Network Security
Elements of Network Security Protocols

Secure Shell
(SSH)

- **Architettura**

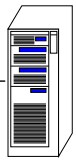
Remote shell (rsh)

Bob è un utente di
foo.iet.unipi.it
(host remoto, client)

```
foo> rsh bar.ing.unipi.it
```

Con rsh ci si può collegare ad un target host (server) da un host remoto (client), oppure si può eseguire un comando su di esso, senza dover inserire la password, purché certi criteri di autenticazione siano soddisfatti.

Bob è un utente di
bar.ing.unipi.it
(host target, server)



`port 22`

Autenticazione ed Autorizzazione

- Quando un utente (Bob) chiede un servizio ad una host target (bar), come fa questi a decidere se la richiesta può essere autorizzata?
- L'host target esegue le seguenti azioni
  I. autentica (identifica) l'utente
     Bob è proprio Bob?
  II. autorizza l'utente
     Bob ha i diritti per usufruire del servizio?

Criteri di Autenticazionedi rsh

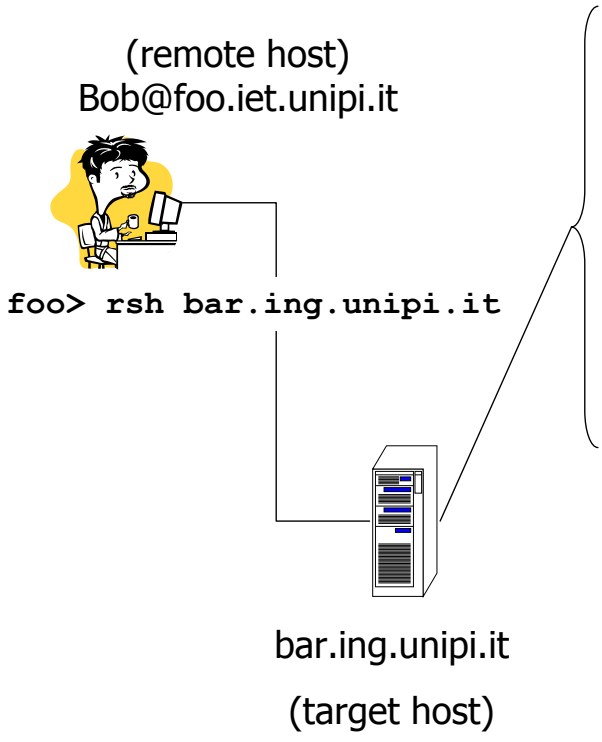
Una host target (bar) accetta il collegamento di un utente (Bob) da un host remoto (foo) se le seguenti condizioni sono verificate:

- La richiesta proviene da una porta TCP privilegiata
- L'host remoto (foo) è fidato, cioè è elencato in

  /etc/hosts.equiv@bar oppure in

  ~Bob/.rhosts@bar

Esempio



- il file /etc/hosts.equiv contiene

  foo.iet.unipi.it [Bob]

  oppure
- il file ~Bob/.rhosts contiene

  foo.iet.unipi.it [Bob]

Ipotesi

- Il meccanismo di autenticazione di rsh si basa sulle seguenti ipotesi:
  1) L'indirizzo di un host è una prova sufficiente della sua identità (address-based authentication)
  2) Un host fidato identifica correttamente un utente
  3) Solo il superuser può utilizzare porte privilegiate
- In pratica, l'host target ragiona così:

  Il superuser (3) di una macchina fidata (1) mi dice che un suo utente (2) si vuole collegare, siccome l'utente ha un account allora lo faccio collegare

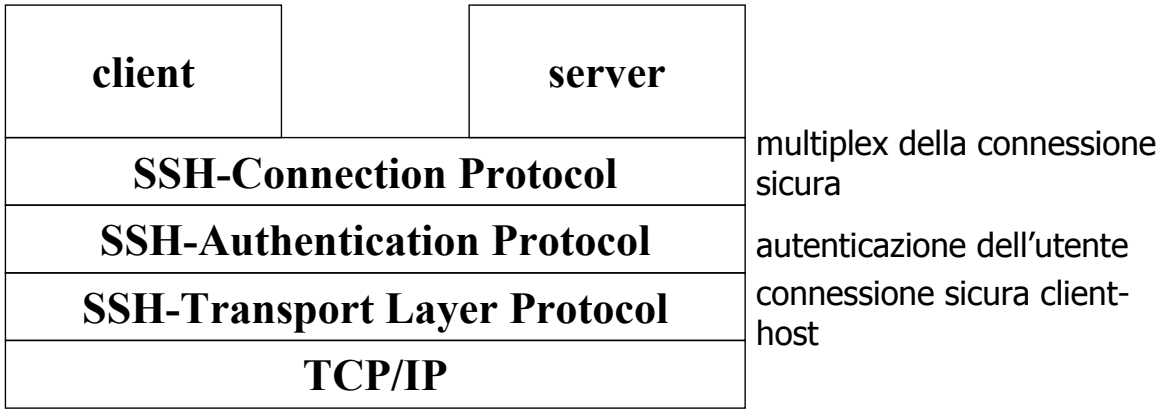
Limitazioni (1)

