SECURE SHELL

OUTLINE

- What is SSH ?
- History
- Functions of Secure Shell ?
- Elements of Secure Shell?
- Architecture
- How Secure Shell works

OUTLINE

- Security Benefits ?
- Why should we use SSH ?
- Conclusion
- References

What is SSH?

- SSH is a protocol for secure remote access to a machine over untrusted networks.
- SSH is a replacement for telnet, rsh, rlogin and can replace ftp.
- Uses Encryption.
- SSH is not a shell like Unix Bourne shell and C shell (wildcard expansion and command interpreter)

Features

- Transmission is secure.
- Transmission can be compressed.
- No login password required

What's wrong with telnet?

- Sends all data in clear text.
- Host between sender and receiver can see what the traffic is.

Why should we encrypt data ?

- Use the same password in more than one place.
- Do you want someone else to read your mail?

History of SSH?

- Created by Tatu Ylönen in July 1995, a student of Helsinki University of Technology
- Free SSH1 version
- Founded SSH Communications Security, Ltd
- SSH 2 version
- Open SSH

Functions

- Secure Command Shell
- Port Forwarding
- Secure file transfer.

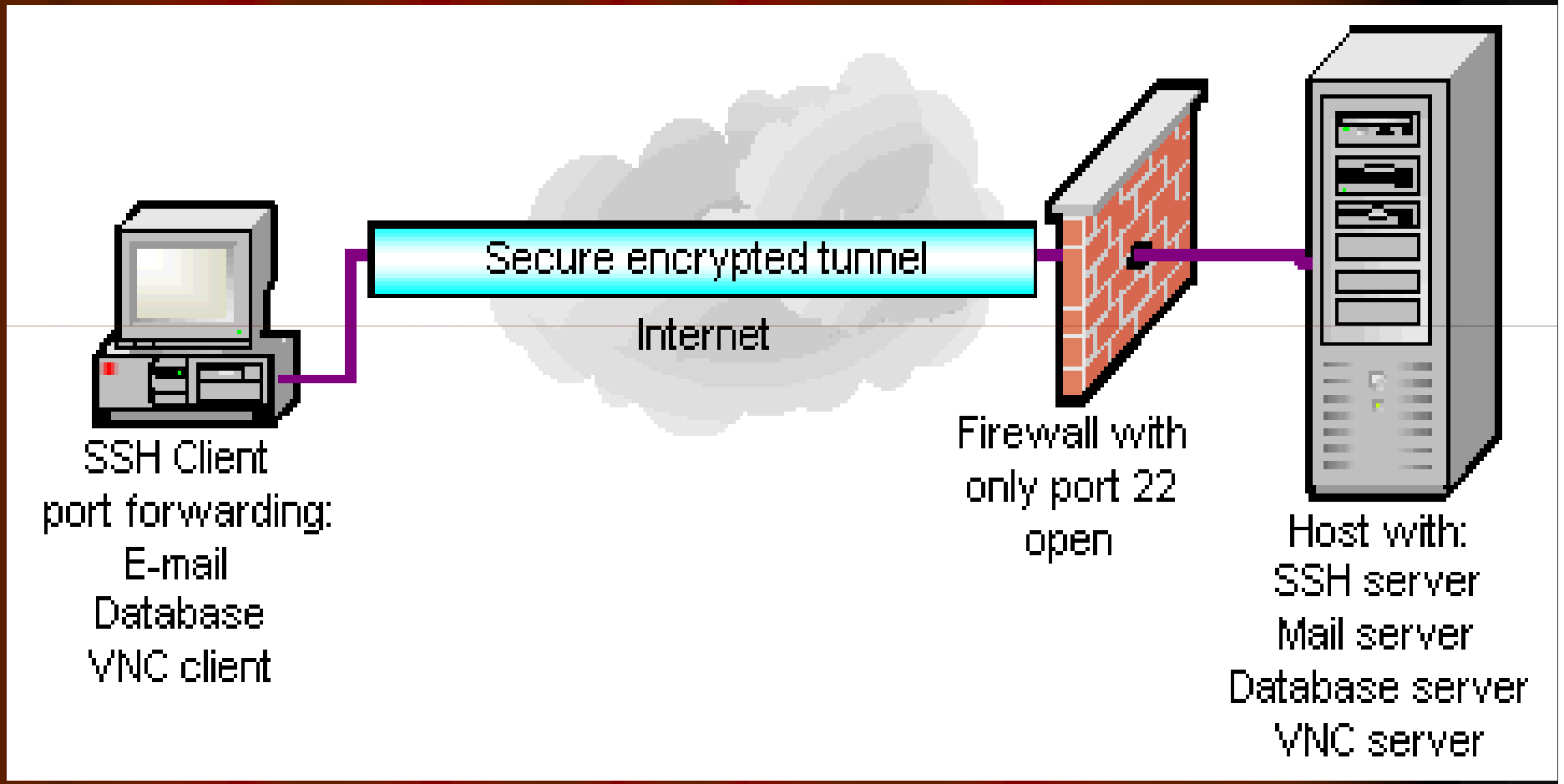
Secure Command Shell

- Allow you to edit files.
- View the contents of directories.
- Custom based applications.
- Create user accounts.
- Change permissions.
- Anything can be done from command prompt can be done remotely and securely.

Port Forwarding

- Powerful Tool.
- provide security to TCP/IP applications including e-mail, sales and customer contact databases, and in-house applications.
- allows data from normally unsecured TCP/IP applications to be secured.

Port Forwarding



Secure File Transfer

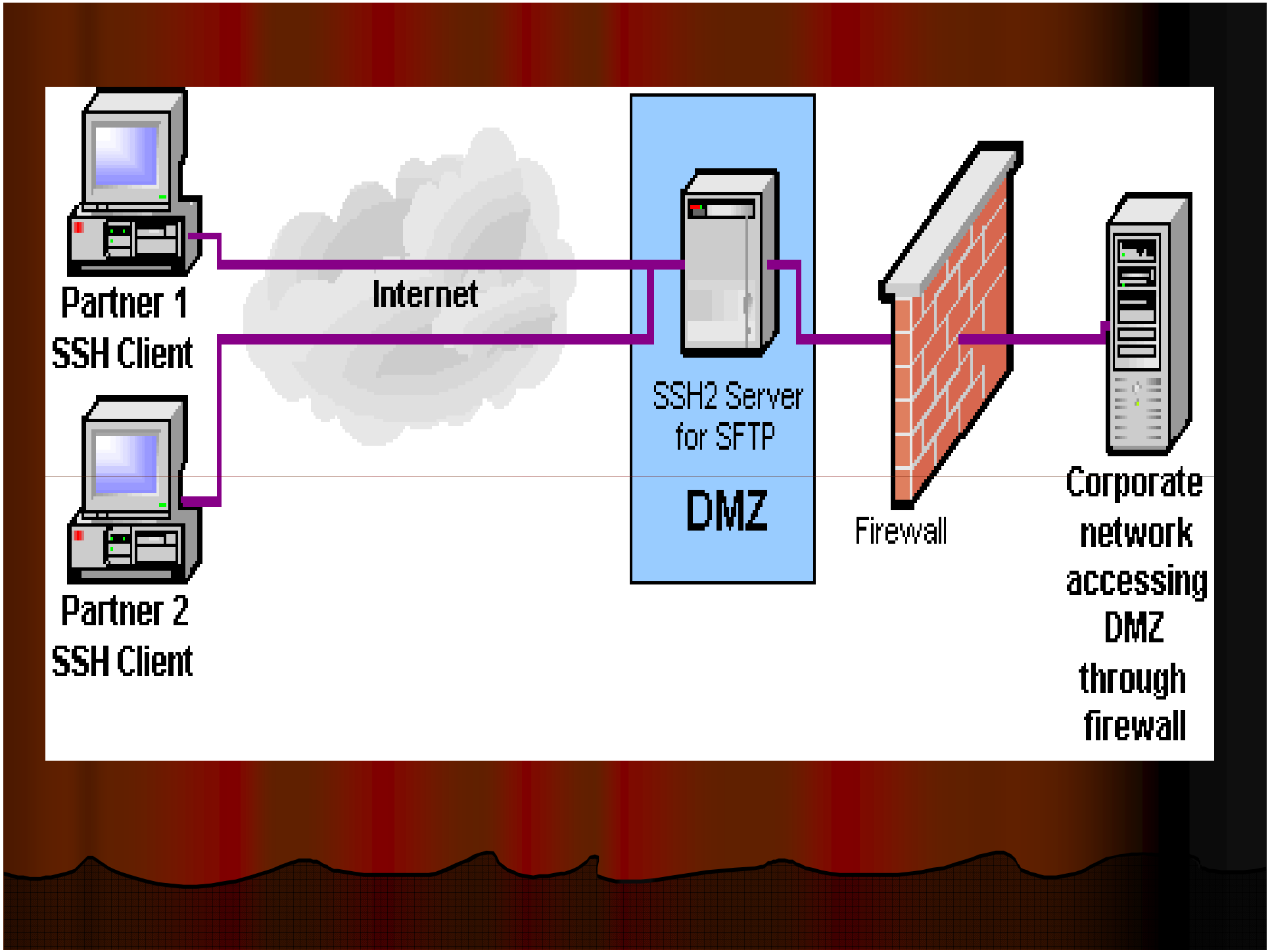
- Secure File Transfer Protocol (SFTP) is a subsystem of the Secure Shell protocol.
- Separate protocol layered over the Secure Shell protocol to handle file transfers.

