

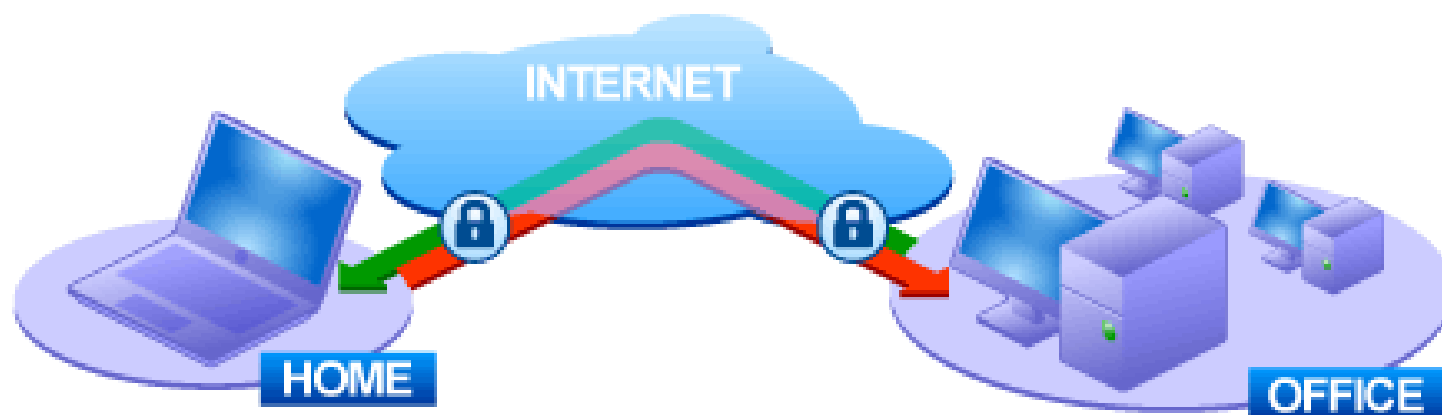
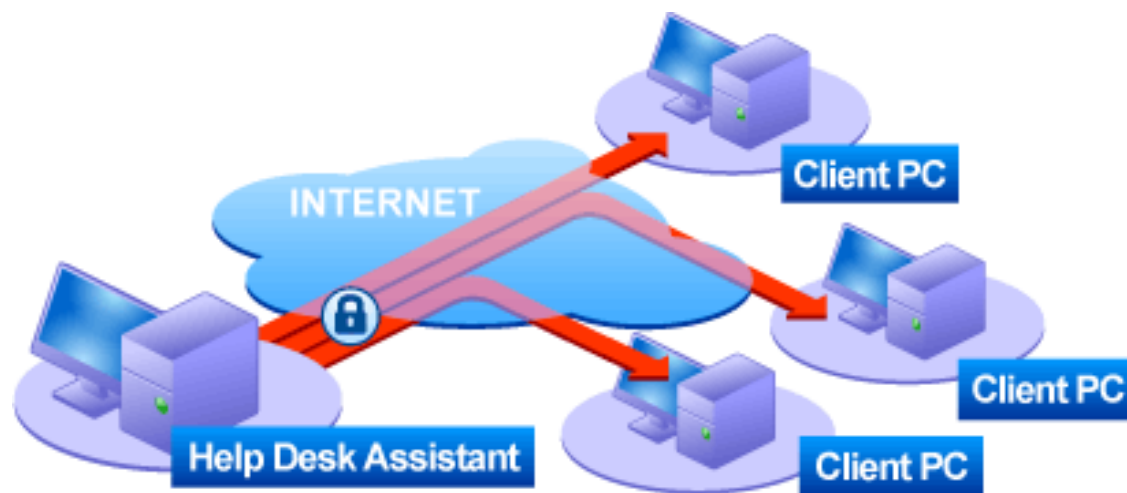


