Protocollo SSH

Introduzione

SSH (o *S*ecure *Sh*ell) è un protocollo che facilita i collegamenti sicuri tra due sistemi, usando un'architettura del tipo client/server permettendo agli utenti di registrarsi in sistemi host server, in modo remoto.

A differenza di altri protocolli remoti di comunicazione, come FTP o Telnet, SSH cripta la sessione di login, impedendo alle persone non autorizzate di ottenere le password in chiaro.

Introduzione

SSH è stato progettato per sostituire applicazioni precedenti, meno sicure utilizzate per l'accesso a sistemi remoti come **telnet** o **rsh**. Un programma chiamato scp sostituisce i programmi meno recenti per copiare i file tra host, quali **rcp**.

Poichè queste applicazioni non cifrano le password tra il client e il server, si consiglia di utilizzarle il meno possibile.

Se usate dei metodi sicuri per collegarvi ad altri sistemi remoti, correte meno rischi per la sicurezza del vostro sistema e del sistema a cui vi collegate.

La prima versione di SSH fu SSH1 ed è tuttora la più utilizzata.

SSH2

SSH2 migliora SSH1 e standardizza definitivamente il protocollo SSH.

E’ documentato nelle RFC 4250-4256.

E’ la versione piu’ sicura, ma con una licenza d’uso piu’ restrittiva e disponibile su una varieta’ di piattaforme piu’ limitata.

Caratteristiche

Il protocollo SSH fornisce le seguenti misure di protezione:

- Dopo una connessione iniziale, il client verifica che il collegamento avvenga con lo stesso server, al quale ci si è collegati precedentemente.
- Il client trasmette le proprie informazioni di autenticazione al server usando una codifica a 128 bit
- Tutti i dati inviati e ricevuti durante la sessione, vengono trasferiti utilizzando una codifica a 128 bit, in questo modo è estremamente complesso decodificare e leggere le trasmissioni.
- Il client può inoltrare le applicazioni X11 applicazioni dal server. Questa tecnica, chiamata *X11 forwarding*, fornisce una applicazione grafica sicura da usare attraverso la rete.

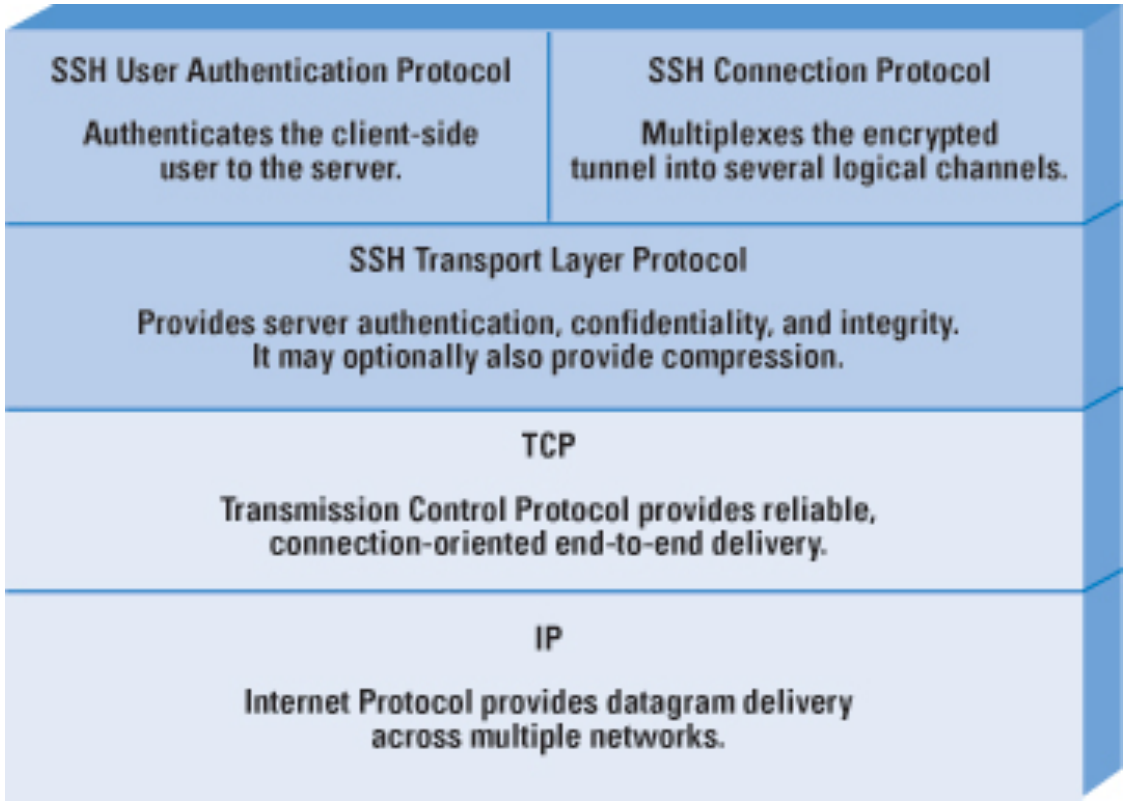
SSH + port forwarding

Poichè il protocollo SSH codifica tutto ciò che invia e riceve, esso può essere usato per cifrare protocolli che altrimenti non sarebbero sicuri.

