

Molti linux-users hanno una configurazione di rete che prevede un pc sempre connesso all'ADSL, ed un gruppo di pc dietro tale computer. Il compito del pc costantemente connesso -che chiameremo gateway da qui in avanti- e' ovviamente permettere all'intera lan lo sfruttamento dell'ADSL, ed altro ancora. Tale gateway deve essere adeguatamente protetto dagli attacchi provenienti dalla rete esterna. Molte metodologie sono possibili. Noi analizzeremo tre software, e precisamente

- a) portsentry
- b) hostsentry
- c) logcheck

Tutti e tre i software sono amministrabili tramite webmin oppure tramite files di configurazione. Installeremo tutti e tre i software, compreso webmin, ma le nostre attenzioni saranno soprattutto per portsentry.

PREPARAZIONE E DOWNLOAD DEL SOFTWARE

Cominciamo col procurarci un pc, formattiamolo ed installiamo l'impareggiabile slackware 10.1. Successivamente ci procuriamo i software occorrenti: webmin lo scarichiamo da:

<http://www.webmin.com/download.html>

portsentry e logcheck li troviamo all'url:

<http://sourceforge.net/projects/sentrytools>

hostsentry invece e' all'url:

http://www.tucows.com/get/51641_31927

per ultimo il modulo webmin per i tre software:

<http://linux.3jk.com/mirror/webmin/download/modules/>

INSTALLAZIONE:

tutti i pacchetti sono ora nel nostro computer, ipotizziamo in /tmp. Iniziamo con portsentry (diventate root se gia' non lo siete):

```
cd /tmp
tar zxvf portsentry-1.2.tar.gz
```

otteniamo la directory

```
portsentry_beta
```

Entriamo in tale directory:

```
cd portsentry_beta
```

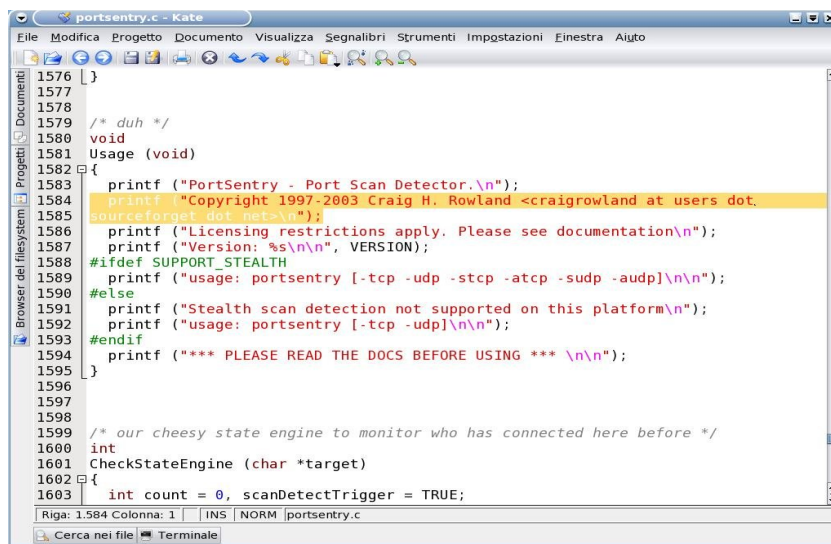
e leggiamo il file README.install. Ora possiamo lanciare la compilazione con il comando

```
make linux
```

...sorpresa !! ecco un bel messaggio di errore !! lo vediamo qui sotto:

```
claudio@orazio:~/Desktop/portsentry_beta$ make linux
SYSTYPE=linux
Making
cc -O -Wall -DLINUX -DSUPPORT_STEALTH -o ./portsentry ./portsentry.c \
    ./portsentry_io.c ./portsentry_util.c
portsentry.c:1584:11: missing terminating " character
portsentry.c: In function `Usage':
portsentry.c:1585: error: parse error before "sourceforget"
portsentry.c:1585: error: stray '\' in program
portsentry.c:1585:24: missing terminating " character
make: *** [linux] Error 1
```

