



**Università degli studi di Trieste**

**Laboratorio di reti di calcolatori**

**dee**  
DIPARTIMENTO DI  
ELETTROTECNICA  
ELETTRONICA  
INFORMATICA

# **Installazione e configurazione di**



**Allievi: Argenton Marco  
Codognotto Alberto  
D'Orlando Marco  
Meneghel Francesco  
Relatore: prof. Bartoli Alberto  
Anno accademico: 2001/02**

## Introduzione

La seguente documentazione è stata prodotta al fine di chiarire le modalità d'installazione e configurazione del programma OpenSSH: secure shell, in altre parole un'applicazione (composto da server e client) capace di emulare un terminale Windows, su di una macchina Windows remota. L'esperienza è stata assegnata come esercitazione pratica, al termine del corso di reti di calcolatori. Lo scopo è di installare l'applicazione (lato server e lato client) e di configurare il sistema con le modalità di autenticazione viste a lezione.

## Installazione del server

- Eseguire il programma d'auto installazione "setup.exe". Il programma richiede che gli siano forniti alcuni dati da parte dell'utente (in questo caso administrator):
  - directory di installazione,
  - componenti da installare, server, client o server e client.

L'auto installazione procede col decomprimere i file, nella cartella di installazione.

Genera le coppie di chiavi pubbliche e private dell'host (ssh\_host\_key, ssh\_host\_dsa\_key, ssh\_host\_key.pub, ssh\_host\_dsa\_key.pub i file con estensione .pub contengono le chiavi pubbliche).

Configura automaticamente i file di sistema per l'esecuzione del programma server.

L'installazione del server è stata eseguita con privilegio di administrator.

- Quindi si deve compilare correttamente il file di configurazione degli utenti. Il file si chiama [Install directory]\etc\passwd. Si dovrebbe aprire automaticamente al termine dell'installazione. Altrimenti si deve utilizzare un editor per visualizzarlo e modificarlo.

```
Administrator:::Administrator:::
alberto::500::Alberto Codognotto:/usr_ssh/alberto:/[installdirpath]/ssh/switch.exe
marco::500::Marco D'Orlando:/usr_ssh/marco:/[installdirpath]/ssh/switch.exe
marco2::500::Marco Argenton:/usr_ssh/marco2:/[installdirpath]/ssh/switch.exe
francesco::500::Francesco Meneghel:/usr_ssh/francesco:/[installdirpath]/ssh/switch.exe
```

Sopra si può vedere il file di registrazione utenti compilato. Ci sono vari campi, separati dal carattere ":".

Il primo campo rappresenta il nome utente, il secondo deve rimanere vuoto, infatti è il campo password e non deve essere memorizzato sul disco in "chiaro". Il terzo campo è lo userid che abbiamo posto 500 per ogni utente. Questo è necessario per sistemi windows per rendere possibile l'autenticazione a chiave pubblica come indicato nella documentazione. Il quarto campo deve rimanere vuoto. Il quinto campo rappresenta la descrizione utente. Il sesto campo la home directory dell'utente sul server. Il settimo campo identifica quale shell aprire all'utente dopo il logon.

Dato che OpenBSD (da cui è estratto OpenSSH) è simile ai sistemi UNIX, non si possono considerare le altre unità DOS servendosi della notazione lettera-due punti (es: D:, E:, etc) per specificare un'unità diversa anziché C:, si deve scrivere: /cygdrive/DRIVER\_LETTER, es: /cygdrive/d/usr\_ssh.

E' importante che la linea administrator non venga eliminata.

Si sottolinea, inoltre, che gli utenti devono essere "utenti di Windows NT" della macchina server. La password a cui si fa riferimento, usata per l'autenticazione "debole", è quella di accesso alla macchina server.

