Percorso 7:
- Linux Advanced Administration

Fulvio Corno
fulvio.corno@polito.it

Bartolomeo Montrucchio
bartolomeo.montrucchio@polito.it

Ubuntu 12.04/14.04 LTS

- È una delle versioni più diffuse di Linux
- Si suggerisce sempre l'adozione delle versioni con supporto prolungato
- Esiste sia la versione Desktop (con interfaccia grafica di default) sia quella Server (priva di interfaccia grafica)
- Sono disponibili due macchine virtuali preconfigurate sia Desktop sia Server
- Virtual Box verrà utilizzato per gestire le installazioni e il dialogo tra le macchine

Virtual Box

- Ha diverse modalità di funzionamento
- Utilizzeremo la modalità NAT qualora ci serva la rete esterna e la modalità Host-Only per avere una rete strettamente locale
- Le macchine virtuali verranno importate come .ova

Virtual Box - NAT

- È come se la macchina virtuale fosse collegata all'esterno tramite un router (con NAT attivato)
- La macchina è irraggiungibile dall'esterno
- Si può fare sftp da guest a host, ma non viceversa
- Non si può far colloquiare tra di loro le macchine guest
- Va bene per navigare su Internet, ma richiede almeno il port forwarding per avere una buona utilità

Virtual Box – Host-only

- È un ibrido tra Bridged Networking e Internal Networking
- Tutte le macchine guest possono parlare tra di loro ed anche con l'host
- Viene usata una apposita interfaccia di loopback che può essere intercettata (solo internamente)
- Funziona bene anche SENZA una connessione fisica esterna
- Non permette connessione all'esterno
- Ma si potrebbero realizzare due reti, una Host-only privata (ad es. con web server e database) ed una Bridged (tra web server e mondo esterno)
  - in tal modo si potrebbe avere un elevato valore di sicurezza (il database è irraggiungibile dall'esterno)

Esercizio

- Creare una nuova macchina virtuale
- Effettuare l'installazione standard della 14.04 LTS in versione server come disponibile dalla ISO
- Provare a far dialogare tramite ssh la macchina appena installata con le precedenti macchine virtuali già presenti sul calcolatore

Webmin

- Permette la gestione di un server tramite interfaccia web (quindi in locale o in remoto)
- È in grado di gestire numerosi aspetti della macchina
- Utilizzare pcm (o pcm2) e relativa password per collegarsi dalla macchina pcm:
  - collegandosi alla porta 10000 si può modificare la configurazione della macchina anche in remoto

Esercizio

- Installare Webmin come da [4]
  - Webmin è già installato sulle macchine virtuali proposte, ma è bene reinstallarlo
- Provare a modificare i parametri di rete, anche in modo remoto
  - prestare attenzione a non chiudersi la via alla rete involontariamente

OpenBSD (5.5)

- Si tratta di un sistema BSD
- Molti settaggi sono diversi rispetto a Linux
  - in particolare i settaggi di rete
    - sh /etc/netstart per far ripartire il demone di rete
- Per spegnere la macchina shutdown –h now come di consueto
  - ma spegnere poi la macchina da Virtual Box

Esercizio

- Provare ad installare una macchina OpenBSD 5.5 utilizzando la ISO disponibile
- Individuare il partizionamento più adeguato del disco virtuale
- Verificare il corretto collegamento in rete
  - NAT
  - Host-only nella rete del laboratorio

