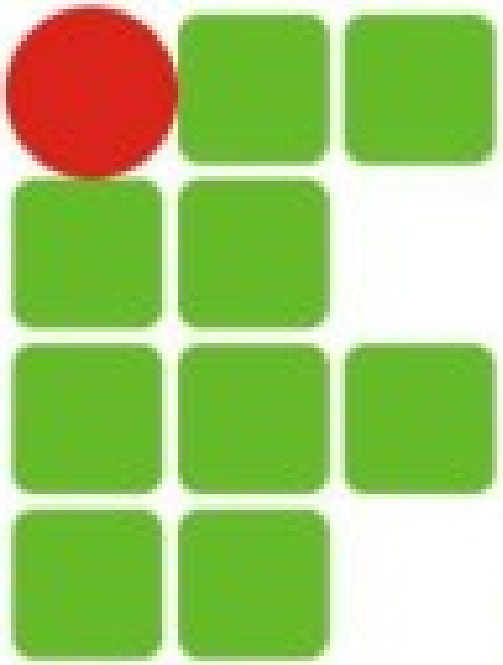
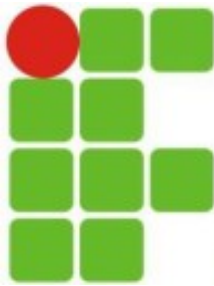


SSH
Secure Shell

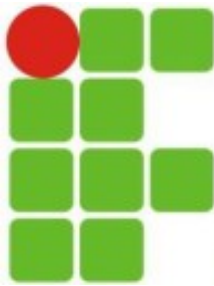
Galileu Batista de Sousa
Galileu.batista -at +ifrn -edu +br

Agenda



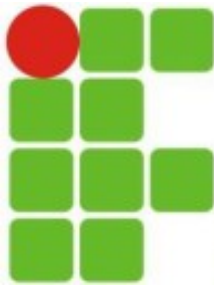
- O que é ?
- Origem
- Como funciona
- Configuração
- Tunelamento

O que é SSH



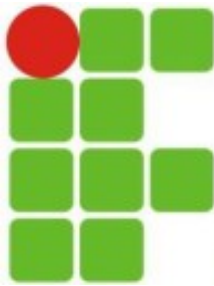
- **SSH** – Um protocolo/aplicação para executar comandos em um sistema remoto.
- Na verdade:
  - Estabelecimento de canal seguro entre hosts
  - Provê confidencialidade, integridade e autenticidade
  - Usos para:
    - Tunelamento
    - Nome do *host*

Por que SSH?



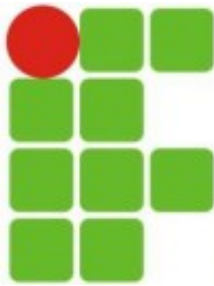
- Resposta a suscetibilidade a *sniffing*.
  - telnet, rsh, rlogin,
- Versão 1 - criada em 1995
  - Foi se tornando proprietário
- Versão 2 – projetada em 1999
  - Incompatível com a versão 1
  - Várias melhorias implementadas
  - Openssh é a implementação mais popular

Implementações correntes



- OpenSSH
  - Cliente e servidor
  - Muito popular nos sistemas UNIX
- PuTTY
  - Apenas cliente
  - Predominante no mundo Windows
- Outras:
  - MindTerm – SSH Java client
  - Mobile Shell (mosh)

Protocolos de suporte

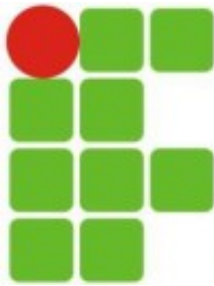


- Transport Layer Protocol
  - Provê confidencialidade e integridade
  - Usa o algoritmo Diffie-Hellman para troca de chaves
  - Muitos algoritmos simétricos implementados:
    - 3DES, Blowfish, twofish, AES, etc
  - Integridade via HMAC (MD5, SHA)
- Autenticação:
  - Servidor para para cliente – Chave pública
  - Cliente para servidor – senha ou chave pública