- Creazione delle varie directory utente [ Home ], in questo caso si dovranno creare le directory:

```
Es:  c:\usr_ssh\alberto
      c:\usr_ssh\marco
      c:\usr_ssh\marco2
      c:\usr_ssh\Francesco
```

Per ogni utente poi si dovrà creare una sotto cartella .ssh.

```
Es:  c:\usr_ssh\alberto\.ssh
```

- Si compila correttamente il file start-sshd.bat:  
E' necessario scrivere correttamente la posizione del file cmd.exe. Nel file si troverà una voce:  
Set ALTCMD= indica di default la directory c:\winnt\system32\cmd.exe  
Per cui se il sistema operativo NT è installato in una directory diversa dalla consueta si dovrà aggiungere il path effettivo, specificando dove si trova il file:  
Es: Set ALTCMD=d:\winnt\system32\cmd.exe
- Creazione dei file: .rhosts, .shosts, authorized\_keys, authorized\_keys2. E' necessaria una copia per ogni utente. Di default questi file non sono creati automaticamente dal programma di installazione, perchè sono utilizzati solo dalla modalità di autenticazione più "debole", quindi si dovranno creare manualmente. La loro presenza tuttavia non creerà problemi agli altri tipi d'autenticazione ne falle di sicurezza.
  - o [HOME]/.rhosts è costituito da coppie nome host, nome utente, separate da uno spazio, una per riga.
  - o [HOME]/.shosts è costituito da coppie nome host, nome utente, separate da uno spazio, una per riga come .rhosts.
  - o [HOME]/.ssh/authorized\_key(2) elencano le chiavi pubbliche (RSA\DSA) che possono essere utilizzate dall'utente per l'autenticazione.  
Dato che abbiamo utilizzato il protocollo 1 che si avvale di chiavi RSA, è stato sufficiente migrare sul server il file identity.pub (tramite floppy) e rinominarlo (qualora venga utilizzato il protocollo 2, le chiavi in id\_dsa.pub e id\_rsa.pub devono essere inserite nel file authorized\_key2).
- Creazione dei file rhost.equiv, shost.equiv.
  - o [Install directory]/etc/rhost.equiv il file è costituito da il nome degli host abilitati al logon nel host-server in questione. Si possono inoltre specificare i nomi degli utenti, per ogni host, abilitati alla connessione al server. Il formato del file è nome: host(space)utente1(space)utente2(space)utente3 e così via.
  - o [Install directory]/etc/shost.equiv analogo al file precedente.

## Configurazione del server

Dopo aver creato tutti i file necessari per i vari metodi di autenticazione supportata, si deve procedere con la configurazione del sistema. Il file preposto a tale scopo è denominato `sshd_config`. Nella installazione di default è contenuto nella directory di installazione.

Riportiamo qui il contenuto del file

```
HostKey /ssh/ssh_host_key
HostDSAKey /ssh/ssh_host_dsa_key
PidFile /ssh/sshd.pid
Protocol 1

PermitRootLogin yes
PasswordAuthentication yes
IgnoreRhosts yes
IgnoreUserKnownHosts yes
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes

SyslogFacility AUTH
LogLevel INFO

Subsystem sftp /ssh/sftp-server
```

Rispetto alla versione di default, abbiamo permesso l'utilizzo del solo protocollo 1 .

## Installazione Client

- Analogamente a quanto fatto in precedenza sul lato server, basta mandare in esecuzione su un'altra macchina il file "setup.exe" selezionando l'installazione **del solo client**. Abbiamo fatto questo perché non ci sembra ragionevole installare il server su più nodi della rete: si noti che questo particolare sarà rilevante per le considerazioni che riporteremo in seguito.
- Installando il solo client non avviene alcuna generazione di chiavi di nodo come avveniva in automatico per quanto riguarda il server.
- Abbiamo generato il file [install directory]/ssh/.ssh/known\_hosts contenente la chiave pubblica del server. Il formato è il seguente:  
[nome DNS del server],[ip address](space)[chiave].  
Il file qualora non esista viene creato automaticamente al primo tentativo di connessione al server chiedendo all'utente il permesso di aggiungere la chiave pubblica del server.  
Per questioni di sicurezza ci è sembrato corretto definire a priori il server fidato.
- A questo punto non resta altro che creare la/le coppia/e di chiavi utente: questo viene fatto mediante l'applicazione ssh-keygen.exe. E' possibile generare i seguenti file: identity e identity.pub (contenenti chiavi RSA usate dal protocollo1), id\_dsa e id\_dsa.pub (contenenti chiavi dsa usate dal protocollo 2), id\_rsa e id\_rsa.pub (contenenti chiavi RSA usate dal protocollo2). Questi vanno in [install directory]/ssh/.ssh
- In fase di generazione, abbiamo sfruttato l'opzione di protezione delle chiavi mediante una "passphrase" che viene poi richiesta in fase di autenticazione (cfr. log1)
- Per la connessione remota l'utente si serve del programma ssh.exe. E' possibile passare parametri opzionali o tramite linea di comando (es: opzione -v "verbose" per visualizzare i log) o tramite il file config (stesso formato di ssh\_config sul lato server) che si può opzionalmente creare in [install directory]/ssh/.ssh. In questo modo è possibile forzare il client ad usare una particolare autenticazione tra quelle permesse dal server.