Cosa puo' essere successo visto che siamo partiti dai sorgenti ed in pratica non abbiamo fatto ancora nulla? Semplicemente abbiamo trovato un errore nel Makefile! Lo sviluppatore o chi ha testato il software ha commesso una svista. Correggiamo l'errore aprendo il file portsentry.c alla linea 1585 ed eliminando il rientro a capo sbagliato.



```
1576 }
1577
1578
1579 /* duh */
1580 void
1581 Usage (void)
1582 {
1583     printf ("PortSentry - Port Scan Detector.\n");
1584     printf ("Copyright 1997-2003 Craig H. Rowland <craigrowland at users dot
sourceforget dot net>\n");
1585     printf ("Licensing restrictions apply. Please see documentation\n");
1586     printf ("Version: %s\n\n", VERSION);
1587     #ifdef SUPPORT_STEALTH
1588     printf ("usage: portsentry [-tcp -udp -stcp -atcp -sudp -audp]\n\n");
1589     #else
1590     printf ("Stealth scan detection not supported on this platform\n");
1591     printf ("usage: portsentry [-tcp -udp]\n\n");
1592     #endif
1593     printf ("*** PLEASE READ THE DOCS BEFORE USING *** \n\n");
1594 }
1595
1596
1597
1598
1599 /* our cheesy state engine to monitor who has connected here before */
1600 int
1601 CheckStateEngine (char *target)
1602 {
1603     int count = 0, scanDetectTrigger = TRUE;
```

[FOTO 1]

Praticamente la linea 1584 e 1585 devono diventare una linea sola. Fatto questo possiamo rilanciare il comando

```
make linux.
```

E successivamente lanceremo il comando

```
Make install.
```

Ora abbiamo portsentry installato nella directory `/usr/local/psionic/portsentry/`. Occupiamoci di `logcheck`:

```
cd /tmp
tar zxvf logcheck-1.1.1..tar.gz
cd logcheck-1.1.1
make linux
make install
```

Anche qui potrebbe capitare un errore, il cui output e' visibile qui sotto:

```
root@ludovacca:~/portsentry/logcheck-1.1.1# make install
Making
cc -O -o ./src/logtail ./src/logtail.c
src/logtail.c: In function `main':
src/logtail.c:51: warning: return type of `main' is not `int'
Creating temp directory /usr/local/etc/tmp
Setting temp directory permissions
chmod 700 /usr/local/etc/tmp
Copying files
cp ./systems//logcheck.hacking /usr/local/etc
cp: cannot stat `./systems//logcheck.hacking': No such file or directory
make: *** [install] Error 1
```

Se cio' capita (a me e' successo solo 1 volta) occorre modificare il Makefile, come vedete nella foto 2 [FOTO 2]. Cosa e' successo? Direi che la variabile che contiene il path giusto si e' azzerata, per cui il comando " cp " non puo' funzionare. Effettivamente la cosa e' strana.

```
Terminale
File Modifica Visualizza Terminale Schede Aiuto
UW PICO(tm) 4.9 File: Makefile-old

clean:
    /bin/rm ./src/logtail ./src/logtail.o

uninstall:
    /bin/rm $(INSTALLDIR_SH)/logcheck.sh
    /bin/rm $(INSTALLDIR)/logcheck.ignore
    /bin/rm $(INSTALLDIR)/logcheck.hacking
    /bin/rm $(INSTALLDIR)/logcheck.violations
    /bin/rm $(INSTALLDIR)/logcheck.violations.ignore
    /bin/rm $(INSTALLDIR_BIN)/logtail

install:
    @echo "Making $(SYSTYPE)"
    $(CC) $(CFLAGS) -o ./src/logtail ./src/logtail.c
    @echo "Creating temp directory $(TMPDIR)"
    @if [ ! -d $(TMPDIR) ]; then /bin/mkdir $(TMPDIR); fi
    @echo "Setting temp directory permissions"
    chmod 700 $(TMPDIR)
    @echo "Copying files"
    cp ./systems/$(SYSTYPE)/logcheck.hacking $(INSTALLDIR)
    cp ./systems/$(SYSTYPE)/logcheck.violations $(INSTALLDIR)
    cp ./systems/$(SYSTYPE)/logcheck.violations.ignore $(INSTALLDIR)
    cp ./systems/$(SYSTYPE)/logcheck.ignore $(INSTALLDIR)
    cp ./systems/$(SYSTYPE)/logcheck.sh $(INSTALLDIR_SH)
    cp ./src/logtail $(INSTALLDIR_BIN)
    @echo "Setting permissions"
    chmod 700 $(INSTALLDIR_SH)/logcheck.sh
    chmod 700 $(INSTALLDIR_BIN)/logtail
    chmod 600 $(INSTALLDIR)/logcheck.violations.ignore
    chmod 600 $(INSTALLDIR)/logcheck.violations

^G Get Help    ^O WriteOut    ^R Read File    ^V Prev Pg     ^K Cut Text     ^C Cur Pos
^X Exit        ^J Justify     ^W Where is    ^N Next Pg     ^U UnCut Text  ^T To Spell
```