SFTP

- SFTP encrypts both the username/password and the data being transferred.
- Uses the same port as the Secure Shell server, eliminating the need to open another port on the firewall or router.
- Using SFTP also avoids the network address translation (NAT) issues that can often be a problem with regular FTP.

SFTP

- An ideal use of SFTP is to fortify a server or servers outside the firewall or router accessible by remote users and/or partners (sometimes referred to as a secure extranet or DMZ).


Partner 1
SSH Client
Partner 2
SSH Client
Internet
SSH2 Server
for SFTP
DMZ
Firewall
Corporate
network
accessing
DMZ
through
firewall

Secure File Transfer Protocol

Secure extranet is one of the safest ways to make specific data available to customers, partners and remote employees without exposing other critical company information to the public network. Using SFTP on your secure extranet machines effectively restricts access to authorized users and encrypts usernames, passwords and files sent to or from them.

Components of Secure Shell

- SSHD Server: A program that allows incoming SSH connections to a machine, handling authentication, authorization.
- Clients: A program that connects to SSH servers and makes requests for service
- Session: An ongoing connection between a client and a server. It begins after the client successfully authenticates to a server and ends when the connection terminates.

SSH Architecture

- The user initiates an SSH connection. SSH attempts to connect to port 22 on the remote host.
- If successful, SSHD on the machine Remote forks off a child SSHD process. This process will handle the SSH connection between the two machines.

- The child SSHD now forks off the command received from the original SSH client.
- The SSHD child process now encrypts every messages that has to be send to the ssh client.

- The SSH client decrypts the information and sends it to the user application.

How Secure Shell Works ?

- When SSHD is started , it starts listening on port22 for a socket. When a socket get connected the secure shell daemon spawns a child process. Which in turn generates an host key e g. RSA. After key is generated the secure shell daemon is ready for the local client to connect to another secure shell daemon or waits for a connection from remote host.

Security Benefits

- User Authentication
- Host Authentication
- Data Encryption
- Data Integrity

User Authentication

- User Identity
- System verifies that access is only given to intended users and denied to anyone else.

Password Authentication

- Passwords, in combination with a username, are a popular way to tell another computer that you are who you claim to be.
- If the username and password given at authentication match the username and password stored on a remote system, you are authenticated and allowed access.