Em resumo

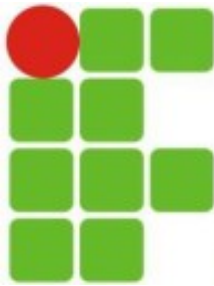
Application Layer	**ssh-connection** Session multiplexing, X11 and port forwarding, remote command execution, SOCKS proxy, etc.
	ssh-userauth User authentication using public key, password, host based, etc.
	ssh-transport Initial key exchange and server authentication, setup encryption
Transport Layer	**TCP**
Internet Layer	**IP**
Network Access Layer	**Ethernet**

Operação básica



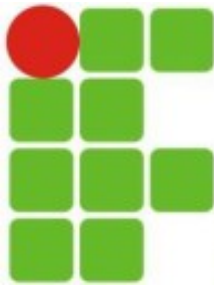
- Servidor:
  - Escuta na porta **22**, por padrão
- Cliente:
  - **`ssh usuario@host`**
  - **`ssh -l usuario -p 22 host`**
  - **`ssh meu-usuario`**
    - Um arquivo de configuração diz os parâmetros de meu-usuario.

SSH - configuração



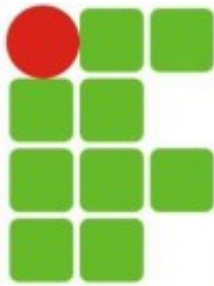
- Instalação dos pacotes:
  - `$ sudo apt-get install openssh-server`
  - `$ sudo apt-get install openssh-client`
- Arquivos:
  - `/etc/ssh/sshd_config`
    - Arquivo de configuração geral do servidor
  - `/etc/ssh/ssh_config` **e** `~/.ssh/config`
    - A configuração para uso do cliente
    - A configuração do *home* tem precedência

Configuração do Servidor - /etc/ssh/sshd_config



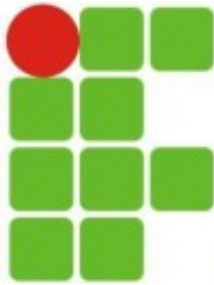
- **`Port 22`**
  - A porta em que o servidor escutará
- **`ListenAddress 0.0.0.0`**
  - Os interfaces (endereços) em que o servidor escutará
- **`Protocol 2`**
  - A versão do protocolo a utilizar (2 é preferível)
- **`PermitRootLogin prohibit-password`**
  - Não se se logar como root por meio de senha
- **`IgnoreRhosts yes`**
  - Ignora os arquivos de equivalência de hosts

Configuração do Servidor - /etc/ssh/sshd_config



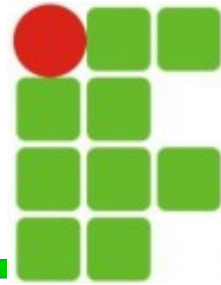
- **`UsePrivilegeSeparation yes`**
  - Cria um processo de baixa prioridade para tratar conexão
- **`TCPKeepAlive yes`**
  - Permite ao servidor detetar perda de conexão a clientes
- **`LoginGraceTime 120`**
  - O tempo que o servidor permite (120s) para que o cliente faça o login com sucesso.
  - *Default* é zero – sem limite
- **`ChrootDirectory /tmp`**
  - Muda o dietório raiz para /tmp, após a conexão

Operações sobre o servidor



- Iniciar:
  - `sudo systemctl start ssh`
- Parar:
  - `sudo systemctl stop ssh`
- Status / log:
  - `sudo systemctl status ssh`
  - `sudo journalctl -xe`

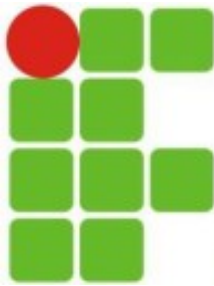
Configurações do cliente - ~usuario/.ssh/config



- `Host asa-vm`
  - Inicia especificação de parâmetros de conexão a asa-vm
- `Hostname 10.25.1.142`
  - IP do servidor de SSH associado a asa-vm
- `Port 22`
  - Porta do servidor de SSH associado a asa-vm
- `User gbat`
  - Usuário de conexão associado a asa-vm