Correggete il path e rilanciate il comando

```
make install
```

Ora abbiamo logcheck installato in /usr/local/etc ; sia l'eseguibile sia i files necessari.

Adesso e' il turno di hostsentry:

```
cd /tmp
tar zxvf hostsentry-0.02.tar.gz
cd hostsentry-0.02
make
make install
```

Hostsentry e' cosi' installato nella directory /usr/local/abacus/hostsentry. Continuiamo le installazioni con webmin:

```
cd /tmp
tar zxvf webmin-1.220.tar.gz
cd webmin-1.220
./setup.sh
```

seguite le istruzioni, e quando vi viene chiesto di attivare webmin all'avvio del pc rispondete di si.

Ci siamo quasi. Ora dobbiamo inserire il modulo che ci consente una gestione grafica dei tre software: portsentry, hostsentry e logcheck. Apriamo un browser e nella barra degli indirizzi immettiamo il seguente url:

http://pc_utilizzato:10000

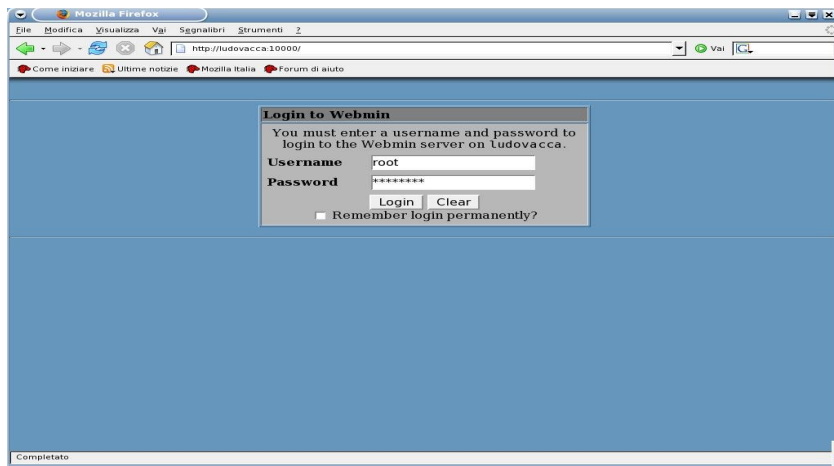
Ovviamente al posto di pc_utilizzato dovrete inserire l'indirizzo ip del computer utilizzato per le prove. Se tale pc e' lo stesso ove state lavorando (non siete quindi in una rete locale) l'url potrebbe essere il seguente:

<http://localhost:10000>

Si presentera' la maschera di login, come nella foto qui sotto. [FOTO 3]

