Percorso 5:
- Network Configuration/IP Tables

Bartolomeo Montrucchio
bartolomeo.montrucchio@polito.it

Giovanni Squillero
giovanni.squillero@polito.it

Sistema operativo e rete

- All'interno del sistema operativo è implementato lo stack TCP/IP
- La rete viene configurata via software
- Il firewall può essere implementato via software
- Iptables è il firewall di riferimento in Linux (presente nel kernel)
- La gestione è svolta ai vari livelli ISO/OSI – TCP/IP

Il modello ISO/OSI(1)

- Strutturato a strati
  - Application: provvede una interfaccia utente
  - Presentation: presenta i dati e gestisce trattamenti quali l'encryption
  - Session: mantiene i dati di differenti applicazioni separati
  - Transport
  - Network
  - Data link
  - Physical

Il modello ISO/OSI(2)

- Strutturato a strati
  - Application
  - Presentation
  - Session
  - Transport:
    - gestisce la consegna (affidabile o non affidabile)
    - effettua l'error correction prima della ritrasmissione
  - Network:
    - fornisce l'indirizzamento logico usato dai routers per l'individuazione del percorso
  - Data link:
    - combina pacchetti in byte e i byte in frame
    - fornisce accesso ai media tramite l'indirizzo MAC
    - fornisce l'error detection (non l'error correction)
  - Physical
    - sposta i bit tra i dispositivi
    - specifica voltaggio, velocità di comunicazione e piedinatura dei cavi

Il modello ISO/OSI(3)

- Application
  - File, stampa, messaggi, database e servizi applicativi (SOLO DATI)
- Presentation
  - Data encryption, compressione e traduzione (SOLO DATI)
- Session
  - Controllo della comunicazione (SOLO DATI)
- Transport:
  - Connessione end-to-end (SEGMENTO – PDU (Protocol Data Unit))
- Network:
  - Routing (PACKET o DATAGRAM – PDU)
- Data link:
  - Framing (FRAME – PDU)
- Physical
  - Topologia fisica (BITS – PDU)

TCP/IP – modello DOD (Department of Defense)

- Process/Application corrisponde a Application + Presentation+Session
- Host to Host corrisponde a Transport
- Internet corrisponde a Network
- Network Access corrisponde a Data Link + Physical

Protocolli

- I principali protocolli di nostro interesse (lato firewall) sono:
- Process/Application: telnet, ftp, tftp, nfs,smtp,lpd,X Window, SNMP, DNS, BooP,DHCP
- Host to Host: TCP (Transmission Control Protocol), UDP (User Datagram Protocol)
- Internet: IP (Internet Protocol), ICMP (Internet Control Message Protocol ad es per ping), ARP (Address Resolution Protocol parte da un IP destinazione per averne il MAC), RARP (Reverse ARP parte da MAC per averne IP)
- Network Access: Ethernet, Fast Ethernet, Token Ring, FDDI

Domini di collisione

- I router rompono i domini di broadcast
- Gli switch e i bridge rompono i domini di collisione
- I Firewall risentono di questi domini nelle loro possibilità di intervento

Porte TCP e UDP

- Numeri di porta tipici sono:
  - FTP 21
  - ssh 22
  - telnet 23
  - Doom 666
  - DNS 53
  - TFTP 69
  - POP3 110
  - News 144
- UDP non ha il controllo di flusso (i pacchetti in eccesso saranno persi e non ritrasmessi) e non crea un circuito virtuale
- Le porte al di sotto della 1024 sono well-known
- Al di sopra della 1024 sono usate per differenziare le sessioni con i vari host

DNS

- Domain Name System traduce i nomi in indirizzi IP
- Ha una struttura articolata su più livelli
- Ha un parziale equivalente nel WINS
  - Quali porte utilizza netbios per i suoi servizi gestiti in rete locale?
- Su di una rete locale sarebbe necessario avere un DNS specifico

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

ping

- Permette di interrogare, utilizzando ICMP, una macchina remota
  - Utilizzare il nome della macchina se possibile (?)
- È possibile variare la dimensione del pacchetto (ping of death)
- Indica anche il tempo richiesto al pacchetto per arrivare a destinazione
- Spesso è disattivato per ragioni di sicurezza

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

NAT e IP Masquerading

- Il NAT permette di presentarsi all'esterno (con un indirizzo IP pubblico) pur avendo solo un indirizzo IP privato
  - Gli indirizzi IP privati sono visibili solo all'interno della rete locale (ad es. in casa a valle del modem/router ADSL)
  - Tutte le connessioni vengono perciò rimappate
- Nell'IP Masquerading non solo l'indirizzo viene rimappato, ma anche le porte TCP/UDP, tramite una tabella presente nel router

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 – NAT Network

- A patto di creare una apposita rete interna a Virtual Box, permette di:
  - Far dialogare tra di loro le macchine all'interno della rete anche senza connessione fisica esterna
  - Far dialogare le macchine verso l'esterno con TCP e UDP con IP v4 e v6
- Le macchine non sono però raggiungibili dall'esterno e i guest non possono parlare con l'host
- Virtual Box può fornire anche un servizio DHCP, come negli altri casi peraltro

Virtual Box – Bridged networking

- A patto di non avere firewall attivati è possibile far colloquiare tra di loro i guest e i guest con l'host e viceversa
- Gli indirizzi IP devono essere tutti diversi e compatibili con quelli della rete locale esterna (no indirizzi privati interni)
- E però richiesta una connessione valida a livello fisico (portante) ad una rete anche locale, in quanto i pacchetti escono e poi rientrano dopo il passaggio nello switch esterno
  - In caso contrario non funziona nulla