## Considerazioni

- Una volta terminata la fase di installazione e configurazione del tool, abbiamo provato ad utilizzare le modalità di autenticazione viste a lezione. Come prima cosa, abbiamo notato che l'autenticazione fra host non è mutua<sup>1</sup>: il client infatti non ha nessuna chiave di nodo! Questo non risulta subito chiaro dalla documentazione, ma trova conferma dalla esecuzione di ssh in modalità "verbose" (cfr. log1). Si vede chiaramente che il client aspetta la chiave pubblica di nodo del server, la confronta con la propria copia contenuta nel file `Known_hosts` e se coincidono genera un numero casuale, lo critta con la chiave appena ricevuta e lo invia al server; a questo punto il random-number viene usato da entrambi i lati come session-key per crittare la comunicazione (descrizione semplificata).  
Questi passaggi sono sì spiegati nel manuale `sshd`, ma in quest'ultimo ci sono anche frasi fuorvianti che vanno interpretate nel giusto contesto: per esempio si dice che ogni host è caratterizzato da una chiave di nodo, ma dal contesto si capisce si sta parlando di un host in cui sia installato il server.
- Abbiamo notato, inoltre, che l'utente che desidera accedere al server deve essere un utente NT della macchina su cui questo è installato. Lo username ssh deve quindi coincidere con quello NT.  
Non c'è correlazione invece con lo username utilizzato per accedere alla macchina client.  
Quindi nel caso di autenticazione con password l'utente pippo abilitato sul client ma non sul server, può logarsi a quest ultimo dicendo di essere pluto (utente abilitato) a patto di conoscerne la password.
- Le due modalità di autenticazione di utente che siamo quindi riusciti a realizzare sono quella a chiave pubblica e quella tramite password. L'autenticazione utilizzata è la più forte tra quelle ammesse da entrambi i lati. Se per esempio sono previste entrambe le modalità su ambo i lati e se l'utente è sprovvisto delle chiavi, automaticamente il sistema si "abbassa" richiedendo la password; qualora questa venga digitata in modo errato per 3 volte viene restituito il seguente messaggio di errore: "too many authentications failed".  
Si noti che in condizioni operative è giusto che sia il server a fissare la modalità di autenticazione.
- Una prova interessante è stata la reinstallazione del server, con conseguente generazione di nuove chiavi di nodo, senza l'aggiornamento del file `known_hosts` sul lato client.  
Quando l'utente tenta di accedere al server il client non riconosce la nuova chiave, avvisa l'utente con un messaggio di warning e disabilita l'autenticazione con password per evitare che l'utente ignaro digiti la password pregiudicandone la segretezza.  
Il sistema ammette comunque l'autenticazione a chiave pubblica (cfr log2).  
In questo caso l'utente non può essere sicuro di connettersi al server corretto.
- Tra gli allegati inseriamo due capture dello sniffer: una in cui compaiono i pacchetti relativi alla risposta crittata ad un comando "dir", ed una in cui i pacchetti sono sempre relativi ad un comando "dir" ma questa volta eseguito su sessione telnet e quindi in chiaro.