- Il meccanismo di autenticazione basato sull'indirizzo è debole
  - Tecniche di spoofing (IP spoofing, DNS spoofing, Routing spoofing) rendono falsa l'ipotesi 1
  - Per mezzo di queste tecniche un avversario può indurre l'host target a ritenere fidata un host remoto che in realtà non lo è
- Possibili alternative
  - Telnet
    - L'autenticazione è basata sulle password, ma la password è trasmessa in chiaro
  - Secure shell (ssh)

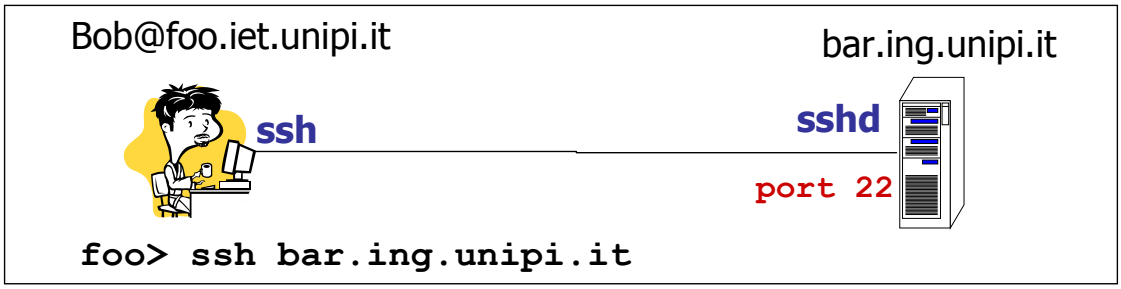
Limitazioni (2)

- I file .rhosts/hosts.equiv contengono l'elenco delle macchine fidate
  - Il file /etc/hosts.equiv è gestito dall'amministratore di sistema
  - Il file .rhosts è gestito dall'utente, che
    - può quindi "battezzare" come affidabile una macchina remota
    - può rendere .rhosts accessibile in scrittura da tutti
- costituiscono dei validi "ingressi" per un nemico
- forniscono "suggerimenti" su quali altre macchine attaccare

Architettura di ssh



Overview



1. Si stabilisce una connessione sicura tra l'host remoto (client) e l'host target (server)
2. Si autentica l'utente usando la connessione sicura
3. Si lancia una shell con cui l'utente interagisce attraverso la connessione sicura

   Le comunicazioni ai punti 1-2 sono crittate con una chiave di sessione stabilita in al punto 1