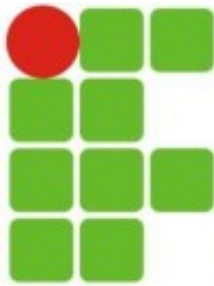
`ssh asa-vm`

Outros arquivos importantes



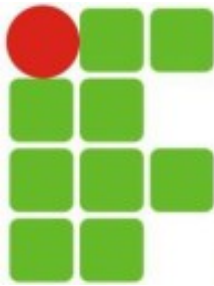
- Servidor:
  - **`~usuario-serv/.ssh/authorized_keys`**
    - Contém a chave pública de um **`usuario-cli`**
    - Mecanismo de *bypass* do processo de *login*
    - Contém uma entrada para cada chave pública de um cliente que pode "logar sem senha"
  - **`/etc/ssh/ssh_host_rsa_key.pub`**
  - **`/etc/ssh/ssh_host_rsa_key`**
    - Chaves RSA públicas e privadas do servidor
    - Geradas quando da instalação do servidor

Geração de chaves para cliente



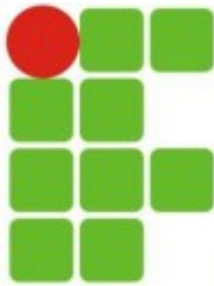
- O cliente pode ser seu par de chaves com:
  - **`ssh-keygen`**
  - Dois arquivos são gerados:
    - **`~usuario/.ssh/id_rsa`** – Chave privada do usuário
    - **`~usuario/.ssh/id_rsa.pub`** – Chave pública
- Muita atenção às permissões desses arquivos:
  - **Chave privada – 600**
  - Chave pública – 644

Confiança entre usuários



- O usuário pode copiar a sua chave pública para um usuário do servidor
  - `ssh-copy-id -i id_rsa.pub usuario-serv@serv`
  - `cat id_rsa.pub | ssh usuario-serv@serv "cat - >> ~/.ssh/authorized_keys"`
  - Ou simplesmente o usuário do servidor adiciona manualmente a chave do cliente
- Agora o usuário autorizado, loga sem senha

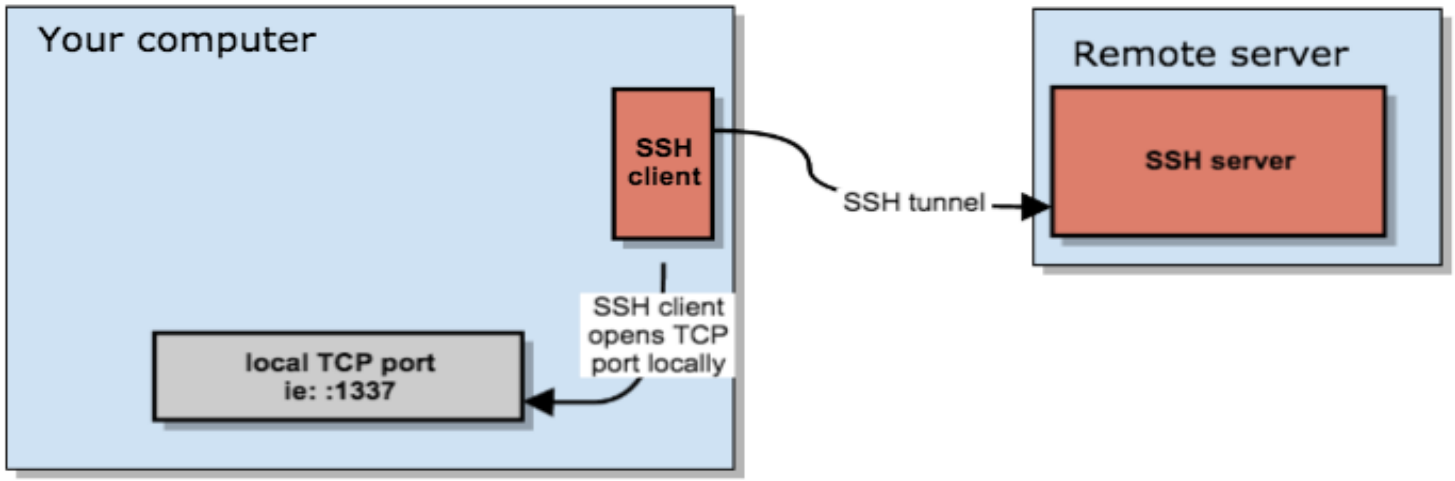
Outros arquivos importantes



- Cliente:
  - `~usuario-cli/.ssh/known_hosts`
    - Contém informações de identidade de servidores em que o usuário da máquina cliente confia
- A primeira vez que se loga em um servidor:
  - Tem a oportunidade de aceitar ou não as crendenciais do servidor
  - É possível aceitar por padrão (PERIGOSO?)
    - `ssh -o StrictHostKeyChecking=no user@host`