---

<sup>1</sup> Per dissolvere i nostri dubbi circa le modalità di autenticazione fra host, abbiamo contattato via mail uno dei realizzatori del tool presso la NetworkSimplicity: alleghiamo la mail di risposta che conferma le nostre ipotesi.

**LOG1**

```

C:\PROGRA~2\NETWOR~1\ssh>ssh -l -v alberto@egle
OpenSSH_2.9p2, SSH protocols 1.5/2.0, OpenSSL 0x0090600f
debug1: Reading configuration data /usr/lib/.ssh/config
debug1: Seeding random number generator
debug1: restore_uid
debug1: ssh_connect: getuid 500 geteuid 500 anon 1
debug1: Connecting to egle [169.0.0.1] port 22.
debug1: temporarily_use_uid: 500/544 (e=500)
debug1: restore_uid
debug1: temporarily_use_uid: 500/544 (e=500)
debug1: restore_uid
debug1: Connection established.
debug1: identity file /usr/lib/.ssh/identity type 0
debug1: Remote protocol version 1.99, remote software version OpenSSH_2.9p2
debug1: match: OpenSSH_2.9p2 pat ^OpenSSH
debug1: Local version string SSH-1.5-OpenSSH_2.9p2
debug1: Waiting for server public key.
debug1: Received server public key (768 bits) and host key (1024 bits).
debug1: Host 'egle' is known and matches the RSA1 host key.
debug1: Found key in /usr/lib/.ssh/known_hosts:1
debug1: Encryption type: 3des
debug1: Sent encrypted session key.
debug1: Installing crc compensation attack detector.
debug1: Received encrypted confirmation.
debug1: Trying RSA authentication with key 'alberto@HELENA'
debug1: Remote: /usr_ssh/alberto/.ssh/authorized_keys, line 1: bad key syntax
debug1: Remote: /usr_ssh/alberto/.ssh/authorized_keys, line 2: bad key syntax
debug1: Received RSA challenge from server.
Enter passphrase for RSA key 'alberto@HELENA':
debug1: Sending response to host key RSA challenge.
debug1: Remote: RSA authentication accepted.
debug1: RSA authentication accepted by server.
debug1: Requesting pty.
debug1: Requesting shell.
debug1: Entering interactive session.
Last login: Tue Jan 22 12:30:20 2002 from helena
Microsoft Windows 2000 [Versione 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

c:\usr_ssh\alberto>dir
Il volume nell'unit. C S MAIN
Numero di serie del volume: 206E-0FEB

Directory di c:\usr_ssh\alberto

22/01/2002  10.53      <DIR>          .
22/01/2002  10.53      <DIR>          ..
22/01/2002  10.53      <DIR>          .ssh
22/01/2002  11.30              99 .rhosts
22/01/2002  11.31              99 .shosts
                2 File                198 byte
                3 Directory          385.945.600 byte disponibili

c:\usr_ssh\alberto>exit
Connection to egle closed.
debug1: Transferred: stdin 9, stdout 646, stderr 28 bytes in 12.0 seconds
debug1: Bytes per second: stdin 0.8, stdout 54.0, stderr 2.3
debug1: Exit status 0

C:\PROGRA~2\NETWOR~1\ssh>