Configurazione e Uso Avanzato di SSH

Luca Bruno (kaeso)
<lucab@debian.org>
14/05/2009



Gestione remota





Implementazione standard



- Protocollo definito negli RFC 4251/4256, a opera del gruppo di lavoro IETF “secsh”
- L'implementazione più diffusa è quella curata dal team OpenBSD
- Rilasciata sotto licenza (perlopiù) *BSD 2-clause* e reperibile presso www.openssh.org



Portable e patch



- Lo stesso team cura il branch **portable** per SO diversi da OpenBSD
- Il nostro riferimento contiene patch di terze parti, reperibili presso <http://diff.debian.net/package/openssh>
- Downstream diversi possono fornire pacchetti frammisti con codice GPL



Advertising

www.***OpenSSH***.com

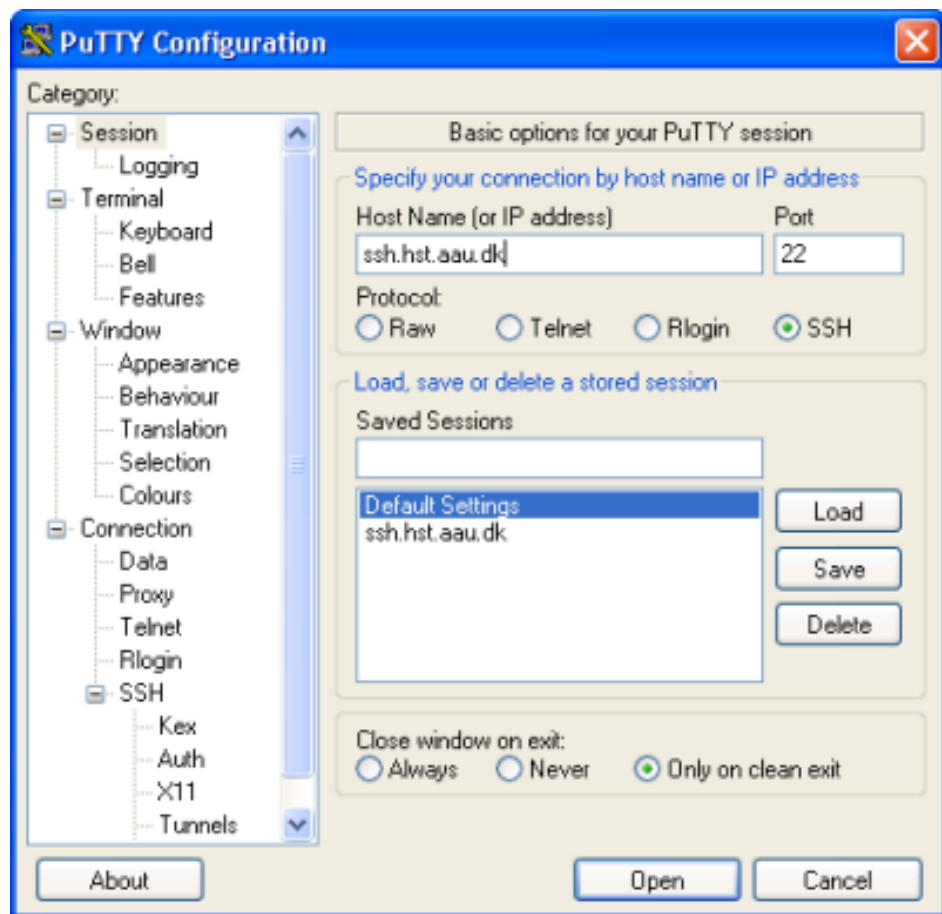


Putting an end to unencrypted network logins



Altre interfacce

- *Grafiche*
 - Hotssh
 - Putty
 - Secpanel
 - Grcm
- *Parallele/clusterizzate*
 - Dish
 - Pssh
 - Clusterssh
- *Embedded/minimali*
 - Dropbear
 - Lssh





