAllow`
e. `UsersDeny`

Anotações:

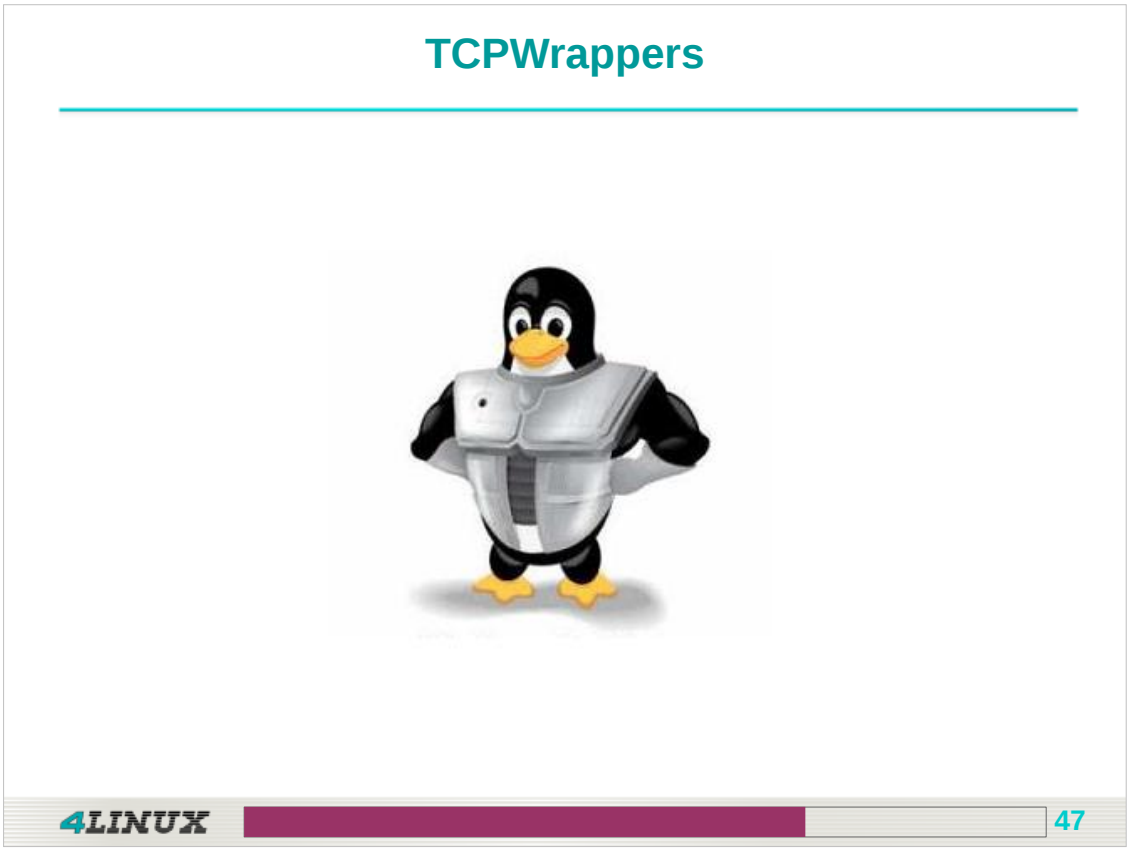


Alternativa A: RESPOSTA CORRETA!

A opção AllowUsers permite especificar uma lista de nomes de usuários que poderão acessar o ssh, conforme alternativa A.

NOTA:

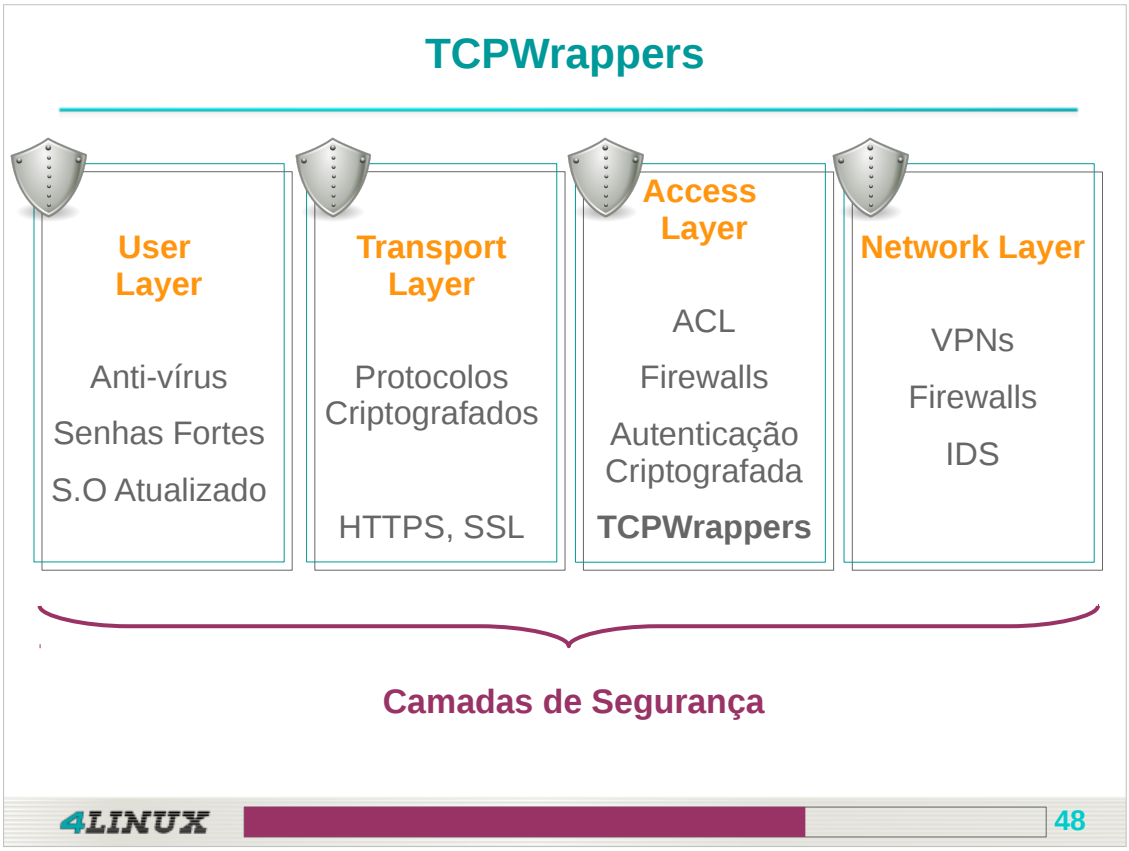
Outras diretivas podem ser utilizadas para Proibir o acesso de usuários (DenyUsers), Proibir o acesso de grupos (DenyGroups) e Liberar o acesso de Grupos (AllowGroups).