```

**LOG2**

```

C:\Program Files\NetworkSimplicity\ssh>ssh alberto@egle
debug1: Seeding random number generator
debug1: restore_uid
debug1: ssh_connect: getuid 500 geteuid 500 anon 1
debug1: Connecting to egle [169.0.0.1] port 22.
debug1: temporarily_use_uid: 500/544 (e=500)
debug1: restore_uid
debug1: temporarily_use_uid: 500/544 (e=500)
debug1: restore_uid
debug1: Connection established.
debug1: identity file /usr/lib/.ssh/identity type 0
debug1: Remote protocol version 1.5, remote software version OpenSSH_2.9p2
debug1: match: OpenSSH_2.9p2 pat ^OpenSSH
debug1: Local version string SSH-1.5-OpenSSH_2.9p2
debug1: Waiting for server public key.
debug1: Received server public key (768 bits) and host key (1024 bits).
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@      WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!      @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA1 host key has just been changed.
The fingerprint for the RSA1 key sent by the remote host is
0f:ef:9d:e8:aa:bf:73:60:97:77:e2:f7:ea:ce:d3:21.
Please contact your system administrator.
Add correct host key in /usr/lib/.ssh/known_hosts to get rid of this message.
Offending key in /usr/lib/.ssh/known_hosts:1
Password authentication is disabled to avoid trojan horses.
debug1: Encryption type: 3des
debug1: Sent encrypted session key.
debug1: Installing crc compensation attack detector.
debug1: Received encrypted confirmation.
debug1: Trying RSA authentication with key 'alberto@HELENA'
debug1: Remote: /usr_ssh/alberto/.ssh/authorized_keys, line 1: bad key syntax
debug1: Remote: /usr_ssh/alberto/.ssh/authorized_keys, line 2: bad key syntax
debug1: Received RSA challenge from server.
Enter passphrase for RSA key 'alberto@HELENA':
debug1: Sending response to host key RSA challenge.
debug1: Remote: RSA authentication accepted.
debug1: RSA authentication accepted by server.
debug1: Requesting pty.
debug1: Requesting shell.
debug1: Entering interactive session.
Last login: Tue Jan 22 12:50:50 2002 from max
Microsoft Windows 2000 [Versione 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

c:\usr_ssh\alberto>

```



**Log3**

```
C:\>ssh alberto@egle
debug1: Seeding random number generator
debug1: restore_uid
debug1: ssh_connect: getuid 500 geteuid 500 anon 1
debug1: Connecting to egle [169.0.0.1] port 22.
debug1: temporarily_use_uid: 500/544 (e=500)
debug1: restore_uid
debug1: temporarily_use_uid: 500/544 (e=500)
debug1: restore_uid
debug1: Connection established.
debug1: identity file /usr/lib/.ssh/identity type 0
debug1: Remote protocol version 1.99, remote software version OpenSSH_2.9p2
debug1: match: OpenSSH_2.9p2 pat ^OpenSSH
debug1: Local version string SSH-1.5-OpenSSH_2.9p2
debug1: Waiting for server public key.
debug1: Received server public key (768 bits) and host key (1024 bits).
debug1: Host 'egle' is known and matches the RSA1 host key.
debug1: Found key in /usr/lib/.ssh/known_hosts:1
debug1: Encryption type: 3des
debug1: Sent encrypted session key.
debug1: Installing crc compensation attack detector.
debug1: Received encrypted confirmation.
debug1: Doing challenge reponse authentication.
debug1: No challenge.
debug1: Doing password authentication.
alberto@egle's password:
debug1: Requesting pty.
debug1: Requesting shell.
debug1: Entering interactive session.
Last login: Wed Jan 16 17:57:00 2002 from max
Microsoft Windows 2000 [Versione 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

c:\usr_ssh\alberto>
```

**CommView**

File Search View Tools Settings Rules Help

Allied Telesyn AT-2400 PCI Ethernet Adapter

IP Statistics Packets Logging Rules

**Ethernet II**

- Destination MAC: 00:A0:D2:14:4D:43
- Source MAC: 00:A0:D2:14:4B:DB
- Ethertype: 0x0800 (2048) - IP
- Direction: Out
- Time / Delta Time: 16:00:57,337 / 0,011
- Frame size: 338 bytes

**IP**

- Source port: 22
- Destination port: 3405
- Sequence: 0xAE44C439 (2923742265)
- Acknowledgement: 0x505C11D6 (1348211158)
- Header length: 0x05 (5) - 20 bytes
- Flags: PSH ACK
- URG: 0
- ACK: 1
- PSH: 1
- RST: 0
- SYN: 0
- FIN: 0
- Window: 0x4010 (16400)
- Checksum: 0x0B09 (2825) - correct
- Urgent Pointer: 0x0000 (0)
- TCP Options: None
- Data length: 0x11C (284)