Uso base: login remoto

```
lucab@thetis:~$ ssh lucab@io.debian.net
```

```
lucab@io:~$ uname -a
```

```
GNU/kFreeBSD io.debian.net 7.1-1-686
```

```
lucab@io:~$ ssh lucab@albeniz.debian.org
```

```
lucab@albeniz:~$ uname -a
```

```
GNU/Linux albeniz 2.6.26-1-alpha-generic  
alpha
```



Uso base: copia remota

```
lucab@thetis:~$ scp luca@inkscape.org:  
    /etc/gentoo-release .
```

```
gentoo-release      100%    37      0.0KB/s
```

```
lucab@thetis:~$ cat gentoo-release  
Gentoo Base System release 1.12.11.1
```




Uso base: ftp remoto

```
lucab@thetis:~$ sftp people.debian.org
Connecting to people.debian.org...
sftp> pwd
Remote working directory:
/home/lucab
sftp> dir
public_html
sftp> help
Available commands:
[...]
```



Uso base: single-hit remoto

```
$ ssh crest.debian.net \  
    'cat /proc/cpuinfo; uptime'
```

CPU: 68060

MMU: 68060

FPU: 68060

Clocking: 49.5MHz

BogoMips: 99.12

Calibration: 495616 loops

21:41:30 up 153 days, 4:55, 0 users,
load average: 0.85, 0.26, 0.21



Uso base: azioni ripetute

```
$ cat ./macchine  
people.debian.org  
unstable.it
```

```
$ parallel-ssh -P -h ./macchine \  
  'hostname'  
unstable.it: katana  
[1] 23:29:41 [SUCCESS] unstable.it 22  
  
people.debian.org: ravel  
[2] 23:29:43 [SUCCESS]  
people.debian.org 22
```



Comincia il viaggio

Riferimenti (RTFM):



- *man ssh*
- *man sshd*

- *man ssh_config*
- *man sshd_config*



Tuning lato client

Configurazione tramite file

- **`~/.ssh/config`**
- **`/etc/ssh/ssh_config`**

Equivalentemente, opzioni specificabili da linea di comando:

```
$ ssh -C -6 -p 18288 -o "VisualHostKey=yes" ::1
```