OpenBSD (5.5) installazione pacchetti

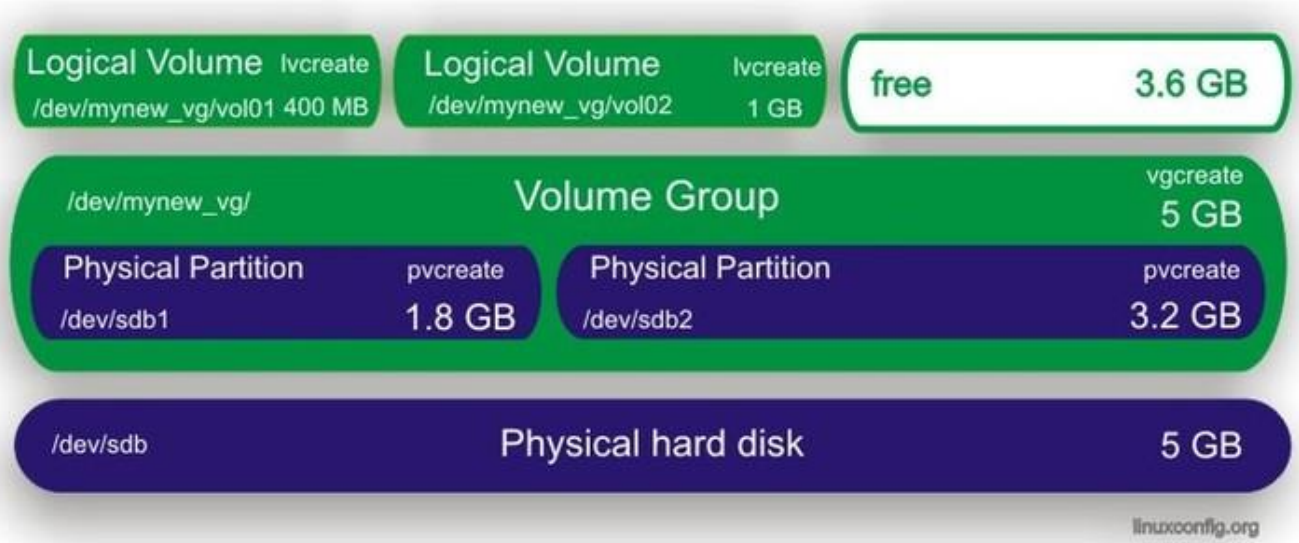
- pkg_add, pkg_delete, pkg_info
  - svolgere l'operazione da root
- Installazione da server ftp (qui per i386, altrimenti `machine –a` invece di i386)
  - $ export PKG_PATH=ftp://openbsd.mirror.garr.it/pub/OpenBSD/5.5/packages/i386/
  - pkg_add –i wget
  - pkg_add –i owncloud

Esercizio

- Provare ad attivare ed utilizzare Apache sulla macchina OpenBSD
  - usare lynx
  - Creare un piccolo file .html
- Per attivare il server Web:
  - httpd start in /etc/rc.d
  - controllare /etc/rc.conf per una attivazione persistente
- Provare ad installare (sola installazione) un sistema OAMP (OpenBSD/Apache/MySQL/PHP) (invece di LAMP, Linux) seguendo [3]
  - si tenga presente che con la versione 5.6 (Nov 2014) OpenBSD passerà da Apache a nginx come Web Server

LVM(1)

- Il Logical Volume Manager [5] serve a gestire i dischi con maggiore flessibilità
  - Si può per esempio estendere un disco già esistente in caso di spazio insufficiente



LVM(2)

- Creare le partizioni
  - fdisk per creare una o più partizioni fisiche
  - /dev/sdb1 e /dev/sdb2 in questo esempio
- Creare i volumi fisici
  - # pvcreate /dev/sdb1
  - # pvcreate /dev/sdb2
  - # pvdisplay
- Creare i gruppi virtuali (qui uno composto da due volumi fisici)
  - # vgcreate mynew_vg /dev/sdb1 /dev/sdb2
  - si possono anche estendere con # vgextend mynew_vg /dev/sdb2 (partendo da uno solo)

LVM(3)

- Creare i volumi logici
  - # lvcreate -L 400 -n vol01 mynew_vg
  - # lvcreate -L 1000 -n vol02 mynew_vg
  - #lvdisplay
  - #vgdisplay
- Creare i file system sui volumi logici
  - # mkfs.ext3 -m 0 /dev/mynew_vg/vol01

LVM(4)

- Preparare /etc/fstab con i dati richiesti

```
# /etc/fstab: static file system information.
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
proc            /proc           proc    defaults        0       0
/dev/sda1       /               ext3    defaults,errors=remount-ro 0       1
/dev/sda9       /home           ext3    defaults        0       2
/dev/sda8       /tmp            ext3    defaults        0       2
/dev/sda5       /usr            ext3    defaults        0       2
/dev/sda6       /var            ext3    defaults        0       2
/dev/sda7       none            swap    sw              0       0
/dev/hdc        /media/cdrom0   iso9660 ro,user,noauto  0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto  0       0
/dev/mynew_vg/vol01 /home/foobar ext3   defaults        0       2
~
~
~
~
~
~
~
~
                                                      13,65         All
```

