ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
DIPARTIMENTO DI
INGEGNERIA DELL'ENERGIA ELETTRICA E
DELL'INFORMAZIONE
"GUGLIELMO MARCONI"

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA PER L'ENERGIA E L'INFORMAZIONE

TECNICHE DI ATTACCO E DIFESA PER SISTEMI FIREWALL

Elaborato in
Laboratorio di reti e programmazione dispositivi mobili

Relatore
Walter Cerroni

Presentata da
Andrea Fato

Anno Accademico 2017/2018

INDICE

ABSTRACT

L’obiettivo principale di questa tesi è di analizzare alcuni software e tecniche di sicurezza informatica. In particolare vengono descritti alcuni firewall, fra cui il packet filter, e il comando iptables che serve per la realizzazione del firewall stesso.

Uno degli argomenti principali dell’elaborato è il port knocking, esso è un sistema per modificare dinamicamente le regole del firewall tramite l’invio di tentativi di connessione a una sequenza prestabilita di porte chiuse. In particolare viene mostrato come configurare tale servizio di port knocking e vengono fatti alcuni esempi di cattiva configurazione che renderebbero tale servizio poco efficiente.

Nella tesi vengono inoltre descritte alcune tecniche di attacco, fra cui l’attacco DoS e DDoS, e alcuni metodi per mitigare tali tentativi di danneggiamento del sistema.

La parte finale dell’elaborato mostra esempi pratici di attacco e difesa del sistema. In particolare viene mostrato come configurando in maniera intelligente il firewall sia possibile difendere il sistema dai tentativi di intrusione o di danneggiamento da parte di un utente esterno.

1. INTRODUZIONE

In un mondo sempre più digitalizzato e connesso, la protezione dei dati diventa un problema di vitale importanza.
Il tema della sicurezza sul web coinvolge certamente tutti i settori interessati dal processo di digitalizzazione, ma anche nel nostro piccolo la protezione dei dati è di cruciale importanza per la nostra privacy.
La diffusione di internet implica un aumento progressivo di utenti nella rete, ma fra essi non tutti hanno buone intenzioni. Infatti negli ultimi anni il numero di attacchi informatici è decisamente aumentato, uno dei dati del Rapporto Clusit 2018, Associazione Italiana per la Sicurezza Informatica, riporta 122 attacchi gravi in media al mese nel primo semestre del 2018.

Dunque non è affatto raro oggigiorno sentire parlare di Cyber security, o sicurezza informatica, data la sua rilevanza.
Ma cos'è veramente la Cyber security?

Esistono molteplici definizioni di sicurezza informatica, di seguito ne vengono riportate alcune.

Secondo l'Enciclopedia Treccani si può definire la Cyber security come l'insieme delle tecniche e dei dispositivi, sia software sia hardware, mediante i quali si attua la protezione di dati e sistemi informatici.
In particolare gli aspetti principali della sicurezza informatica riguardano la difesa delle informazioni dai tentativi di intrusione a scopo di spionaggio o di danneggiamento dell'intero sistema, e la salvaguardia della loro integrità.

Una seconda definizione viene da NIST, Computer Security Handbook [NIST95], secondo cui la definizione di sicurezza informatica riguarda la protezione di sistemi informatici nella loro integrità, disponibilità e nella riservatezza delle informazioni in essi contenute.

In particolare quest'ultima definizione introduce tre concetti chiave della sicurezza informatica:

- **Riservatezza**
  – Riservatezza dei dati: Assicurarsi che le informazioni private e/o confidenziali non siano disponibili.
  – Privacy: Assicurarsi che chi ha il controllo delle informazioni di privati non le divulghi.
- **Integrità**
  – Integrità dei dati: Assicurarsi che le informazioni vengano modificate solo da chi ha l'autorizzazione per poterlo fare.
  – Integrità del sistema: Assicurarsi che le funzionalità del sistema non vengano alterate da chi non ha l'autorizzazione per poterlo fare.
- **Disponibilità**: Assicurarsi che il sistema funzioni e che non vengano negati i servizi alle persone autorizzate.