Opzioni host-specifiche

*Host **

GSSAPIAuthentication=no

CompressionLevel=9

ServerAliveInterval=30

*Host *.unstable.it*

ForwardAgent=yes

Host piano2

Hostname 172.19.172.19

User admin.big

Host dynamic

Hostname foo.dyndns.org

CheckHostIP no

Port=22000

Host muletto.local

UserKnownHostsFile /dev/null

StrictHostKeyChecking no

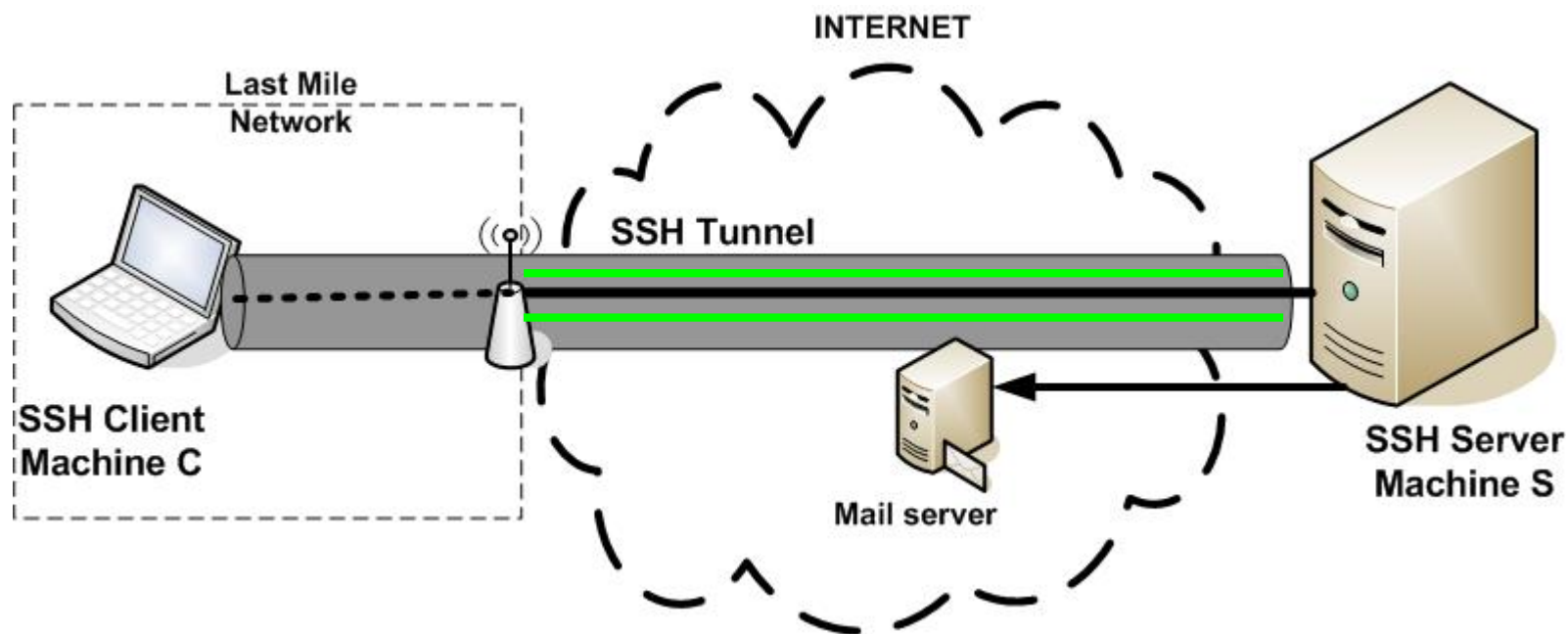


Singolo Canale

Host *

ControlPath ~/.ssh/sock/%r@%h:%p

ControlMaster auto





Ssh console escaping

Ssh possiede una console, raggiungibile tramite **<ENTER>** ~ (nessun risconto visuale).

Sequenze di escape supportate:

- ~? *help ed elenco di comandi*
- ~. *termina connessione*
- ~B *manda un BREAK al sistema remoto*
- ~C *apre la linea di comando*
- ~R *richiede un rekey (solo SSHv2)*
- ~^Z *sospende ssh*
- ~# *elenca le connessioni attivi*
- ~& *manda in background ssh*
- ~~ *escape letterale di ~*



Tuning lato server

Configurazione tramite file

- ***/etc/ssh/sshd_config***

Equivalentemente, opzioni specificabili da linea di comando:

```
$ sshd -f /dev/null -6 -D -p 18288 -o  
    "DenyUsers=test pippo foo"
```



Restrizioni mirate

Match User ups

PasswordAuthentication no

RSAAuthentication yes

PubkeyAuthentication yes

ForceCommand 'foobar'