LVM(5)

```
linuxconfig.org# mkdir /home/foobar
linuxconfig.org# mount -a
linuxconfig.org# cd /home/foobar/
linuxconfig.org# df -h .
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/mynew_vg-vol01
                      388M  8.1M  360M   3% /home/foobar
linuxconfig.org#
```

- Preparare la directory e montarla
- Estendere un volume logico
  - # lvextend -L +800 /dev/mynew_vg/vol01
  - # resize2fs /dev/mynew_vg/vol01

LVM(6)

- Rimuovere un volume logico
  - # lvremove /dev/mynew_vg/vol02
- lvdisplay serve poi a vedere che sia tutto in ordine

Esercizio

- Provare ad installare una macchina virtuale Linux desktop lasciando spazio libero sul disco
- Provare ad utilizzare LVM per creare e modificare un file system, seguendo quanto appena visto [6]
- Potrebbe essere necessario installare il package lvm2 con apt-get

Utenti

- /etc/passwd
- /etc/shadow
- comando passwd (bit setuid)
- shell
- partizione utenti
  - backup
  - directory utente, permessi
- gruppi e permessi
- adduser

Esercizio

- Provare ad aggiungere un utente in modo manuale in tutto e per tutto

Boot

- Loader del sistema operativo
  - Nei personal computer, esso è nell'MBR (primo settore del dispositivo di boot)
  - La dimensione limitata dell'MBR (512 byte) rende quasi impossibile installare un vero e proprio OS loader in esso
  - Viene dunque chiamato un loader secondario situato su di una partizione del disco
  - Normalmente si usa LILO oppure GRUB
  - Possono entrambi funzionare da loader secondario (richiamati da un MBR installato dal DOS) oppure da loader in due parti, completo di MBR+seconda parte dalla partizione di root
  - Il compito del loader è di trovare il kernel nel disco, caricarlo e lanciarlo, eventualmente passandogli dei parametri
  - Vi è solitamente anche un minimo di interfaccia

Rete

- La gestione della rete prevede:
  - settaggio dei parametri standard
  - verifica del livello di sicurezza
  - messa a punto del firewall

netstat

- Permette di vedere le connessioni presenti sul calcolatore
  - netstat –a
- Permette anche di vedere numerose altre cose, tra cui:
  - Le interfacce: netstat –i
  - Le mappe di routing: netstat -r

nmap

- È in grado di analizzare in modo remoto la configurazione di un computer
- Siccome può essere utilizzato anche per avere informazioni per un successivo attacco informatico il suo uso è da considerare con attenzione
  - ricevere una scansione con nmap senza esserne a conoscenza è da considerare un atto ostile
  - equivale al ladro che telefona o bussa per conoscere gli orari degli abitanti della casa
- La scansione deve essere il più possibile completa in termini di porte e di caratteristiche del SO
  - nmap -O -sS -p1-65535 xxx.xxx.xxx.xxx

Esercizio

- Utilizzando netstat individuare quali porte sono aperte sulla macchina
- Provare ad utilizzare nmap per testare quali porte sono aperte sulle singole macchine
  - lavorare solo tra macchine virtuali al fine di evitare scansioni erronee di macchine esterne (per i sistemisti è un atto ostile)
  - provare a vedere gli effetti della scansione sui file di log, per quanto possibile
- Provare ping su varie macchine, anche esterne

SSH

- Ssh, sftp, scp possono effettuare collegamenti e trasferimenti di file, anche non controllati direttamente dall'utente
- La porta utilizzata è la 22
- Si ricorda che la sintassi è del tipo
  - ssh nomeutente@nomemacchina.dominio.it
- Copiando la chiave pubblica (ad es. da /utente/.ssh/) della macchina da cui collegarsi in /utente/.ssh/authorized_keys della macchina in cui collegarsi l'autenticazione è automatica
  - per esempio usare ssh-keygen –t rsa –b 4096 senza passphrase
  - fare attenzione ai permessi dei file e delle directory

