# Uso e configurazione di SSH

Paolo Amendola
GARR-CERT
paolo.amendola@ba.infn.it

# Uso e configurazione di SSH

- Perche' SSH
- Cenni sulla crittografia
- Come funziona SSH
- Installazione, configurazione ed uso
- Disponibilita' per le varie piattaforme
- Performance

# Perche' SSH

- Debolezza intrinseca del protocollo IP
  - DNS spoofing, IP spoofing, routing spoofing
- I dati viaggiano in chiaro
- Autenticazione tra le parti insicura
- Sicurezza di X insufficiente
- Rischi di alterazione delle sessioni

**SSH risolve tutti questi problemi**

# Cenni sulla crittografia

- Crittografia a chiave simmetrica
  - utilizza una stessa chiave per la codifica e decodifica dei dati, piu' veloce.
- Crittografia a chiave asimmetrica
  - utilizza due chiavi diverse, una per la codifica ed una per la decodifica, piu' lenta.

SSH utilizza un misto delle due tecniche

# Come funziona SSH (1)

- Esistono due versioni del protocollo SSH:
  - SSH 1: la versione 1, piu' vecchia, ma con una licenza d'uso piu' permissiva;
  - SSH 2: la versione 2, piu' sicura, ma con una licenza d'uso piu' restrittiva e disponibile su una varieta' di piattaforme piu' limitata.

A causa di cio' la versione 1 e' molto piu' diffusa

# Come funziona SSH (2)

- Nato per rimpiazzare i comandi Berkeley r* (rsh, rcp, rlogin) con le rispettive versioni sicure (ssh, scp, slogin).

- Tale sostituzione puo' essere fatta in maniera trasparente per l'utente.

# Come funziona SSH (3)

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

# Installazione (1)

- Scaricare la distribuzione e la signature:
  *ftp://ftp.cs.hut.fi/pub/ssh/ssh-1.2.27.tar.gz*
  *ftp://ftp.cs.hut.fi/pub/ssh/ssh-1.2.27.tar.gz.sig*

- Scaricare la chiave pubblica pgp di ssh, presente nel file:
  *ftp://ftp.cs.hut.fi/pub/ssh/README*

# Installazione (2)

- Aggiungere la chiave al proprio keyring:

  *$ pgp -ka README*

  - Bits: **1024**
  - KeyID: **DCB9AE01**
  - Data: **24/04/1995**
  - User ID: **Ssh distribution key <ylo@cs.hut.fi>**
  - Fingerprint: **C8 90 C8 5A 08 F0 F5 FD 61 AF E6 FF CF D4 29 D9**
- Verificare la signature:

  *$ pgp ssh-1.2.27.tar.gz.sig ssh-1.2.27.tar.gz*

# Installazione (3)

- Scompattare la distribuzione e compilare:
  - $ *gunzip -c ssh-1.2.27.tar.gz | tar xvf -*
  - $ *cd ssh-1.2.27*
  - $ *./configure [parametri]*
  - $ *make*
  - $ *su*
  - $ *make install*

Viene generata la coppia di chiavi per l’host

# Installazione (4)

- E’ possibile dare alcuni parametri al comando *configure* per personalizzare la compilazione. Per esempio:
  - *--with-libwrap*

    per utilizzare il supporto ai TCP wrappers.
  - *--with-rsh=<path di rsh>*

    per sostituire del tutto i comandi r*.
  - *--help*

    per avere una lista completa dei parametri.

# Configurazione - elenco files

## Files utilizzati sul client

/etc/ssh_config
/etc/ssh_known_hosts
$HOME/.ssh/known_hosts
$HOME/.ssh/config
$HOME/.ssh/identity
$HOME/.ssh/identity.pub

## Files utilizzati sul server

/etc/sshd_config
/etc/ssh_host_key
/etc/ssh_host_key.pub
/etc/ssh_known_hosts
/etc/hosts.equiv
/etc/shosts.equiv
$HOME/.ssh/authorized_keys
$HOME/.ssh/known_hosts
$HOME/.rhosts
$HOME/.shosts

# Configurazione - ssh_config

- Host
- Cipher
- Compression
- CompressionLevel
- ForwardX11
- IdentityFile
- LocalForward
- RemoteForward
- StrictHostKeyChecking
- User
- UseRsh

# Configurazione - sshd_config

- AllowHosts
- AllowTCPForwarding
- AllowUsers
- DenyHosts
- DenyUsers
- HostKey
- PermitRootLogin
- Port
- X11Forwarding

# Configurazione - sostituzione dei comandi r* (1)

- Spostare i comandi rsh, rlogin e rcp in una directory apposita, ad esempio /usr/bin/oldrsh/

  *$ mv /usr/bin/rsh /usr/bin/rlogin /usr/bin/rcp \ /usr/bin/oldrsh/*

- Compilare ssh fornendo al comando *configure* i seguenti parametri:

  *$ configure --with-rsh=/usr/bin/oldrsh/rsh \ --program-transform-name='s/^s/r/'*

# Configurazione - sostituzione dei comandi r* (2)

- Inserire sul server le chiavi pubbliche degli host client in */etc/ssh_known_hosts*.
- Il nome dell'host client deve essere inserito in */etc/hosts.equiv* o in *$HOME/.rhosts* (o, preferibilmente, in */etc/shosts.equiv* o *$HOME/.shosts*).