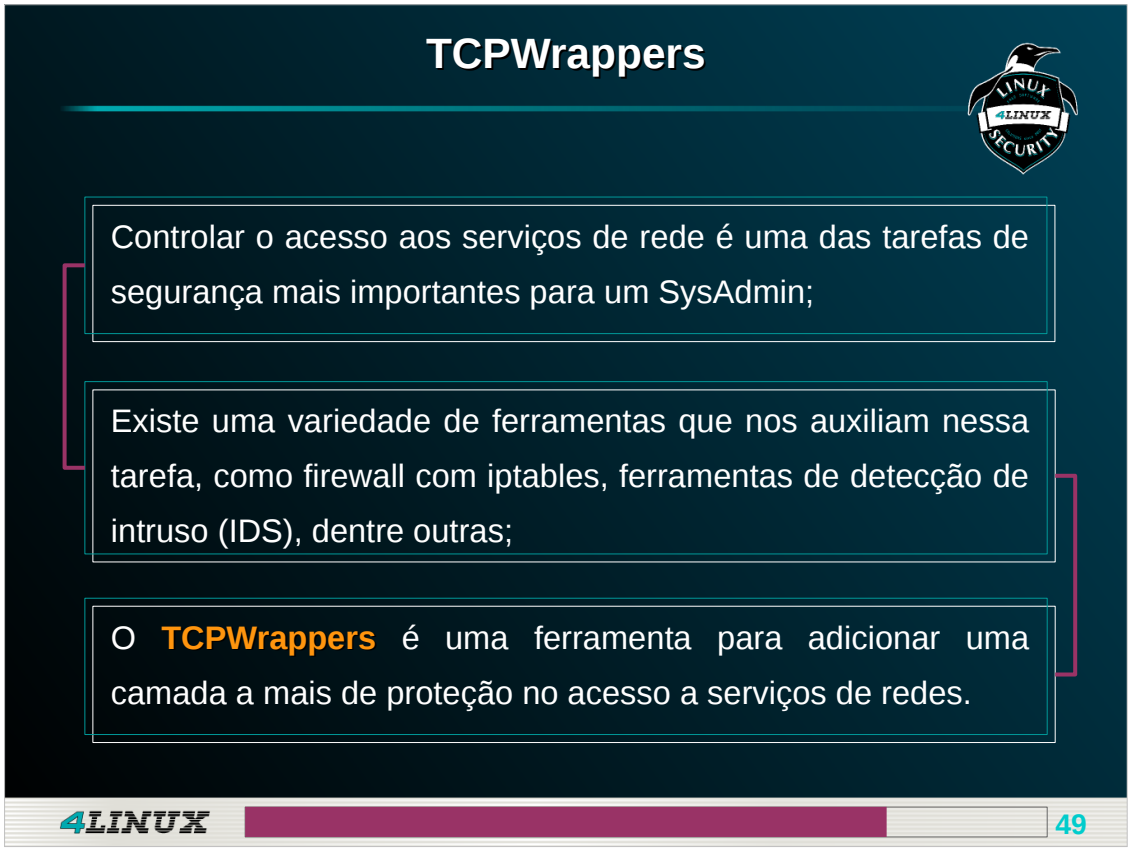
Fundamentação

Os “TCP Wrappers” são usados para aplicar regras de acesso a diversos serviços em seu servidor, podendo permitir ou negar conexões a eles.

Eles são controlados por dois arquivos: “/etc/hosts.allow” - configuração de acessos permitidos para determinados IPs e “/etc/hosts.deny” - configuração de acessos negados para determinados Ips. TCP - Sigla para "Transmission Control Protocol".


TCPWrappers
User Layer
Anti-vírus
Senhas Fortes
S.O Atualizado
Transport Layer
Protocolos Criptografados
HTTPS, SSL
Access Layer
ACL
Firewalls
Autenticação Criptografada
TCPWrappers
Network Layer
VPNs
Firewalls
IDS
Camadas de Segurança
4LINUX
48

Anotações:


TCPWrappers
Controlar o acesso aos serviços de rede é uma das tarefas de segurança mais importantes para um SysAdmin;
Existe uma variedade de ferramentas que nos auxiliam nessa tarefa, como firewall com iptables, ferramentas de detecção de intruso (IDS), dentre outras;
O TCPWrappers é uma ferramenta para adicionar uma camada a mais de proteção no acesso a serviços de redes.
4LINUX
49

Anotações:

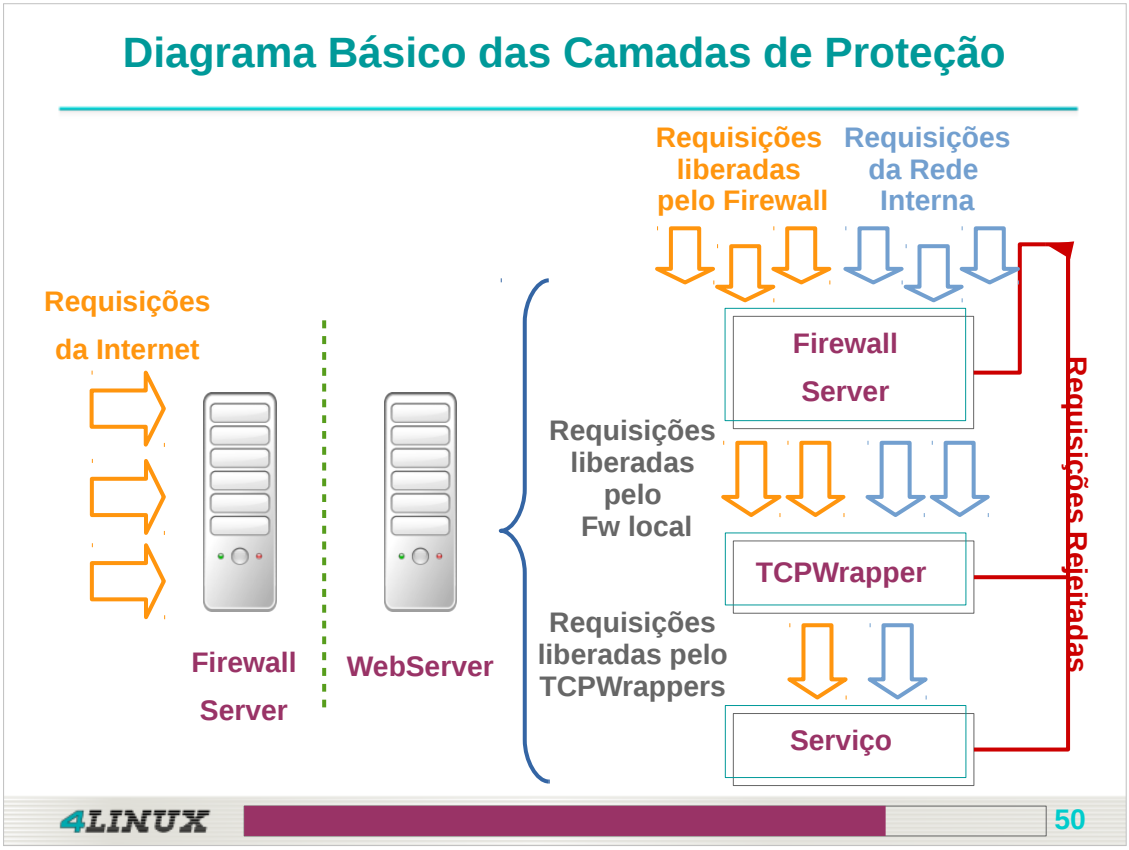


Diagrama Básico das Camadas de Proteção
Requisições liberadas pelo Firewall
Requisições da Rede Interna
Requisições da Internet
Firewall Server
WebServer
Requisições liberadas pelo Fw local
Firewall Server
TCPWrapper
Requisições liberadas pelo TCPWrappers
Serviço
Requisições Rejeitadas
4LINUX
50

Anotações:

TCPWrappers

- O **TCPWrappers** se resume basicamente em 2 arquivos de controle de acesso:

/etc/hosts.allow → Regras de liberação de acesso;
/etc/hosts.deny → Regras de bloqueio de acesso;

- O arquivo hosts.allow tem prioridade, pois é o primeiro a ser lido;
- Se um cliente é liberado para se conectar, o **TCPWrappers** libera o controle da conexão para o serviço solicitado e não participa mais da comunicação entre o cliente e o servidor.

Anotações:

TCPWrappers

Servidor: Webserver Interno

Suporte ao TCPWrappers:

```
1# which sshd
2# ldd /sbin/sshd
3# ldd /usr/sbin/sshd | grep wrap
   libwrap.so.0 =>
/lib/libwrap.so.0...
4# ldd /sbin/httpd | grep wrap
```

Nenhuma saída!