Esercizio

- Provare ad aprire una connessione ssh tra due macchine virtuali collegate tra loro in modalità Host-Only
- Verificare l'apertura della relativa porta all'innesco della connessione
- Provare a trasferire dei file mediante sftp
- Provare a svolgere il medesimo compito usando scp
- Provare a fare gli stessi esercizi senza l'utilizzo di password
- Provare a realizzare uno script bash che prenda come parametro da linea di comando un nome di file e copi quel file su di un'altra macchina senza dover esplicitamente inserire la password

rsync e i backup

```
#!/bin/sh
LOGFILE=/etc/rsync_backup/backup.log

echo "**********INIZIO**********" | tee -a >> $LOGFILE
date | tee -a >> $LOGFILE
echo "      **********" | tee -a >> $LOGFILE

/usr/local/bin/rsync -avc -progress -e ssh /home/utente root@129.191.12.16:/backup/destinazione 2>&1 | tee -a >> $LOGFILE

echo "      **********" | tee -a >> $LOGFILE
date | tee -a >> $LOGFILE
echo "**********FINE**********" | tee -a >> $LOGFILE

# alla fine manda in uscita su stdout la fine del file di log per
# poterla spedire via e-mail
/usr/bin/tail $LOGFILE
```

crontab

01 23 * * * /etc/rsync_backup/backup.rsync | mail -s "`date` macchina.polito.it" bartolomeo.montrucchio@polito.it

Esercizio

- Replicare quanto illustrato nelle due slide precedenti
- Gestire anche la spedizione della mail
  - NAT?

Firewall

- I firewall hanno lo scopo di controllare e restringere il passaggio di dati a diversi livelli dello stack ISO/OSI
- Possono lavorare a livello 2 (come gli switch) bloccando i MAC address
- Oppure a livello 3 (come i router e gli switch layer 3)
- Oppure a livello 4 (TCP/UDP)
- Oppure a livello applicazione
- Si noti che nella rappresentazione TCP/IP i livelli sono differenti!

Netfilter

- In Linux Netfilter è un framework per la manipolazione dei pacchetti
- Funziona tramite dei “ganci” interni allo stack del protocollo desiderato (es. IPv4). Il kernel può registrarsi per esaminare il pacchetto prima che venga mandato (eventualmente) avanti.
- Le possibili azioni sul pacchetto esaminato sono [5]:
  - NF_ACCEPT: continua la traversata normalmente.
  - NF_DROP: scarta il pacchetto; non continuare la traversata.
  - NF_STOLEN: ho prelevato il pacchetto; non continuare la traversata.
  - NF_QUEUE: accoda il pacchetto (di solito per la gestione in userspace).
  - NF_REPEAT: chiama di nuovo questo hook.

IP tables

- Al di sopra del framework netfilter è stato realizzato un sistema di selezione dei pacchetti in transito, iptables (al momento incluso di default nella maggior parte delle distribuzioni Linux)
- iptables è di fatto un programma a linea di comando (userspace)
- iptables gestisce anche il NAT ed esiste anche per IPv6 (ip6tables)
- Per iptables sono state realizzate numerose interfacce, sia testuali sia grafiche
- iptables gestisce un certo numero di tabelle (tables appunto) ciascuna contenente un certo numero di chains, ciascuna delle quali contiene delle regole
  - qui vedremo la tabella filter

iptables - rules

- iptables lavora confrontando il traffico di rete con un insieme di regole (rules) [7]
- Ogni regola definisce le caratteristiche che un pacchetto deve avere per soddisfare quella regola e l'azione da intraprendere per i pacchetti che la soddisfano
- Per la regola ci si può basare su:
  - Tipo di protocollo
  - Porta sorgente o destinazione
  - Interfaccia di rete
  - Relazione con precedenti pacchetti
  - etc…

iptables - azione

- I pacchetti che soddisfano la regola sono soggetti ad un'azione
- L'azione (chiamata target) può essere:
  - accept
  - drop
  - Il pacchetto viene spostato ad un'altra chain (gruppi di regole)
  - semplicemente effettuare il log del pacchetto