Public Key Authentication

- Most secure Method to authenticate using Secure Shell
- Public key authentication uses a pair of computer generated keys - one public and one private. Each key is usually between 1024 and 2048 bits in length

Public Key Authentication

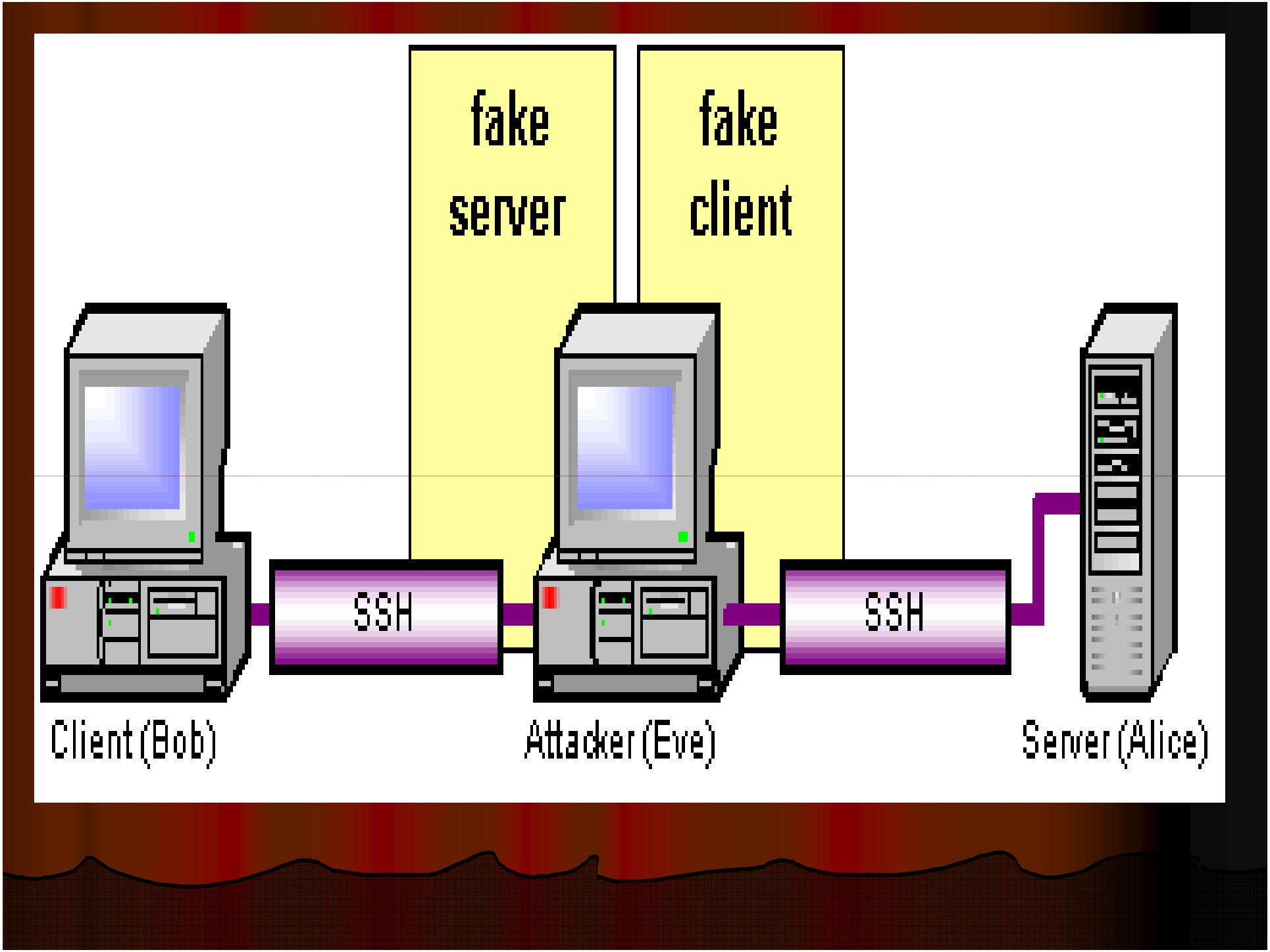
- To access an account on a Secure Shell server, a copy of the client's public key must be uploaded to the server. When the client connects to the server it proves that it has the secret, or private counterpart to the public key on that server, and access is granted.