Host Key

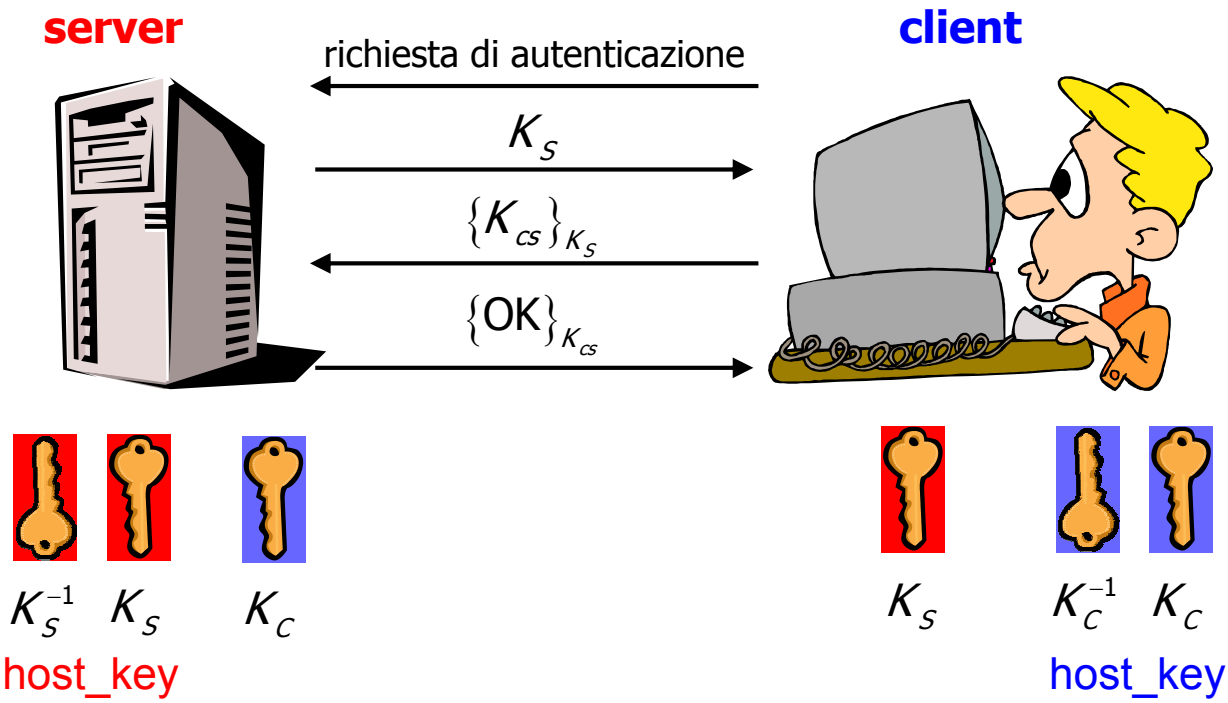
Ogni host ha una coppia di chiavi (host key)

- /usr/local/etc/ssh_host_key contiene la chiave privata
  - creato all'installazione
  - owned dalla root
  - readable and writable solo da root
- /usr/local/etc/ssh_host_key.pub contiene la chiave pubblica
  - creato all'installazione
  - owned da root
  - writable solo da root; readable da all

SSH Transport Layer Protocol

- Autentica l'host target ed esegue il Key Exchange stabilendo una chiave di sessione tra host remoto ed host target
- La corrispondenza tra un server e la sua chiave pubblica deve essere verificata dall'host remoto (client)
  - Il client confronta la server host public key con quella memorizzata in un database locale;
  - normalmente, ma non necessariamente, un client accetta host public key di host sconosciuti e le memorizza nel database per utilizzi futuri
  - nessuna verifica: il protocollo è più facilmente usabile ma diviene soggetto al man-in-the-middle

Key exchange



SSH authentication protocol

L'utente viene autenticato con uno dei seguenti metodi:

- RhostAuthentication: address-based authentication basato su rhosts/hosts.equiv (dovrebbe essere disabilitato)
- RhostRSAAuthentication (HostBasedAuthentication): sistema di autenticazione misto in cui l'host remoto è autenticato per mezzo della sua host key e l'autenticazione dell'utente è basta su .rhost/ hosts.equiv
- RSAAuthentication: sistema di autenticazione dell'utente completamente basato sulla crittografia a chiave pubblica
- TISAuthentication: sistema di autenticazione dell'utente basato su di un authentication server
- PasswordAuthentication: sistema di autenticazione dell'utente basato sulle password

Configurazione

Le modalità di autenticazione sul target host sono specificate nel file sshd_config

Esempio di sshd_config

```
HostBasedAuthentication yes
IgnoreRHosts yes
IgnoreUserKnownHosts yes
PasswordAuthentication no
RhostsAuthentication no          #protocol 1 only
RhostsRSAAuthentication yes  #protocol 1 only
```

Questa autenticazione accredita tutti gli utenti del client; per modificare questa politica si può usare AllowUsers e DenyUsers

Metodo RhostRSAAuthentication (continua)

Il server mantiene la lista degli host conosciuti

- /usr/local/etc/ssh_known_hosts (file di sistema)
- $HOME/.ssh/known_hosts (file d'utente)

questi file contengono coppie (host name, public host key)

La connessione viene concessa se:

1. la richiesta proviene dal corretto indirizzo IP;
2. se il client appare tra host conosciuti;
3. se l'indirizzo IP del cliente appare in hosts.equiv o $HOME/.rhosts

Metodo RhostRSAAuthentication

- Questo metodo di autenticazione richiede che l'indirizzo IP del cliente rimanga costante,
  - un vincolo non adatto per un laptop usato in viaggio
  - La dipendenza dall'IP aggiunge un po' di sicurezza in quanto anche se la chiave di un client viene rubata, essa non può essere utilizzata senza la contraffazione dell'indirizzo IP
- È possibile ovviare a questa dipendenza dall'IP del client usando l'autenticazione [RSA,DSA]Authentication
  - si basa sulla presenza di una chiave privata nel key-ring dell'utente
- [RSA,DSA]Authentication ed RhostRSAAuthentication possono essere combinate: ssh applica prima l'una e poi l'altra