iptables – chain

- Una catena è un insieme di regole (zero o più) nei cui confronti il pacchetto viene verificato (in modo sequenziale)
- ATTENZIONE: quando un pacchetto in arrivo soddisfa una delle regole nella catena, la relativa azione viene eseguita e le successive regole nella catena vengono ignorate
- Possono essere create nuove catene
- Ci sono tre chains definite di default nella tabella filter (quella usata di default):
  - INPUT: gestisce tutti in pacchetti in ingresso
  - OUTPUT: gestisce I pacchetti in uscita
  - FORWARD: gestisce i pacchetti in transito; di fatto gestisce un routing
- Ogni catena ha una policy di default, che definisce cosa accade al pacchetto se non soddisfa nessuna delle regole; può essere di:
  - DROP (pacchetto scartato)
  - ACCEPT (pacchetto accettato)

iptables – connessioni

- iptables può anche tenere traccia delle connessioni
- Si possono creare regole per definire come comportarsi con un pacchetto sulla base della sua correlazione con i pacchetti precedenti
- Si parla di state tracking o connection tracking o state machine

iptables - riassunto

- Riassumendo iptables:
  - Manda il pacchetto alla catena appropriata
  - Confronta il pacchetto con ogni regola (in ordine dalla prima della catena) finché non avviene che il pacchetto soddisfa una di tali regole
    - In tal caso si ferma con l'applicazione di quella regola
  - Se nessuna regola può essere applicata, la policy di default viene considerata
- Se la policy di default è drop è importante prendere precauzioni per mantenere le connessioni (ad es. ssh) attive
- L'ordine delle regole nella catena è importante:
  - Prima devono esserci le regole più specifiche
  - Poi le più generali, fino alla policy di default se nessuna regola è valida
- Se la policy di deafult è ACCEPT le regole effettueranno il drop dei pacchetti
- Se la policy di default è DROP le regole della catena conterranno eccezioni per I pacchetti da accettare

iptables – esempi(1)

- iptables va utilizzato avendo i permessi di root
- iptables –L mostra la lista delle regole correnti (con --line-numbers l'elenco delle regole è numerato per comodità) [8]
- iptables –S mostra i comandi necessari per abilitare le regole e le policy correnti
  - per replicare la configurazione corrente basta replicare le varie linee
- Se si è collegati in remoto si presti attenzione ad eventuali policy di DROP di default che potrebbero fermare la connessione in corso
- Iptables –F cancella le regole in corso, ma non le policy di default delle chains
  - per cui nuovamente attenzione ad eventuali policy di DROP che fermerebbero le connessioni
  - dare magari prima:

    iptables -P INPUT ACCEPT

    iptables –P OUTPUT ACCEPT

iptables – esempi(2)

- Per accettare esplicitamente la connessione ssh corrente (regola molto specifica, quindi all'inizio):

  iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

  - -A INPUT aggiunge una regola al fondo della catena di INPUT
  - -m conntrack attiva il modulo aggiuntivo conntrack di iptables
    - --ctstate è uno dei comandi del modulo conntrack e permette di agganciare i pacchetti sulla base di come sono correlati con i pacchetti già visti in precedenza
      - ESTABLISHED aggancia i pacchetti che sono parte di una connessione già esistente
      - RELATED aggancia i pacchetti di una nuova connessione correlata alla connessione stessa
  - -j ACCEPT indica che i pacchetti appena selezionati vanno accettati

iptables – esempi(3)

- Per mantenere aperte le porte 22 e 80 (regole meno specifiche della precedente, quindi da porre dopo):

  sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT

  sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT

  - -p tcp aggancia il protocollo TCP (connection-based)
    - --dport opzione di –p tcp per indicare il numero di porta (22 oppure 80)
- -j ACCEPT indica che i pacchetti appena selezionati vanno accettati

iptables – esempi(4)

- Per garantire il passaggio dei pacchetti sull'interfaccia di loopback (regole più specifica delle precedenti, quindi da porre prima):

  sudo iptables -I INPUT 1 -i lo -j ACCEPT

  - -I INPUT 1 inserisce una regola in una posizione (qui 1), non la aggiunge in coda; la posizione 1 indica la posizione più specifica
  - -i lo indica l'interfaccia di loopback
- -j ACCEPT indica che i pacchetti appena selezionati vanno accettati

iptables – esempi(5)