Virtual Box – Internal network

- Simile a Bridged Networking, ma i pacchetti non devono più uscire dall'interfaccia fisica (evidenti vantaggi di sicurezza) sulla quale potrebbero essere intercettati da uno sniffer (es. Wireshark)
- I Guest possono parlare solo tra di loro
- È utile per ragioni di sicurezza

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) [6]

Esercizio

- Verificare la configurazione corrente della macchina virtuale
- Provare a modificare i parametri della macchina virtuale
- Quali parametri sono modificabili?
- Verificare il server DHCP di VirtualBox e provare a disattivarlo

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

Indirizzi IP(v4)

- 32 bit in blocchi da 8
- L'indirizzo di broadcast contiene tutti 255 nella parte di host, che è di 3 byte nella classe A, 2 byte nella classe B e un byte nella classe C (le classi D ed E sono a parte)
- A: i 32 bit cominciano con 1_2, quindi da 0 a 127 come primo byte; 16777214 nodi per ciascuna delle 126 reti
- B: i 32 bit cominciano con 10_2, quindi da 128 a 191 come primo byte; 65534 nodi per ciascuna delle 16384 reti
- C: I 32 bit cominciano con 110_2, quindi da 192 a 223; 254 nodi per ciascuna delle 2097152 possibili reti

Sottoreti

- Una parte dei bit di host vengono usati per creare sottoreti
- Le subnet mask standard sono:
  - A 255.0.0.0
  - B 255.255.0.0
  - C 255.255.255.0
- Ad es. Una classe B con subnet mask pari a 255.255.255.128 avrà 510 sottoreti, ciascuna con 126 host
- L'indirizzo di broadcast è quello in cui, data una sottorete, i bit di host sono posti ad uno

Indirizzi IP privati

- Indirizzi IP privati:
  - classe A: 1 singola classe A da 10.0.0.0 - 10.255.255.255 per circa 16 milioni di host 10.0.0.0/8 (255.0.0.0)
    24 bit di host - 8 bit di maschera
  - classe B: 16 classi B da 172.16.0.0 – 172.31.255.255 per circa un milione di host 172.16.0.0/12 (255.240.0.0)
    20 bit di host - 12 bit di maschera
  - classe C: 256 classi C da 192.168.0.0 – 192.168.255.255 per circa 65000 host 192.168.0.0/16 (255.255.0.0)
    16 bit di host - 16 bit di maschera

DHCP e parametri di rete

- Il server Dynamic Host Configuration Protocol provvede a fornire alla macchina l'indirizzo IP
- È una sorta di BootP dynamico, in quanto non ha bisogno di conoscere a priori l'indirizzo hardware (pur non distribuendo anche il sistema operativo come nel BootP)
- Il DHCP fornisce:
  - indirizzo IP
  - subnet mask
  - dominio
  - default gateway (router)
  - DNS
  - WINS
- Essi sono i parametri da impostare nella macchina, a mano o con un tool

ifconfig

- È in grado di configurare una interfaccia di rete
  - Può attivarla e disattivarla
- Per gli scopi di questo corso serve soprattutto a vedere la configurazione:

root@pcm:~/.ssh# ifconfig -a eth0

eth0 Link encap:Ethernet HWaddr 08:00:27:85:ff:f1

inet addr:192.168.56.150 Bcast:192.168.56.255 Mask:255.255.255.0 inet6 addr: fe80::a00:27ff:fe85:fff1/64 Scope:Link

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

RX packets:322 errors:0 dropped:0 overruns:0 frame:0

TX packets:456 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000

RX bytes:50637 (50.6 KB) TX bytes:50015 (50.0 KB)

System settings
gnome-control-center(1)

- unity-control-center a partire dalla versione 14.04
- È in grado di settare parecchi parametri, tra cui la rete, in modo semplice
- È anche in grado di creare e gestire utenti
- I parametri di rete vanno inseriti per la scheda desiderata

Settaggio manuale parametri di rete

- In /etc/NetworkManager/NetworkManager.conf portare managed a true
- In /etc/network/interfaces settare:

  Auto eth0

  Iface eth0 inet static

  address xxx.xxx.xxx.xxx

  netmask xxx.xxx.xxx.xxx

  network xxx.xxx.xxx.xxx

  broadcast xxx.xxx.xxx.xxx

  gateway xxx.xxx.xxx.xxx
- Fare reboot per disabilitare il Network Manager
- Dopo basterà fare /etc/init.d/networking restart

Esercizio

- Cambiare l'indirizzo IP con l'interfaccia grafica
- Verificare con ifconfig e tramite ssh che tutto funzioni
- Provare a cambiare l'indirizzo in modo statico
- Provare anche a cambiare la sottorete, se necessario anche in Virtual Box
  - quali problemi si incontrano?
- Infine riattivare il DHCP

Routing

- Il routing è usato per veicolare pacchetti da una rete all'altra
- Può essere:
  - Statico (programmato dal sistemista di rete sul router)
  - Dinamico (il router ricava le informazioni comunicando con i router delle reti adiacenti)
    - Distance vector (RIP, IGRP)
    - Link state (shortest path first, OSPF)
    - ibrido

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)

Bibliografia

- https://wiki.ubuntu.com/
- http://wiki.ubuntu-it.org
- http://help.ubuntu-it.org/
- http://free-electrons.com/docs/
- [5] http://www.netfilter.org/documentation/HOWTO/it/netfilter-hacking-HOWTO-3.html
- [6] http://technology.amis.nl/2014/01/27/a-short-guide-to-networking-in-virtual-box-with-oracle-linux-inside/
- [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