***AuthorizedKeysFile ***
.ssh/authorized_keys_ups

Match Address 192.168.0.0/16

PermitRootLogin=yes

È un mondo difficile



Pratiche già viste, usate (abusate?) in giro per il mondo

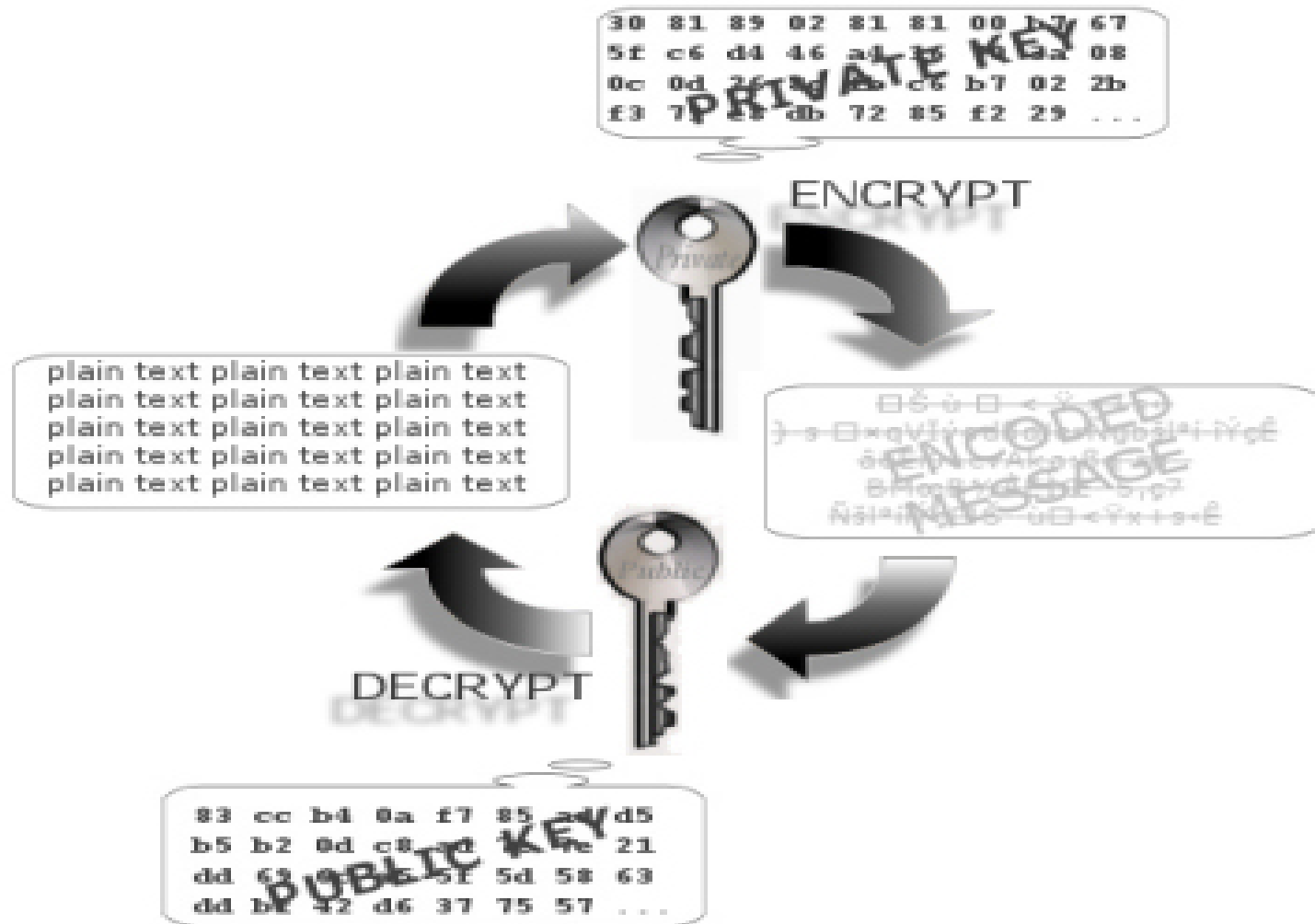


- Port-knocking
- Cambio di porta
- fail2ban, denyhosts
- iptables mirroring/reflecting

Se non assolutamente necessarie, creano quasi più fastidio del problema che dovrebbero risolvere...



Chiavi private e pubbliche





Generazione chiavi

```
$ ssh-keygen
```

Generating **public/private rsa key pair**.

Enter file in which to save the key

(`~/.ssh/id_rsa`):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your **identification** has been saved in

`~/.ssh/id_rsa`.

Your **public key** has been saved in

`~/.ssh/id_rsa.pub`.