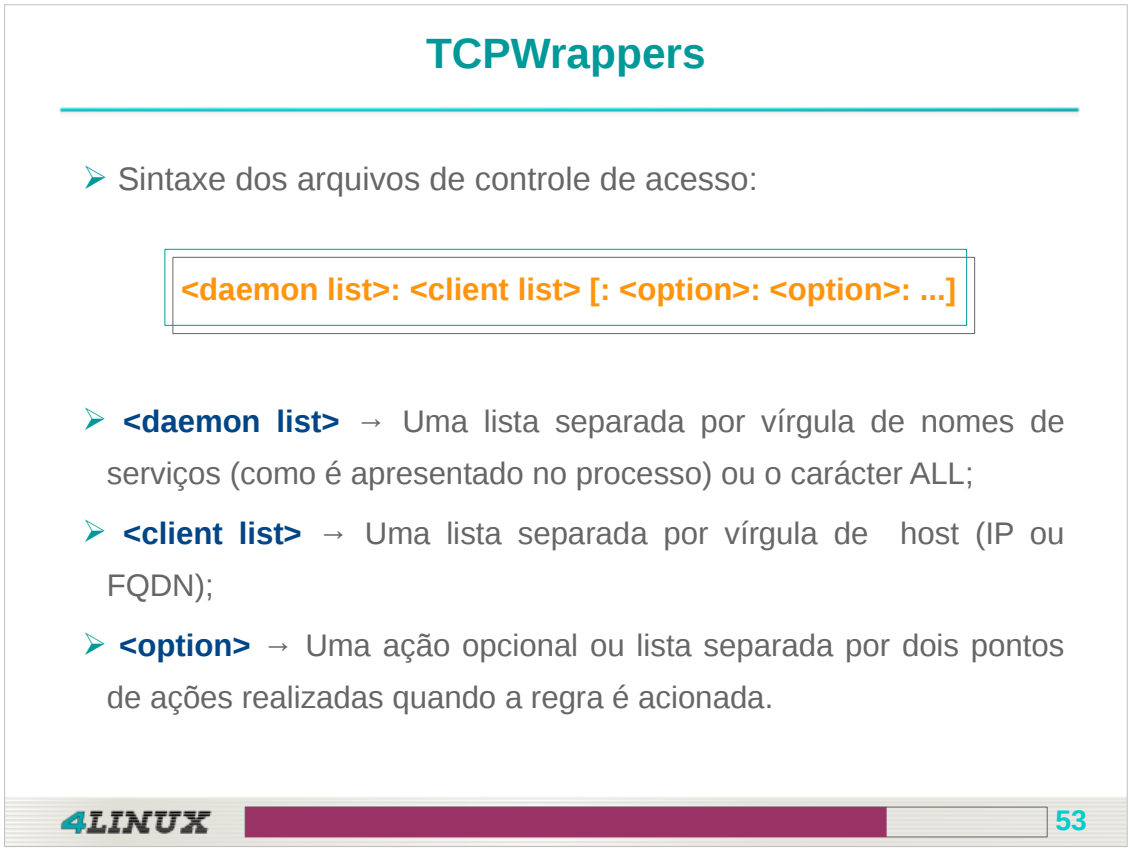
Fique atento aos serviços que possuem suporte o TCPWrappers. Repare que enquanto o SSH tem suporte, o Apache não tem.

Para que seja possível realizar controle de acesso pelo TCPWrapper primeiramente você precisa verificar se o serviço em específico tem **suporte a biblioteca libwrap**.

O comando **ldd** é usado para listar todas as bibliotecas de um determinado comando.

No capítulo sobre bibliotecas você verá mais detalhes sobre o ldd.

Anotações:



Entendendo o TCPWrappers

Existem dezenas de possibilidades de configuração para o tcp_wrappers e você pode estudá-las em extensão através das páginas de manual “hosts_access” e “hosts_options”.

Curingas podem ser utilizados tanto na lista de daemons quanto na lista de clientes. Entre os existentes, pode-se destacar os seguintes:

ALL - Significa todos os serviços ou todos os clientes, dependendo apenas do campo em que se encontra.

LOCAL - Este curinga casa com qualquer nome de máquina que não contenha um caractere ponto “.”, isto é, uma máquina local.

PARANOID - Casa com qualquer nome de máquina que não case com seu endereço. Geralmente ocorre quando um servidor DNS está mal configurado ou alguma máquina está tentando se passar por outra.

TCPWrappers

Servidor: Webserver Interno

Bloqueando o acesso de toda rede ao servidor da Dexter Courier:

```
# vim /etc/hosts.deny

   sshd: 192.168.200.   (Representa toda rede 192.168.200.0)
```

Tente acessar o servidor WebServerInterno pela máquina Linux Interna.