Se usate la tecnica chiamata *port forwarding*, un server SSH può diventare un condotto per rendere sicuri protocolli non sicuri, come POP, aumentando la sicurezza dei dati e del sistema in generale.

Il protocollo

SSH è composto da tre protocolli che “girano” sopra la pila TCP/IP:



I tre protocolli

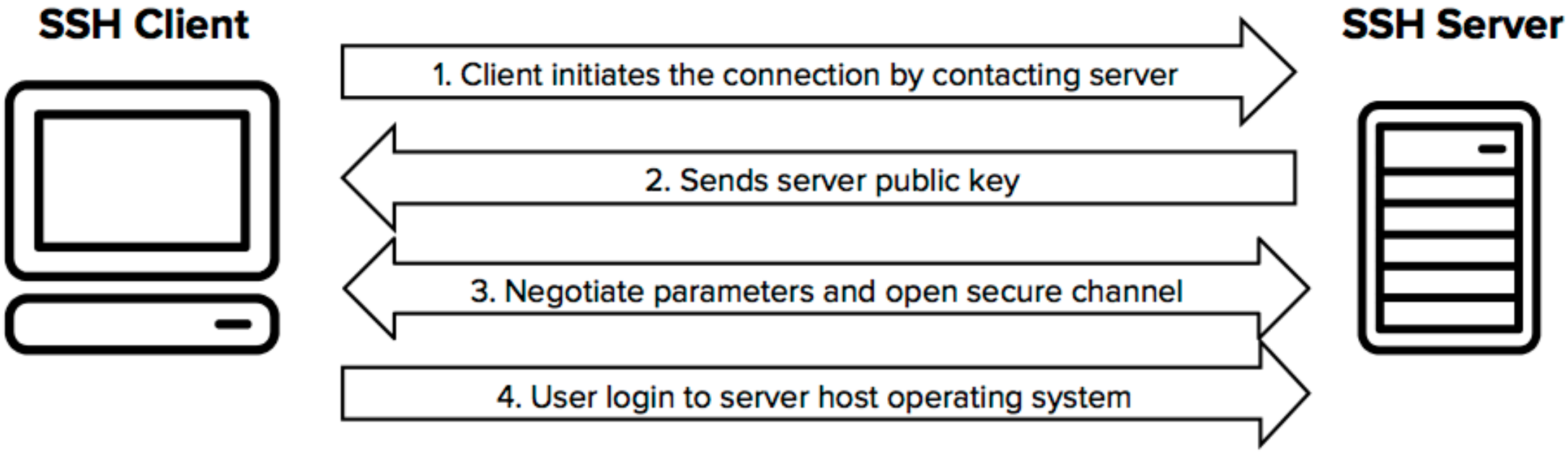
- *Transport Layer Protocol:* fornisce autenticazione lato server, riservatezza, integrità, “forward secrecy”, eventualmente compressione dati.
- *User Authentication Protocol:* autenticazione lato client
- *Connection Protocol:* “multiplexa” più canali di comunicazione logica su una singole connessione SSH sottostante.

Come funziona? (in sintesi….)

- Ogni host possiede una coppia di chiavi: pubblica e privata.
- Il client genera una sequenza di bit casuale e la comunica al server crittografandola con la chiave pubblica del server
- Tale sequenza di bit sarà utilizzata come chiave per l'algoritmo di crittografia a chiave simmetrica scelto per la sessione

=> Crittografia ibrida!