The key **fingerprint** is:

**`3e:c0:62:ab:38:3b:7a:77:62:6d:2e:38:9e:d
8:d4:d2 lucab@thetis`**



Parte pubblica

```
$ ssh-keygen -lvf ~/.ssh/id_rsa.pub  
1024 c0:0c:ec:83:42:e9:b1:a8:10:a6:94:4a:62:20:84:44  
.ssh/id_rsa.pub (RSA)
```

```
+--[RSA 1024]----+  
|Oeo..  
|=X .+  
|%.00 +  
|*0. 0 .  
|0. . S  
|. .  
+-----+
```



```
ssh-copy-id -i ~/.ssh/id_rsa.pub s123456@cclix2.polito.it
```



Parte privata

```
$ file ~/.ssh/id_rsa  
~/.ssh/id_rsa: PEM RSA private key
```

```
$ ssh-keygen -e -f ~/.ssh/id_rsa  
---- BEGIN SSH2 PUBLIC KEY ----  
Comment: "1024-bit RSA, converted  
from OpenSSH by lucab@thetis"  
[...]  
---- END SSH2 PUBLIC KEY ----
```



Restrizione di chiavi

```
$ cat .ssh/backup_rsa.pub  
no-port-forwarding,no-X11-  
forwarding,no-agent-forwarding,  
no-pty, command="/usr/local/bin/foobar"  
ssh-rsa [keydata] backup@example.org
```

In questa maniera si possono realizzare **trigger** sicuri!



Net hardcore \m/_

Tunneling, port forwarding,
encrypted proxying, host jumping

ovvero

come sopravvivere se la vostra
rete vi rende claustrofobici...



SOCKSv5 forwarding

```
$ ssh -N -f -D localhost:4444 unstable.it
```

```
$ cat ~/.tsocks.conf  
server = 127.0.0.1  
server_type = 5  
server_port = 4444
```

Vedi ***DynamicForward*** dal man,
nonchè il pacchetto indipendente
tsocks



Local Forwarding

```
$ ssh -L 10080:polito.it:80 \  
    lucab@example.org
```

Ogni connessione a localhost:10080
verrà inoltrata sul canale sicuro a
example.org e di qui alla destinazione
polito.it:80



Remote Forwarding

```
$ ssh -L 2222:localhost:22 \  
    lucab@example.org
```

Ogni connessione alla porta 2222 della macchina remota verrà inoltrata sul canale sicuro alla macchina locale e di qui alla destinazione localhost:22



Tun Forwarding

Ssh consente la creazione di tunnel Layer 2 e 3.

```
$ ssh -f -w 0:0 root@example.org  
'ifconfig tun0 10.0.50.1 pointopoint  
10.0.50.2'
```

```
# ifconfig tun0 10.0.50.2 pointopoint  
10.0.50.1
```

Richiede il server configurato con
PermitTunnel yes (default no)



X11 Forwarding

```
$ ssh -X example.org
```

```
example.org:$ printenv DISPLAY  
localhost:10.0
```

```
example.org:$ gui-app
```

L'applicazione grafica gui-app verrà eseguita sul server remoto, ma mostrerà l'output sul server X locale.



Agent Forwarding

```
$ ssh-agent bash  
$ ssh-add ~/.ssh/id_rsa  
$ env | grep -i ssh  
SSH_AGENT_PID=3599
```

```
$ ssh -A example1.org  
example1.org:~$ ssh example2.org
```

La connessione a example2.org farà uso dell'agent locale, inoltrato su example1.org



Host jumping

Host foo.gl bar.gl

ProxyCommand ssh -q -a -p 443 -x
gluck.debian.org 'nc -w1 \$
(basename %h .gl) 22'



Tips and tricks

RFC 4255

Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints

```
$ dig +short -t sshfp prime.gushi.org  
1 1 931BA3F63CB4BF9C9E7[...]
```



Tips and tricks

Escaping e port forwarding al volo

```
user@remote:~$ <ENTER>~C
```

```
ssh> help
```

Commands:

- L[bind_address:]port:host:hostport

Request local forward

- R[bind_address:]port:host:hostport

Request remote forward

- KR[bind_address:]port

Cancel remote forward