Anotações:

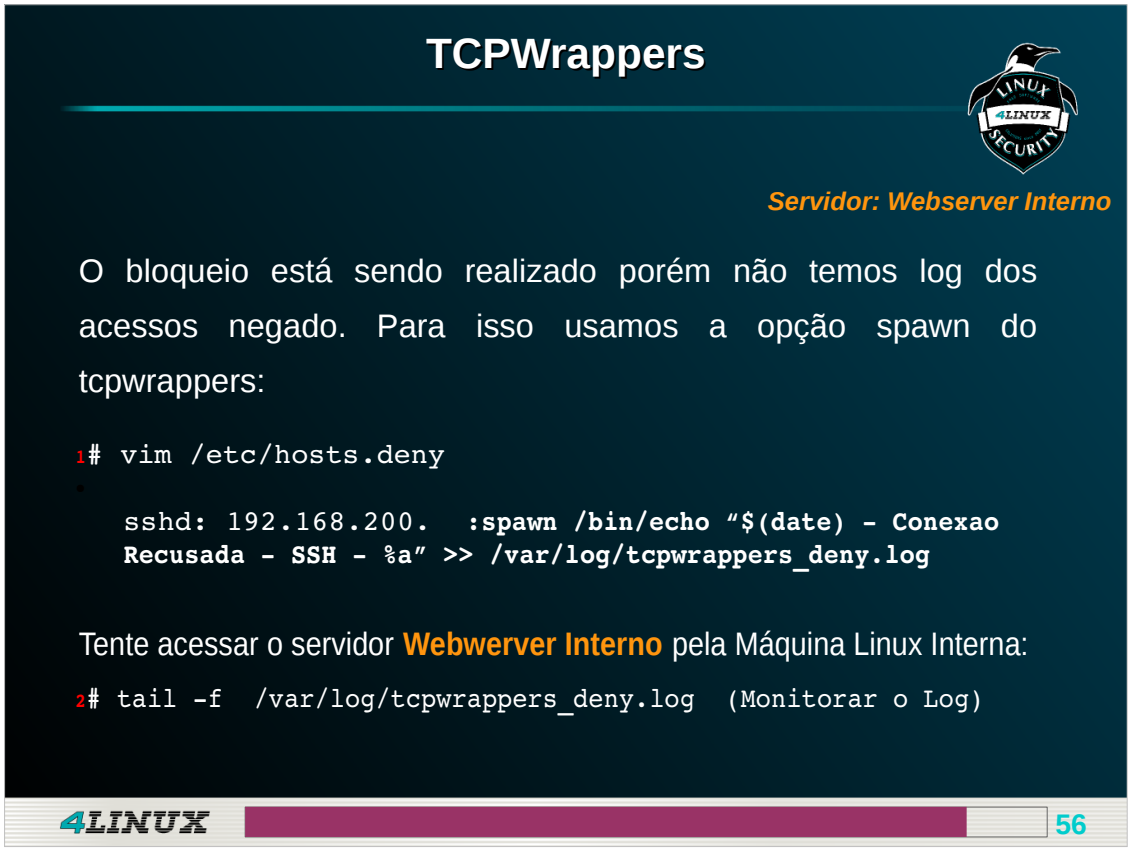
TCPWrappers

Servidor: Máquina Linux Interna

- Verificando o Acesso:
  - Abra um Terminal na Máquina Linux Interna: **Aplicativos > Acessórios > Terminal**

```
# ssh root@192.168.200.20
ssh_exchange_identification: Connection closed by remote host
```


Anotações:



Criando log no TCPWrappers

Muitas vezes precisamos ter um melhor controle sobre quem esta acessando nosso servidor e onde estes acessos se originam.

A opção spawn no serviço TCPWrapper irá realizar uma auditoria referente as ações do deamon especificado, direcionando toda a saída para um arquivo de log, especificado na própria sintaxe da opção.

Vale lembrar que o arquivo de log apenas será criado caso caia na regra especificada, no caso dos arquivos hosts.allow e hosts.deny.

TCPWrappers

Servidor: Webserver Interno

- Toda rede está bloqueada;
- Agora, libere o acesso apenas para o IP da sua máquina **Linux Interna**.

Dica:

O bloqueio da rede foi realizada no arquivo **/etc/hosts.deny**. Agora temos que liberar o acesso apenas para um determinado IP.

Anotações:

TCPWrappers

Servidor: Webserver Interno

Toda a rede está bloqueada. Agora, libere o acesso apenas para o IP da sua máquina **Linux Interna**:

```
1# vim /etc/hosts.allow
   sshd: 192.168.200.10
```

Tente acessar o servidor **Webserver Interno** pela máquina Linux Interna.