Noi seguiremo il percorso

Webmin configuration



e successivamente

Webmin modules

Indichiamo nel campo di input il path esatto del file sentry.wbm e clicchiamo su

Install modules

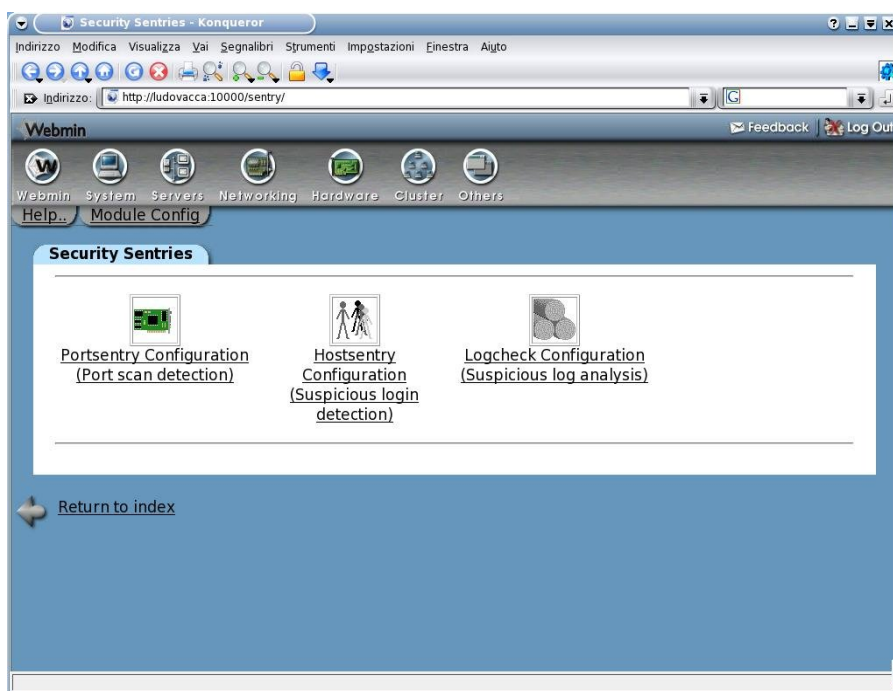
Non e' ancora finita!! Rechiamoci ora sulla sezione

System

Che si trova in alto, sulla sinistra, e poi cerchimo il tasto

Security Sentries

e gustimo il frutto del nostro lavoro: l'aspetto della schermata deve essere come in figura 4 [FOTO 4]



Un clic su

Portsentry Configuration

ci conduce alla schermata di configurazione. Correggete -se occorre- il path dell'eseguibile e cominciate gli esperimenti!!

COSA FA' IL SOFTWARE ?

Non ne abbiamo ancora parlato, occupati ad installare il tutto.

Portsentry e' un software che vigila sui port-scan a noi diretti. Quando incontra uno di questi port scan , ecco che identifica l'indirizzo ip da cui veniamo osservati, ed inserisce tale indirizzo ip nel file /etc/hosts.deny. Fine della faccenda. L'ip e' bloccato e non puo' piu' infastidirci. A questo punto fatevi una cultura sui files hosts.deny, host.allow, hosts.equiv e sul loro funzionamento. Portsentry e' abbastanza versatile e se lo lanciamo da riga di comando accetta le seguenti opzioni :

-tcp = tramite questo flag ecco che portsentry sorvegliera' le porte TCP indicate nel file /usr/local/psionic/portsentry/portsentry.conf

-udp = come sopra, ma con le porte UDP

-stcp = monitorizza le porte TCP indicate nel file di configurazione utilizzando un socket

-sudp = come sopra ma con le porte UDP

-atcp = (se ho capito bene il file README.install), tramite questa opzione portsentry legge il file di configurazione, e controlla gli scan dalla porta 1 fino al tetto indicato. Default = 1024. Viene consigliato di non oltrepassare tale limite, benché sia possibile arrivare a 65535. In pratica tutte le porte disponibili.

-audp = come sopra ma per UDP

Utilizzando le opzioni -atcp e -audp abbiamo la possibilita' di inserire nel file /usr/local/psionic/portsentry/portsentry.conf una lista di porte che non saranno sorvegliate. Lo scopo di tale esclusione e' impedire dei falsi positivi verso porte il cui utilizzo e' comune. Esempio posta, ssh, dns ecc.

Se vogliamo che un certo range di ip vengano esclusi dal controllo di portsentry, dobbiamo inserirli nel file portsentry.ignore. Ecco il contenuto di tale file dopo un'installazione standard :

```
# Put hosts in here you never want blocked. This includes the IP addresses
# of all local interfaces on the protected host (i.e virtual host, mult-home)
# Keep 127.0.0.1 and 0.0.0.0 to keep people from playing games.
#
# PortSentry can support full netmasks for networks as well. Format is:
```

