SSH : The Secure Shell

Introduction

- Widely used utilities to login through a network in Unix environment are telnet and rlogin
- Problem - user's login name and password transmitted as clear text
- Data transmission after login - also in clear text !

“SSH, the Secure Shell is a powerful, software-based approach to network security that provides a secure channel for data transmission through a network”

History

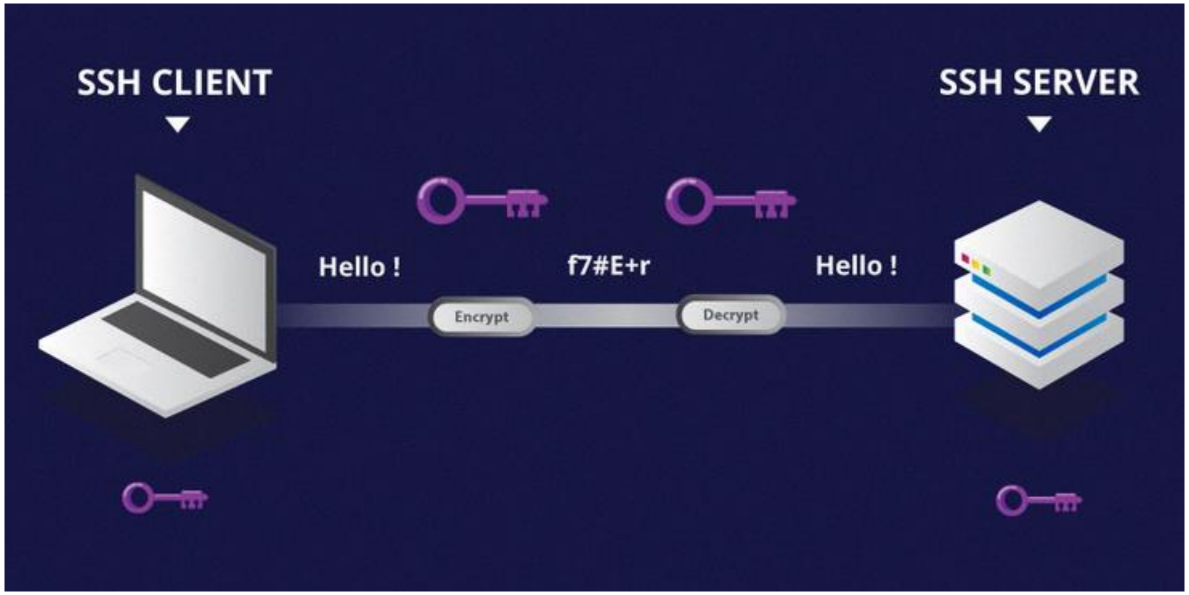
- Developed by Tatu Ylonen , Helsinki University of Technology, Finland in 1995 to prevent network attack against University network
- Published the protocol SSH-1 as an IETF (Internet Engineering Task Force) draft
- Founded SSH communications security Ltd., in late 1995 (http://www.ssh.com)
- Later released SSH-2

What is Secure Shell ?

- Powerful, convenient approach to protecting communications on a computer network
- Provides a secure channel for data transmission
- Provides a secure pipe to open up a command interpreter

What is Secure Shell ? (Contd..)

- Supports secure remote logins, secure remote command execution, secure file transfers
- Has a client server architecture – SSH server program and client program



Features

- Privacy : via strong end-to-end encryption- DES, IDEA, Blowfish
- Integrity : via 32 bit Cyclic Redundancy Check (CRC-32)
- Authentication : server via server's host key, client usually via password or public key
- Authorization : controlled at a server wide level or per account basis
- Forwarding : encapsulating another TCP based service such as Telnet within an SSH session

Security Mechanism

Establishing the Secure Connection

- The client initiates the connection by sending a request to the TCP port of the SSH server
- Server reveals it's SSH protocol version to the client
- If the client and server decide their versions are compatible, the connection proceeds

Establishing the Secure Connection (Contd..)

- SSH server sends the following to the client - host key, the server key, a list of supported encryption, compression and authentication methods, and a sequence of eight random bytes
- Client checks identity of server by using the host key against known hosts database
- Client generates a session key and double encrypts it using the host key & server key
- Client sends encrypted session key along with check bytes and acceptable algorithm

Authentication

- Server then decrypts the encrypted session key it received
- Server sends a confirmation encrypted with this session key
- Client receives confirmation, confirms server authentication
- Client Authentication usually either by Password Authentication or Public key Authentication

Authentication (Contd..)

- Server confirms client authorization
- Generates a 256 bit random challenge, encrypts it with clients public key, and sends to client
- Client decrypts challenge, generates a hash value with a session identifier (commonly generated random string at beginning of session), and sends to server
- Server generates hash, if both match, session is authenticated

SSH2 vs. SSH1

- SSH2 has separate transport, authentication, and connection protocols.SSH1 has one monolithic protocol
- SSH2 has strong cryptographic integrity check using MAC, SSH1 has weak checking using CRC-32
- SSH2 supports any number of session channels per connection (including none),SSH1 exactly one
- Servers running SSH-2 can also run SSH-1 to take care of clients running SSH-1
- SSH2 allows more than one form of authentication per session, SSH1 allows only one.