Anotações:

Pergunta LPI

Qual arquivo do sistema contém a lista de hosts que não podem acessar os serviços da máquina?

a. `/etc/hosts/denial`

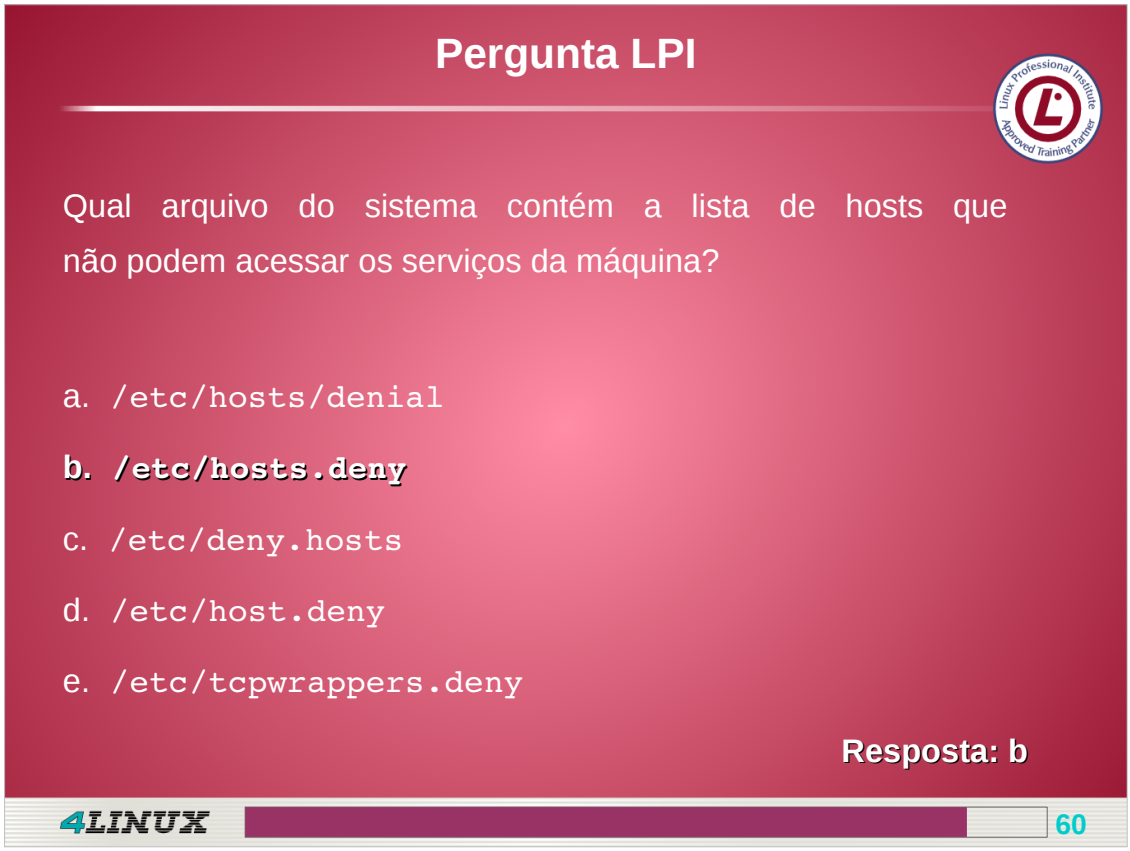
b. `/etc/hosts.deny`

c. `/etc/deny.hosts`

d. `/etc/host.deny`

e. `/etc/tcpwrappers.deny`

Anotações:



Alternativa B: RESPOSTA CORRETA!

Conforme testado em aula o arquivo hosts.deny pode ser usado para restringir o acesso dos usuários a serviços.

NOTA:

Lembre-se: O arquivo hosts.allow tem prioridade sobre o arquivo hosts.deny, pois é o primeiro a ser lido; É muito comum encontrarmos perguntas LPI que cobrem este conceito.

Próximos passos

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- Executar as tarefas do **Practice Lab**;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- Resolver o **Desafio OpenCloud Lab** e postar o resultado no Fórum Temático;
- Responder as questões do **Teste de Conhecimento** sobre o conteúdo visto em aula.

Mãos à obra!