Host Authentication

- A host key is used by a server to prove its identity to a client and by a client to verify a "known" host. Host keys are described as persistent (they are changed infrequently) and are asymmetric--much like the public/private key pairs discussed above in the Public key section. If a machine is running only one SSH server, a single host key serves to identify both the machine and the server. If a machine is running multiple SSH servers, it may either have multiple host keys or use a single key for multiple servers. Host authentication guards against the Man-in-the-Middle attack.

Host Authentication...

- To access an account on a Secure Shell server, a copy of the client's public key must be uploaded to the server. When the client connects to the server it proves that it has the secret, or private counterpart to the public key on that server, and access is granted.


fake
server
fake
client
SSH
SSH
Client (Bob)
Attacker (Eve)
Server (Alice)

Data Encryption

- Encryption, sometimes referred to as privacy, means that your data is protected from disclosure to a would-be attacker "sniffing" or eavesdropping on the wire. Ciphers are the mechanism by which Secure Shell encrypts and decrypts data being sent over the wire.

Data Encryption...

- When a client establishes a connection with a Secure Shell server, they must agree which cipher they will use to encrypt and decrypt data. The server generally presents a list of the ciphers it supports, and the client then selects the first cipher in its list that matches one in the server's list.

Data Integrity

- Data integrity guarantees that data sent from one end of a transaction arrives unaltered at the other end. Even with Secure Shell encryption, the data being sent over the network could still be vulnerable to someone inserting unwanted data into the data stream (See Insertion and replay attacks for more details). Secure Shell version 2 (SSH2) uses Message Authentication Code (MAC) algorithms to greatly improve upon the original Secure Shell's (SSH1) simple 32-bit CRC data integrity checking method.

Protect Against

- IPS Spoofing
- DNS Spoofing
- IP Source Routing

IPS Spoofing

- IP spoofing is a technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host.

IP Source Routing

- where a host can pretend that an IP packet comes from another, trusted host.

DNS Spoofing

- DNS spoofing is a term used when a DNS server accepts and uses incorrect information from a host that has no authority giving that information. DNS spoofing is in fact malicious cache poisoning where forged data is placed in the cache of the name servers. Spoofing attacks can cause serious security problems for DNS servers vulnerable to such attacks, for example causing users to be directed to wrong Internet sites or e-mail being routed to non-authorized mail servers

Reasons to use SSH?

- Designed to be a secure replacement for rsh, rlogin, rcp, rdist, and telnet.
- Strong authentication. Closes several security holes (e.g., IP, routing, and DNS spoofing).
- Improved privacy. All communications are automatically and transparently encrypted.

Reasons to use SSH

- Arbitrary TCP/IP ports can be redirected through the encrypted channel in both directions
- The software can be installed and used (with restricted functionality) even without root privileges.
- Optional compression of all data with gzip (including forwarded X11 and TCP/IP port data), which may result in significant speedups on slow connections.

Conclusion

- SSH it is possible to create a secure communication channel between the server and the client.
- This channel can be used for different purposes, not necessarily for launching a remote terminal session but also for sending any data using the forwarding feature.
- SSH supports a variety of authentication methods, and new options may be added if required.
- Both the client and the server can authenticate each other to enhance security against different kinds of attacks.

References

- http://yakko.cs.wmich.edu/presentations/20021107-ssh/slides/img7.html
- http://www.vandyke.com/solutions/ssh_overview/ssh_overview_functionalit
- http://michaelsteel.tripod.com/cgi-bin/
- http://www.faqs.org/faqs/computer-security/ssh-faq/