- Siccome tutti i pacchetti che non soddisfano le regole che ci siamo poste vanno cancellati, si può:
- Modificare la policy di default di INPUT (qui non abbiamo visto OUTPUT o FORWARD)
  - sudo iptables -P INPUT DROP
- Oppure, per evitare di perdere la connessione a causa della policy di default in caso di cancellazione erronea delle regole, si può lasciare la ACCEPT come policy e aggiungere una regola ALLA FINE della catena (è la regola più generale)

  sudo iptables -A INPUT -j DROP

  tutti i pacchetti rimanenti vengono quindi cancellati, pur mantenendo la policy di default ad ACCEPT (altre regole aggiunte andrebbero però inserite poi prima di quest'ultima)

Esempio completo

- root@pcm:~# iptables –S
- -P INPUT DROP
- -P FORWARD ACCEPT
- -P OUTPUT ACCEPT
- -N INBOUND
- -N LOG_FILTER
- -N LSI
- -N LSO
- -N OUTBOUND
- -A INPUT -i lo -j ACCEPT
- -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
- -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
- -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

iptables – esempi(6)

- iptables-save produce su stdout le regole correnti e iptables-restore le reinserisce (tutte le regole in un'unica volta, non come se si facesse iptables molte volte)
- iptables-apply applica regole da un file (prodotto con iptables-save), ma chiede conferma (con timeout)
- al termine del time-out ripristina il vecchio settaggio, in modo da evitare problemi se i nuovi settaggi sono errati e la connessione si perde applicandoli
- Le regole aggiunte vanno perse facendo ripartire il server
  - può essere un modo per fare prove in sicurezza

iptables – esempi(7)

- Al fine di applicare le regole al boot si può [9]:
  - con iptables-save salvare in un file la configurazione

  iptables-save > /etc/iptables.rules
  - creare uno script (ad es. vim /etc/network/if-up.d/loadiptables) del tipo:

```
#!/bin/bash
/sbin/iptables-restore < /etc/iptables.rules
exit 0
```

- In questo modo la configurazione del firewall verrà caricata al boot

Esercizio

- Provare a riprodurre il precedente esempio
- Collaudare i collegamenti usando ssh

firestarter

- È una interfaccia ad iptables molto comoda, grafica
- Lo sviluppo è sospeso
- Prestare attenzione alla attività del firewall dopo aver chiuso la finestra di firestarter

Esercizio

- Verificare utilizzando Firestarter se nella configurazione della macchina virtuale desktop fornita il firewall è attivo e cosa blocca
  - provare con iptables -L
- Utilizzando Firestarter provare a bloccare il ping tramite ICMP
- Provare poi a bloccare specifiche porte

ufw - gufw

- È un front-end per iptables (anche in forma grafica, gufw)
- È studiato per semplificare le configurazioni più semplici
- Supponendo di essere root:
  - ufw allow ssh/tcp abilita l'access ssh
  - ufw logging on abilita il logging
  - ufw enable abilita il firewall
  - ufw status mostra lo stato del firewall

ufw - gufw

- ufw allow 22 apre la porta dell'ssh
- ufw deny 22 chiude la porta dell'ssh
- ufw disable disabilita il firewall
- ufw allow proto tcp from 192.168.0.2 to any port 22 permette accesso ssh dall'host 0.2; rimpiazzare 192.168.0.2 con 192.168.0.0/24 permetterebbe accesso ssh dall'intera sottorete

Esercizio

- Provare con gufw a bloccare sia servizi sia applicazioni
- Verificare il tutto tramite connessioni da un'altra macchina virtuale

fwbuilder

- Presenta una interfaccia molto sofisticata
- I firewall sviluppati con esso possono funzionare anche con altro hardware/software oltre ad iptables

Esercizio

- Provare a costruire un semplice firewall usando fwbuilder
- Utilizzare anche il manuale d'uso