Metodo RSAAuthentication

- Ogni utente possiede una coppia di chiavi pubblica e privata
  - $HOME/.ssh/identity (-rw-------) contiene la chiave privata
  - $HOME/.ssh/identity.pub (-rw-rw-r--) contiene la chiave pubblica

    (ssh-keygen, passphrase opzionale)
- Il server conosce le chiavi pubbliche dei clienti autorizzati a collegarsi
  - $HOME/.ssh/authorized_keys (-rw-r--r--) contiene le chiavi degli utenti che possono collegarsi
- L'autenticazione dell'utente di tipo challenge-response

  Il server invia all'utente una challenge cifrata con la chiave pubblica dell'utente. L'utente (ssh) decifra la challenge e la ritorna in chiaro al server.

Metodo PasswordAuthentication

- Questo metodo viene utilizzato se ogni altro metodo fallisce
- Ssh chiede all'utente la password che viene inviata al server attraverso la connessione sicura stabilita dalla chiave di sessione

Configurazione in {iet, ing}.unipi.it

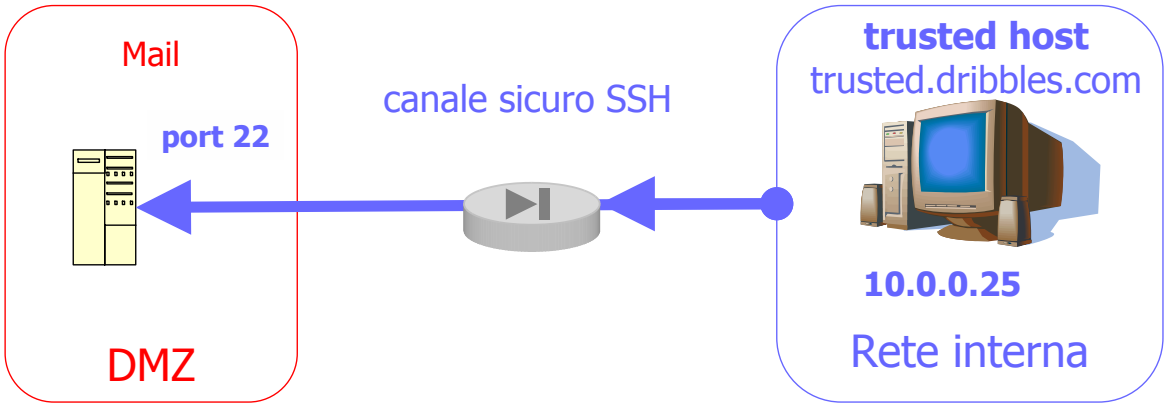
Metodo	Abilitato	Ordine
RhostsAuth	NO	–
RhostsRSAAuth	SI	1
RSAAuth	SI	2
TISAuth	NO	–
PasswordAuth	SI	3

La colonna Ordine indica l'ordine con cui i vari metodi sono applicati

Autenticazione a due fattori

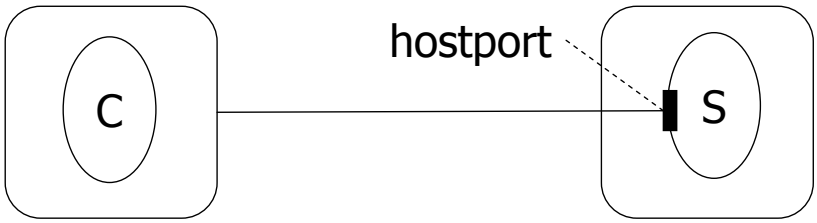
- Finora abbiamo considerato un'autenticazione a fattore singolo (sola chiave pubblica)
  - quando il client ha un livello di sicurezza più elevato del server
  - quando le probabilità che il client sia compromesso sono basse
- Nell'autenticazione a due fattori una passphrase protegge la chiave DSA/RSA
  - se la chiave sta su un laptop e questo viene rubato, la passphrase è l'unica protezione

Connessioni SSH



- La connessione SSH è diretta e non mediata da un proxy. Tuttavia,
- il firewall garantisce che la connessione sia originata da un host interno trusted e sia diretta solo a server in DMZ
- solo gli amministratori di rete hanno accesso al trusted host
- il canale SSH è sicuro

Connection forwarding




```
ssh … -L port:host:hostport …
```