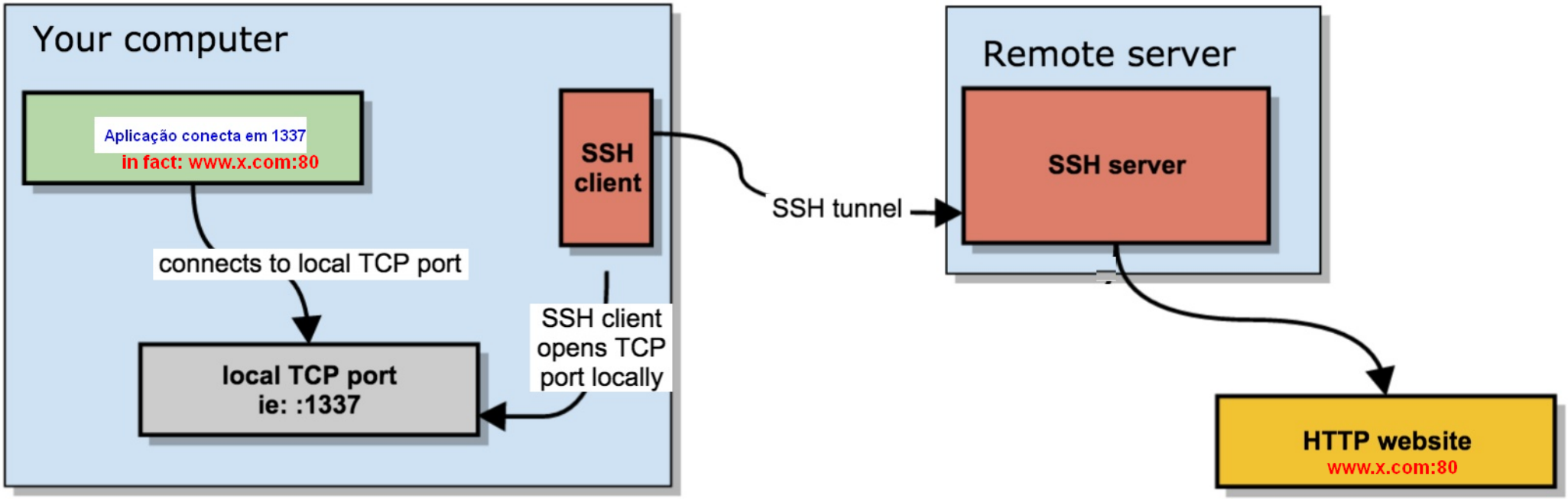
Tunelamento usando SSH

- O SSH pode ser usando como uma canal para atingir servidores, além do próprio
- O cliente tem mais funções:
  - Conectar ao servidor e também escutar uma porta local
  - Redirecionar o tráfego de conexões nessa porta local para o servidor SSH



Tunelamento usando SSH

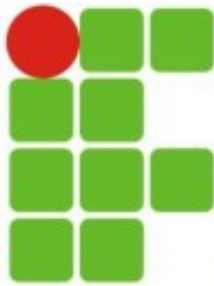
- Quando uma conexão chega ao cliente
  - Ele redireciona os dados da conexão ao servidor
  - O servidor abre uma conexão com um outro servidor




```
ssh -Nf -L1337:www.x.com:80 remote
```

background

Tunelamento usando SSH



- É possível fazer o tunelamento reverso
  - O servidor recebe a conexão do cliente
  - Passa a escutar outra porta
  - Qualquer conexão nesse segunda porta é redirecionada para um novo host, por meio do cliente

```
ssh -fN -R1234:www.x.com:80 remote
```