All'utente e' permesso il login senza password.

# Uso - gestione delle proprie chiavi (1)

- Ogni utente puo’ generare una propria coppia di chiavi:

  *$ ssh-keygen -b 1024 -N ‘password’*

- Le chiavi generate vengono inserite nei files:

  *$HOME/.ssh/identity* (chiave privata)

  *$HOME/.ssh/identity.pub* (chiave pubblica)

# Uso - gestione delle proprie chiavi (2)

- L'utente copia  la propria chiave pubblica all'interno del file *$HOME/.ssh/authorized_keys* sul server.
- Per effettuare il login e' richiesta all'utente la password per sbloccare la propria chiave privata (quella scelta all'atto della generazione).

# Uso - ssh-agent

- ssh-agent memorizza le chiavi private dell'utente, in modo da non dover digitare la password per sbloccarle ogni volta.

  *$ ssh-agent [comando]*

- Per aggiungere delle chiavi all'agent:

  *$ ssh-add [file della chiave privata]*

  (viene richiesta la password per sbloccare la chiave)

# Uso - ssh/slogin

- slogin e’ un link simbolico a ssh

- Sintassi:

  *$ ssh [-vxC] [-l username] [-c cipher] host [comando]*

# Uso - scp

- Sintassi:

  *$ scp [-vrC] [-c cipher] [[user@]host1:]file1 \ [[user@]host2:]file2*

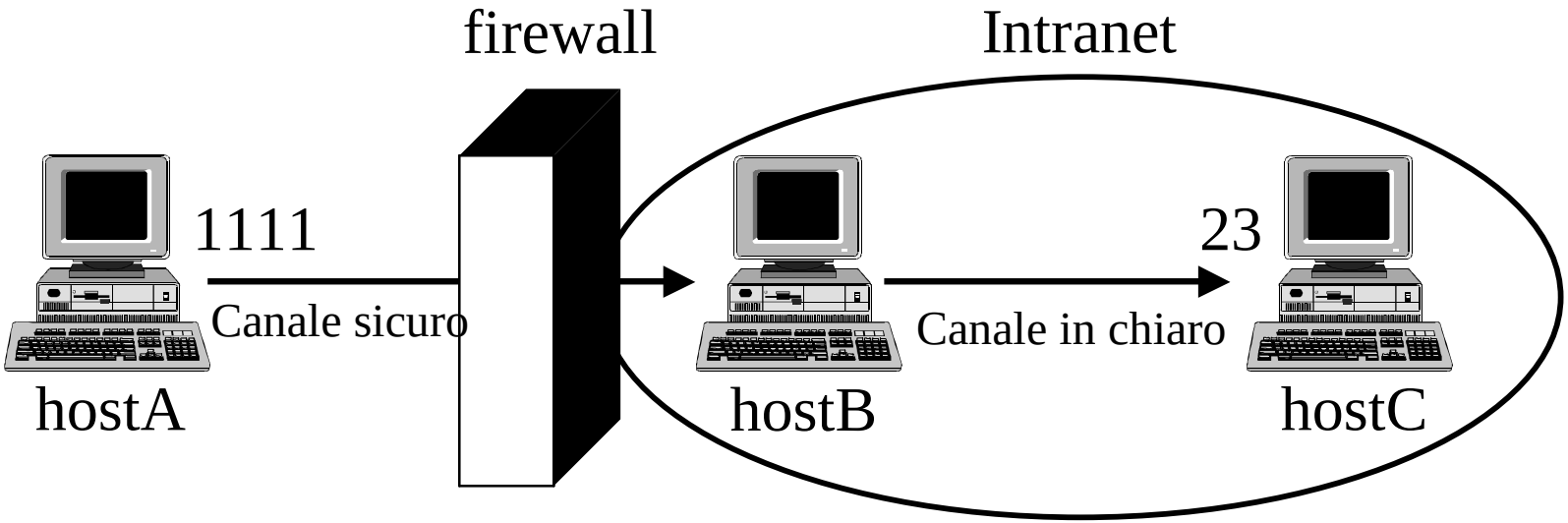
- Esempio:

  *$ scp -v pippo.ps utente@host.dominio:pluto.ps*

# Uso - X11 forwarding

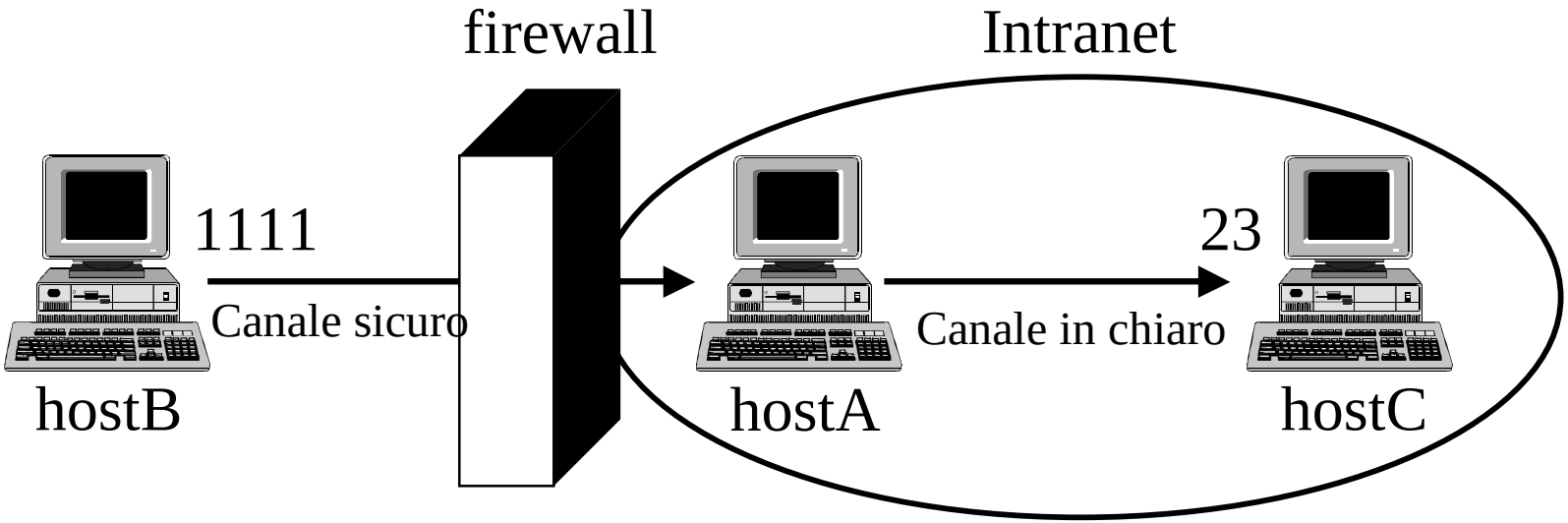
- X11 forwarding automatico se la variabile DISPLAY sul client e' settata.
- Viene creato un X server "virtuale" sul server e la variabile DISPLAY viene modificata per puntarvici.
- La sessione X viene quindi crittografata e forwardata al vero DISPLAY.

# Uso - local port forwarding




*hostA> ssh -L 1111:hostC:23 hostB sleep 100*

# Uso - remote port forwarding




*hostA> ssh -R 1111:hostC:23 hostB sleep 100*

# Disponibilita’

- Esistono client per la grande maggioranza dei dialetti Unix, OpenVMS, Windows 3.x, Windows 9x, Windows NT, Windows CE, Macintosh, Palm Pilot, BeOS, OS/2 e anche in Java.

- Per i server la situazione e’ un po’ piu’ desolante.

# Disponibilita’ -Windows

- Per quanto riguarda Windows, consigliamo Teraterm Pro+TTSSH:

  *http://hp.vector.co.jp/authors/VA002416/teraterm.html*

  *http://www.zip.com.au/~roca/ttssh.html*

- Oppure:

  *http://www.doc.ic.ac.uk/~ci2/ssh/*

# Performance

- L’uso di ssh appesantisce la connessione in modo dipendente dall’algoritmo di crittografia utilizzato.

(Test su connessioni X11)

| | | | |
|---|---|---|---|
| Telnet | 133% | | |
| None | 100% | DES | 70% |
| Arcfour | 88% | IDEA | 68% |
| Blowfish | 80% | 3DES | 45% |

# Per maggiori informazioni

- SSH Communications Security

  http://www.ssh.org
- Data Fellows

  http://ww.datafellows.com
- SSH FAQ

  http://www.employees.org/~satch/ssh/faq

# Queste slides sono disponibili all'URL:

http://www.cert.garr.it/