SSH al posto di Telnet



SSH port forwarding

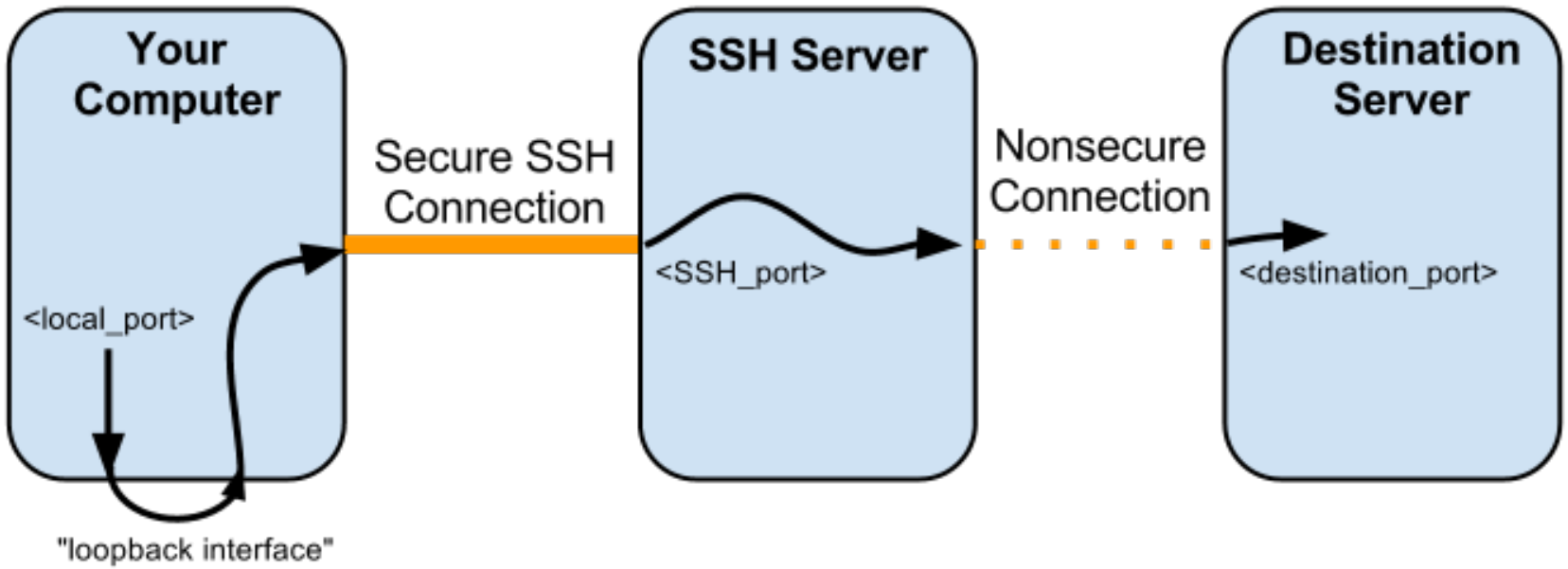
L’SSH port forwarding o tunnelling è un modo per inoltrare traffico TCP insicuro attraverso SSH per renderlo sicuro e non decifrabile. Utilizzando questa tecnica, i dati appartenenti a protocolli intrinsecamente insicuri come POP3, HTTP ecc. possono essere incapsulati in un tunnell SSH e resi sicuri, utilizzando la crittografia a chiave pubblica.

Esistono due tipi di port forwarding, local e remote, noti rispettivamente come outgoing tunnel (tunnel in uscita) e incoming tunnel (tunnel in entrata).

Local SSH port forwarding

Il local port forwarding inoltra il traffico indirizzato su una porta locale del client, attraverso un tunnel SSH, verso un servizio attivo/in ascolto su una porta remota di un server, che non deve necessariamente coincidere col server SSH con cui è stato attivato il tunnel, è sufficiente che sia sulla stessa sottorete del server SSH.

Ad esempio il traffico indirizzato alla porta 8080 del computer locale, potrebbe essere inoltrato sulla porta 80 del Server SSH col quale abbiamo stabilito la connessione(tunnel) o su un altro server presente nella sua stessa rete.


Your Computer
<local_port>
"loopback interface"
Secure SSH Connection
SSH Server
<SSH_port>
Nonsecure Connection
Destination Server
<destination_port>

Esempio

Immaginiamo di volerci collegare dal computer di casa, utilizzando la porta 8080, verso un server HTTP presente a scuola in ascolto sulla porta 80 ed avente indirizzo LAN 192.168.3.1.

Supponiamo, inoltre, che l'IP pubblico del server SSH a scuola sia 69.80.200.34, in questo caso è necessario stabilire una connessione mediante il comando seguente:

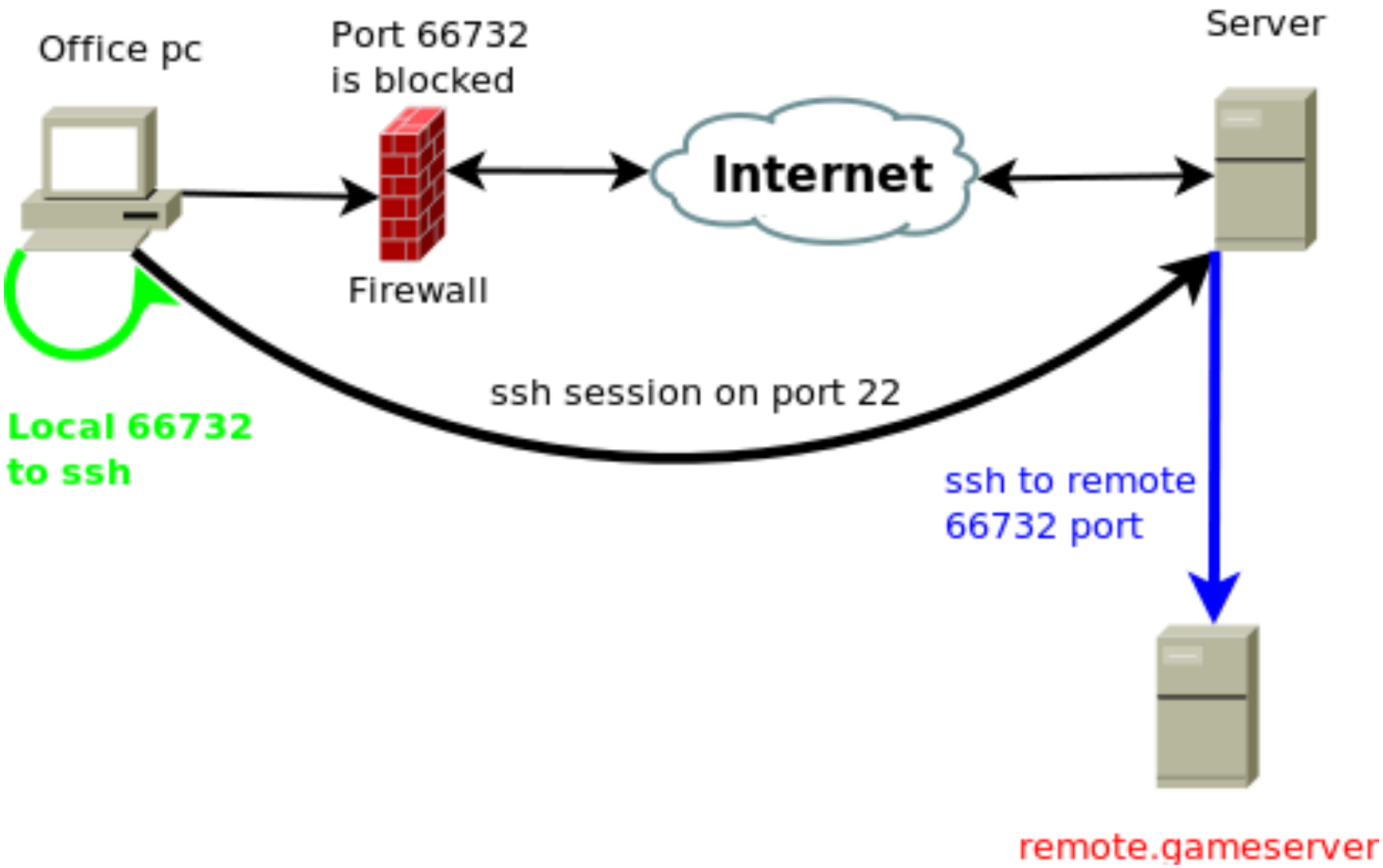
ssh utente-sul-server@69.80.200.34 -L 8080:192.168.3.1:80

- ***ssh*** (nome del comando)
- ***utente-sul-server@69.80.200.34*** (indica il server SSH al quale connettersi mediante il suo IP pubblico e l'utenza abilitata su quel server)
- ***-L*** (opzione di comando per il local port forwarding)
- ***8080:192.168.3.1:80*** (afferma che le connessioni locali verso la porta 8080, vanno inoltrate al server con indirizzo LAN 192.168.3.1 presente nella rete del luogo di lavoro e vanno indirizzate al servizio in ascolto sulla porta 80)

A questo punto se apro il browser del mio computer di casa e sulla barra degli indirizzi digito: **http://localhost:8080**, mi viene aperta la home page presente sul server web della scuola.

Remote SSH port forwarding

Il remote port forwarding è il complementare del local port forwarding, ossia provvede ad inoltrare il traffico da una porta remota verso una porta locale (es. da scuola verso un pc della LAN di casa mia)


Office pc
Port 66732
is blocked
Server
Internet
Firewall
ssh session on port 22
Local 66732
to ssh
ssh to remote
66732 port
remote.gameserver

Esempio

L’esempio in figura mostra come garantirsi la possibilità di accedere da scuola al game server di casa…ecco cosa fare:

Quando siete a casa digitate il comando seguente:

ssh utente-server-ssh-scuola@IP-server-ssh-scuola -R 62732:192.168.2.250:62732

Con questo comando mi garantisco l’accesso dall’ufficio, utilizzando la porta 62732, al game server di casa con indirizzo LAN 192.168.2.250 attivo sulla porta 62732 (le due porte non devono necessariamente coincidere).