pf

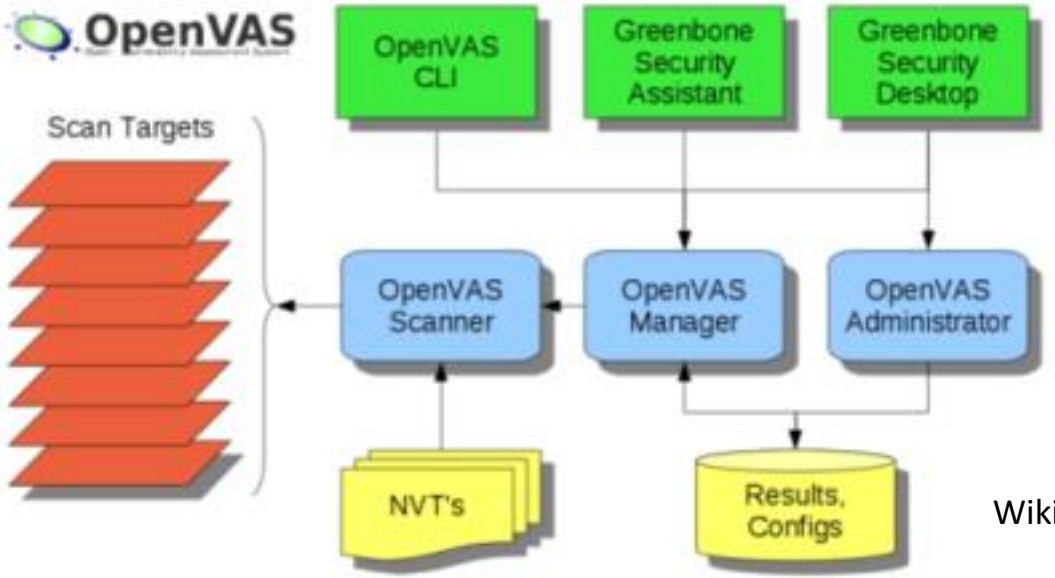
- È il firewall presente in OpenBSD
- È estremamente potente, anche se complicato da usare (interfaccia basata su file di testo)
- fwbuilder può però gestire anche pf
- pf è il firewall presente in tutti i Mac
  - IceFloor è una valida interfaccia grafica per controllarlo comodamente
  - pf permette anche di settare un massimo di accessi prima di respingere le connessioni (utile per port-scan al fine di rigettare le ricognizioni con nmap)

Tool per la sicurezza delle reti

- In [1] è possibile trovare un elenco esaustivo dei tool utilizzati per il test della sicurezza delle reti
- I principali sono:
  - Nessus: ora a pagamento (fino al 2005 era open source)
  - OpenVAS: open source, proveniente da Nessus
  - Core Impact: molto costoso, il più efficace
  - Nexpose
  - GFI LanGuard
  - QualysGuard
  - MBSA, prodotto dalla Microsoft
  - Retina
  - Secunia PSI
  - Nipper
  - SAINT
- La soluzione migliore è probabilmente quella di utilizzare più di un prodotto

OpenVAS

- Deriva da Nessus
- Le recensioni sono talvolta contrastanti
  - È in generale molto buono
- È disponibile una macchina virtuale di prova già configurata



Wikipedia OpenVAS

Esercizio

- Provare ad installare l'appliance di OpenVAS da:
  - http://www.openvas.org/vm.html
- ATTENZIONE: assicurarsi di essere in modalità Host-Only
- Attivare il gestore collegandosi da remoto via Web (trovare l'indirizzo IP al boot della macchina Virtual Box)
  - L'utente da utilizzare è admin, password admin
- In totale servono dunque tre macchine virtuali
  - La macchina OpenVAS
  - La macchina con il client Web (es. Firefox)
  - La macchina da testare
- Provare ad attaccare una macchina (per es. la macchina OpenBSD) ed analizzare il report
  - provare più di una modalità di attacco (più o meno accurata/lenta)

Bibliografia

- [1] http://sectools.org/tag/vuln-scanners/
- [2] http://www.openbsd.org/faq/faq15.html#PkgInstall
- [3] http://www.h-i-r.net/p/hirs-secure-openbsd-apache-mysql-and.html
- [4]http://wiki.ubuntu-it.org/Server/Webmin
- [5] http://www.netfilter.org/documentation/HOWTO/it/netfilter-hacking-HOWTO-3.html
- [6] http://linuxconfig.org/linux-lvm-logical-volume-manager
- [7] https://www.digitalocean.com/community/tutorials/how-the-iptables-firewall-works
- [8] https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-iptables-on-ubuntu-14-04
- [9] http://terraltech.com/saving-iptables-rules-to-be-persistent/#.VAjRU41vYQU