

Secure Shell (SSH)

M4L5

What is SSH?

- “SSH is a protocol for secure remote login and other secure network services over an insecure network.” – RFC 4251
- Secure channel between two computers
 - Provides data confidentiality and integrity
- Many uses other than remote shell

History

- SSH-1 designed in 1995 by Tatu Ylönen
 - In response to a password-sniffing attack
 - Replacement for rlogin, telnet, and rsh
 - Released as freeware in July 1995
 - ~20,000 users in 50 countries by the end of the year
- Ylönen founded SSH Communications Security in December 1995
 - Code became increasingly more proprietary

History (continued)

- SSH-2 designed in 1996
 - Incompatible with SSH-1
 - Security and feature improvements
- Open source implementations (OSSH and OpenSSH) created in 1999
 - OSSH is now obsolete
 - OpenSSH is the most popular SSH implementation as of 2005

Current Implementations (2007)

- OpenSSH – common on UNIX systems
- SSH Tectia – commercial implementation
- PuTTY – client only, Windows
- MindTerm – client only, Java applet



Layering of SSH Protocols

- Transport Layer Protocol
 - Provides server authentication, confidentiality, and integrity
- User Authentication Protocol
 - Authenticates the client-side user to the server
- Connection Protocol
 - Multiplexes the tunnel into logical channels
- New protocols can coexist with the existing ones

Transport Layer Protocol

- Public-key host authentication
 - Lets the client know the correct server is on the other end
 - DSS or RSA, raw or through OpenPGP
- Strong symmetric encryption
 - Uses Diffie-Hellman algorithm for secure key exchange
 - Many ciphers are supported: 3des, blowfish, twofish, aes, etc., most with multiple key sizes
 - New keys generated every 1 GB or 1 hour
- Data integrity via MACs (message authentication codes)
 - SHA-1 and MD5 are supported

User Authentication Protocol

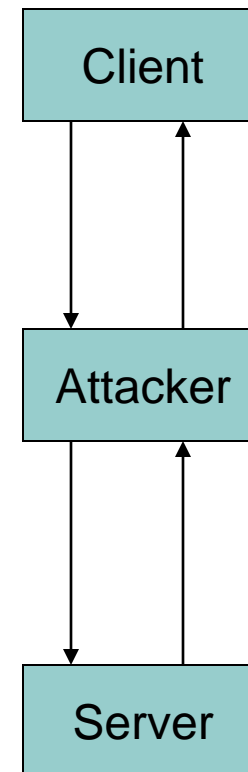
- Multiple authentication methods
 - public-key, password, host-based
 - Extensible
- Server tells client which methods can be used, client picks the most convenient
- Provides a single authenticated channel to the connection protocol

Connection Protocol

- Provides multiple channels:
 - interactive login sessions
 - remote execution of commands
 - forwarded X11 connections
 - forwarded TCP/IP connections
- All channels are multiplexed into a single encryption tunnel

Attacks on SSH

- Man-in-the-middle
 - Very easy if the client does not have the server's public key prior to connecting
 - Attacker masquerades between the client and server
- Denial of service
- Covert channels



System Configuration Files (OpenSSH)

- `/etc/ssh/`
 - `sshd_config` – SSH server configuration
 - `ssh_config` – SSH client configuration
 - `ssh_host*_key` – private host keys
 - `ssh_host*_key.pub` – public host keys
 - `ssh_known_hosts` – list of known public host keys

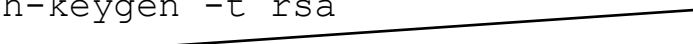
User Configuration Files (OpenSSH)

- ~/.ssh/
 - id_* - private authentication keys
 - id_*.pub – public authentication keys
 - known_hosts – list of known public host keys
 - authorized_keys – list of allowed public authentication keys

Public-Key Authentication Howto

```
$ ssh-keygen -t rsa
```

Accept the defaults and
leave the passphrase blank




```
...
```

```
$ cat ~/.ssh/id_rsa.pub | ssh <remote-host> 'cat - >> ~/.ssh/authorized_keys'
```

```
...
```

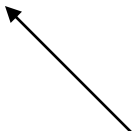
```
$ ssh <remote-host>
```

Enter your password
one last time



```
...
```

Enjoy not having to enter
a password



References and Resources

- RFC 4250-4254
- SSH: The Secure Shell – *The Definitive Guide*
 - <http://www.snailbook.com/index.html>
- http://en.wikipedia.org/wiki/Secure_Shell
- http://www.cs.clemson.edu/~duckwos/ssh_lab/