No	Protocol	MAC Addresses	IP Addresses	Ports	Delta
1	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	56,...
2	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	22 => 3405	0,040
3	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	0,151
4	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	11,...
5	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	22 => 3405	0,040
6	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	0,130
7	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	4,256
8	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	22 => 3405	0,040
9	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	0,110
10	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	4,236
11	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	22 => 3405	0,011
12	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	22 => 3405	0,030
13	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	22 <=> 3405	0,010
23	ADD PCN	00:A0:D2:14:4B:DB -> Broadcast	169.0.0.1 -> 169.0.0.2	N/A	16

0x0000 00 A0 D2 14 4D 43 00 A0 D2 14 4B DB 08 00 45 00 . 0.MC. 0.KU..E.  
 0x0010 01 44 11 7E 40 00 80 06 96 31 A9 00 00 01 A9 00 .D..@.e.-1@...@.  
 0x0020 00 03 00 16 0D 4D AE 44 C4 39 50 5C 11 D6 50 18 ...H@dA9P..@P.  
 0x0030 40 10 0B 09 00 00 00 00 01 13 E8 93 0B 07 9D E2 @.....è"...Dâ  
 0x0040 D7 5F A2 05 26 5B 0D FE AF 3C BA 16 80 BA 3B 03 \*...<[.p~<".e"/.  
 0x0050 82 E4 D0 D6 A9 D9 0C 40 C8 8E 96 83 2F 73 B8 25 ,ad00Ü.0E2-f/s,â  
 0x0060 6C 3D 95 79 2B 32 08 22 B3 A7 5C 4C F7 3E 03 04 1=+y+2."\$\\L+>.  
 0x0070 11 9E C8 05 F4 A8 79 95 A2 9F EB 90 5E 80 ED 92 .âE.ô"y\*cYâO^Ei'  
 0x0080 A3 2A 94 37 58 B2 CF 90 56 AD A7 25 C1 8F F1 73 \$+/"X\*IDV-\$\*âDB{  
 0x0090 F5 AF 64 C8 7B F4 C8 3B 50 34 08 DB 48 5C 32 6E ô"âI{ôE;P4.ÜH\2n  
 0x00A0 FC 51 A8 51 DA 62 20 9A B5 AA D0 7F 4D 81 15 46 uQ"QÜb \$u\*POMD.F  
 0x00B0 EA 71 D3 70 A6 54 BC FC DE F5 55 C3 9B 3E 3B 39 èqOp;T\*âB6UÂ>;9  
 0x00C0 B5 32 E7 21 4F 09 B2 1A 56 B9 5D C5 7C CB 04 79 µ2ç!0.+.V)JÂ|E.y  
 0x00D0 C9 64 4B 7B 4E 5D 4A 20 86 9D 78 E2 E8 92 DF 11 EdK(N)J +Oxââ'â.  
 0x00E0 D4 B5 C4 C8 49 EC BE 8F 77 C9 B6 A3 14 CB 86 1D ÔµâEiI\*âwE7E.E+.  
 0x00F0 4E CA DF D6 42 17 63 07 60 FB 65 30 D7 70 22 89 NÊâÖB.c.'ûeOxp"%  
 0x0100 A8 89 4A B2 54 98 E8 EB 61 5B 2D 18 90 B6 81 D8 "%J\*T"è)a[-.OQD@  
 0x0110 62 50 29 90 6D CA 9A 72 FC 39 21 E7 18 C3 6D 30 bP)OmEÿrû9!ç.âm0  
 0x0120 DE 7A 19 A7 17 90 60 E4 A4 60 BF 87 CE 2D 5D 3D Fz.S.O`âx` ;+I-]=  
 0x0130 A3 F3 66 48 EB 9A 07 7E 8B E2 15 E0 82 50 2D 2E 6ôfHâE..~<â.â.P.  
 0x0140 18 C3 42 F0 2F 28 0B B9 9F 26 ED 05 C8 62 20 A1 ÂBâ/(.âYâi.âb ;  
 0x0150 35 33 53

Notare la crittazione

Capture: Off Pkts: 349 in / 284 out / 104 pass Auto-saving: Off Rules: 1 On 100% CPU Usage

Start Amministrazione di server... CommView telnet.TXT - Blocco note 16.06

Capture 1: Esecuzione di 'dir' su ssh.