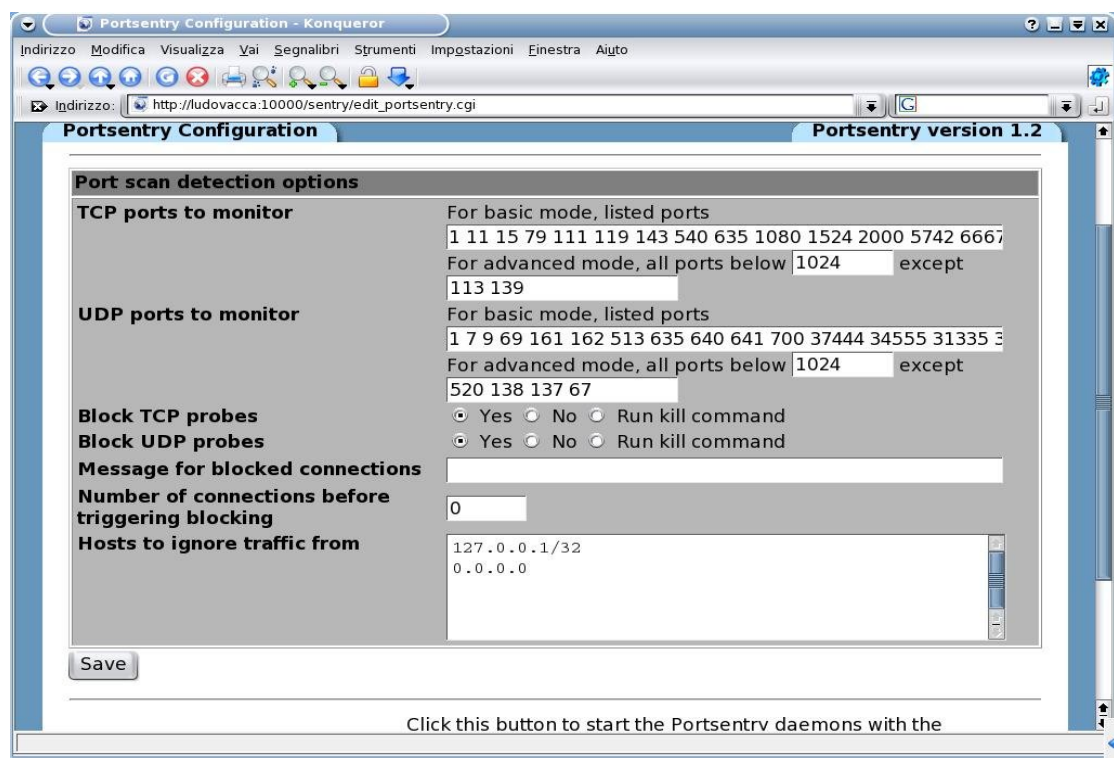
```

#
# <IP Address>/<Netmask>
#
# Example:
#
# 192.168.2.0/24
# 192.168.0.0/16
# 192.168.2.1/32
# Etc.
#
# If you don't supply a netmask it is assumed to be 32 bits.
#
#
127.0.0.1/32
0.0.0.0

```

Direi che e' opportuno inserire nel file gli indirizzi della vostra rete locale, altrimenti la prima volta che giocherellate con nmap vi troverete con meta' del parco pc bloccato.

La foto numero 5 mostra come settare portsentry usando webmin. Si vedono molto chiaramente i concetti qui sopra esposti.



Portsentry opportunamente configurato permette di visualizzare un banner sul pc dell'attacker. Resistere alla tentazione di coprirlo di insulti... Per fare cio' dovete toccare il file /usr/local/psionic/portsentry/portsentry.conf , precisamente nelle righe seguenti:

```

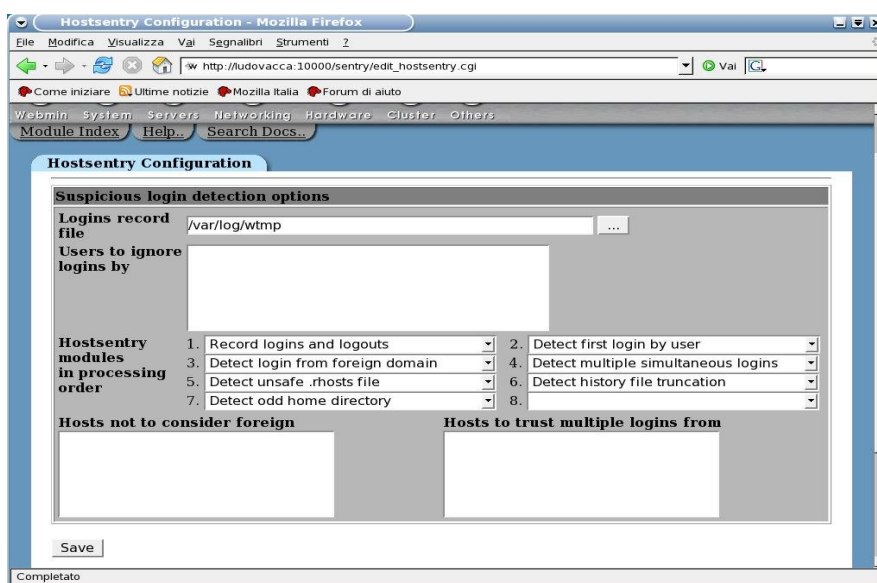
#####
# Port Banner Section#
#####
#
# Enter text in here you want displayed to a person tripping the PortSentry.

```

```
# I *don't* recommend taunting the person as this will aggravate them.
# Leave this commented out to disable the feature
#
# Stealth scan detection modes don't use this feature
#
#PORT_BANNER="** UNAUTHORIZED ACCESS PROHIBITED ** YOUR CONNECTION
#ATTEMPT HAS BEEN LOGGED. GO AWAY."
# EOF
```

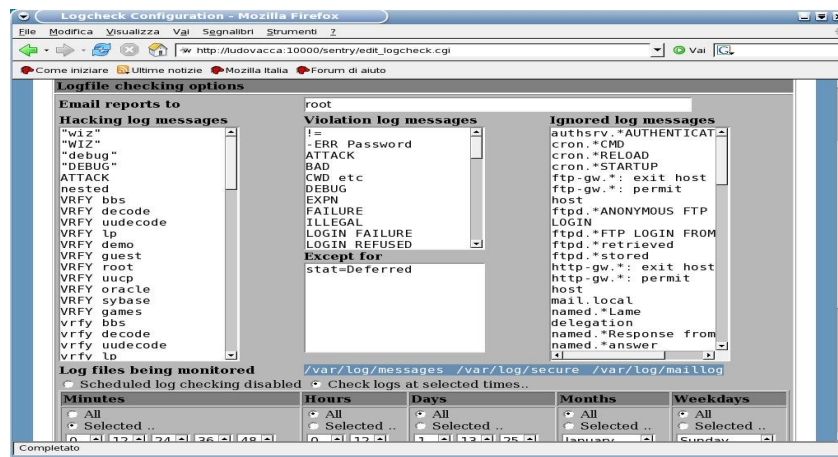
LOGCHECK E HOSTSENTRY

Le nostre attenzioni sono in maggior parte per portsentry, ma anche questi due programmini potranno dare molte soddisfazioni ad un admin attento. Cosa fanno? Hostsentry controlla il file /var/log/wtmp e notifica i login sospetti e gli account strani. Mi spiego meglio: Noi sappiamo che il ragioniere del 3° piano lavora da lunedì a venerdì dalle 9.00 alle 17.00. Come è possibile che si sia loggato sul server domenica alle 14.00? Ovviamente io ho enfatizzato la cosa, ma il succo è questo. Non ho guardato molto dettagliatamente hostsentry e logcheck, il lettore dovrà rimboccarsi le maniche e leggersi (quasi) tutto quello che trova nella directory /usr/local/abacus/hostsentry. Oppure può provare a governare il software utilizzando solo la grafica di webmin. In tal caso lo aspetta qualcosa di simile alla foto successiva [FOTO 6]



Come possiamo notare un campo attira subito la nostra attenzione: "Users to ignore logins by". Si tratta di quelle utenze di sistema che potrebbero generare dei falsi positivi. Vanno quindi esclusi dal controllo. Ovviamente ciò deve essere ponderato dall'admin.

I risultati della compilazione di logcheck sono invece finiti in /usr/local/etc. È qui che bisogna indagare per mettere in moto logcheck. Il suo compito è monitorare i files /var/log/messages /var/log/secure /var/log/maillog. [FOTO 7]



L'utilizzo corretto di logcheck -come viene indicato in /tmp/logcheck-1.1.1/install e' metterlo in cron, ed indicare un user come destinatario dei report. Tali report verranno consegnati sotto forma di posta elettronica nella consueta posizione /var/spool/mail/UTENTE_INDICATO. Possiamo quindi prelevare il tutto anche con un mail-reader qualunque.

HARDWARE DI TEST

Gli esperimenti sono stati condotti su un p2 350, 384 mb ram, hard disk ide 20 gb , 2 schede di rete realtek , lettore cd-rom lite on, scheda video ati rage 4 mb. Il tutto tramite rete locale, senza mai accendere il monitor del pc.