**CommView**  
File Search View Tools Settings Rules Help

Allied Telesyn AT-2400 PCI Ethernet Adapter

**IP Statistics** **Packets** **Logging** **Rules**

**Ethernet II**  
Destination MAC: 00:A0:D2:14:4D:43  
Source MAC: 00:A0:D2:14:4B:DB  
EtherType: 0x0800 (2048) - IP  
Direction: Out  
Time / Delta Time: 17:59:04.304 / 0.111  
Frame size: 1178 bytes

No	Protocol	MAC Addresses	IP Addresses	Ports	Delta
1	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	0,000
2	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	23 => 4265	0,110
3	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	0,131
4	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	1,091
5	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	23 => 4265	0,100
6	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	0,110
7	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	1,172
8	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	23 => 4265	0,100
9	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	0,130
10	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	1,382
11	IP/TCP	00:A0:D2:14:4B:DB => 00:A0:D2:14:4D:43	169.0.0.1 => 169.0.0.3	23 => 4265	0,111
12	IP/TCP	00:A0:D2:14:4B:DB <=> 00:A0:D2:14:4D:43	169.0.0.1 <=> 169.0.0.3	23 <=> 4265	0,110

**TCP Session**  
File Settings

```

d. [25;5Hd. [25;6Hi. [25;6Hi. [25;7Hr. [25;7Hr. [25;8Hd. [1;2H0/10/2000 21.18 <DIR> export
.. [2;1H17/01/2002 17.08 0 AILog.txt. [3;1H22/01/2002 10.53 <DIR> usr_ssh
.. [4;1H22/01/2002 16.07 2.359.350 commview.bmp. [5;2H2/01/2002 13.17 989
sshd.exe.stackdump. [6;1H 10 File 2.378.948 byte [7;1H 19
Directory 383.193.088 byte disponibili. [8;1H. [K. [9;1HC:\>e:
.. [10;1H. [K. [11;1HE:\>dir .. [12;1H Il volume nell'unit... E § 2000
.. [13;1H Numero di serie del volume: 245B-80F5 .. [14;1H. [K. [15;1H Directory di E:\
.. [16;1H. [K. [17;1H17/12/2001 15.55 <DIR> Documents and Settings. [18;14H0.49 <DIR>
Program Files. [19;1H17/12/2001 17.31 <DIR> Programmi .. [20;1H15/01/2002 12.31
<DIR> RETI. [21;1H22/01/2002 10.50 <DIR> WINNT .. [22;16H0 File
0 byte. [23;1H 5 Directory 1.656.091.648 byte disponibili. [25;5H .. [25;5H
  
```

☒ 169.0.0.3 port 4265 => 169.0.0.1 port 23 : 5 bytes in 4 packet(s)  
☒ 169.0.0.1 port 23 => 169.0.0.3 port 4265 : 1.169 bytes in 4 packet(s)  
 Total 1,174 bytes in 8 packet(s), Session time: 4 second(s)

ASCII HTML  
HEX EBCDIC

0x0160 30 20 46 69 6C 65 20 20-20 20 20 20 32 2E 33 37 0 File 2.37  
 0x0170 38 2E 39 34 38 20 62 79-74 65 20 20 20 20 20 8.948 byte  
 0x0180 20 20 20 20 20 1B 5B 37-3B 31 48 20 20 20 20 20 .. [7;1H  
 0x0190 20 20 20 20 20 20 20-20 31 39 20 44 69 72 65 19 Dire

Capture: On Pkts: 114 in / 94 out / 35 pass Auto-saving: Off Rules: 1 On 1% CPU Usage

Start Strumenti di amministr... Amministrazione di ser... CommView TCP Session 18.00

Capture 2: Esecuzione di "dir" su telnet.