Questi tre concetti fondamentali sono le fondamenta di base per la sicurezza dei dati e delle informazioni nei sistemi informatici.

Esistono dunque diverse tecniche di difesa, fra esse la cosiddetta "security through obscurity" (letteralmente sicurezza attraverso l'oscurità). In particolare uno degli argomenti che verranno trattati in questa tesi sarà il Port knocking, il cui scopo è proprio proteggere un servizio attivo sul nostro terminale "nascondendolo" agli utenti non autorizzati ad accedervi. Un altro metodo per proteggere il sistema dalle minacce della rete è l'utilizzo di firewall, in particolare verrà trattato il packet filter che è appunto un firewall il cui scopo è filtrare i pacchetti che lo attraversano in modo da far passare solo quelli autorizzati.

Una strategia utilizzata per difendere il sistema è trovarne le vulnerabilità e una volta individuate attuare un metodo di difesa opportuno.
Lo scopo di questa tesi sarà infatti analizzare diverse configurazioni di firewall, trovarne le "falle" e capire come risolverle, in modo da rendere il nostro sistema il più protetto possibile da eventuali attacchi esterni.
Per prima cosa si andranno a studiare gli strumenti, in questo caso software, a nostra disposizione, poi si analizzeranno alcuni tipi di attacchi e infine alcuni esempi di contromisure utilizzate per mitigare tali tentativi di intrusione o di danneggiamento del nostro sistema.

Prima di iniziare, però, è doveroso dire che la sicurezza informatica non sarà mai totale.
Ai miglioramenti delle tecniche anti-hacking corrisponde il parallelo progresso delle tecniche di hacking.
Dunque non esiste un sistema di sicurezza definitivo, il quale una volta implementato renda il nostro computer sicuro per sempre.

2. FIREWALL

2.1 Tipi di firewall

Un firewall è un dispositivo per la sicurezza della rete che permette di monitorare il traffico in entrata e in uscita mediante una serie di regole impostate per consentire o bloccare il passaggio dei dati.
In particolare il firewall agisce da “filtro software” che serve a proteggere da accessi indesiderati provenienti dall'esterno della rete.

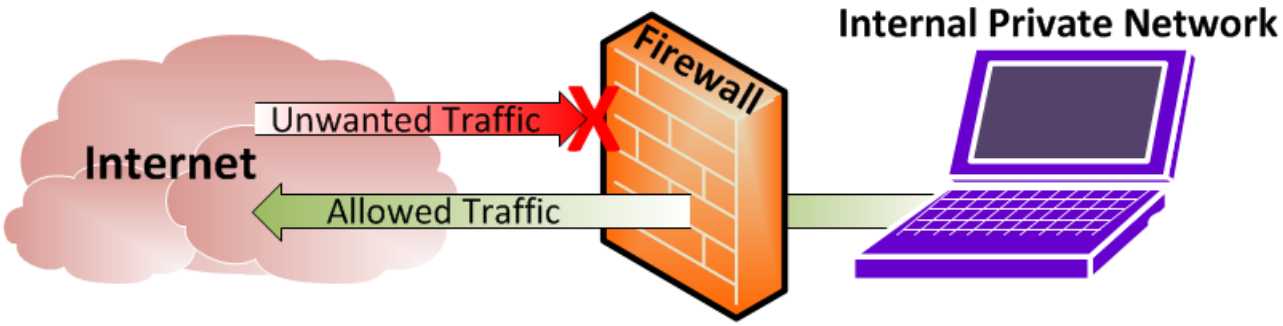


Fig.1] Rappresentazione grafica di un firewall [1]

Dunque solo il traffico autorizzato deve attraversare il firewall, e i servizi di rete ritenuti necessari per il funzionamento del sistema devono essere mantenuti.

Esistono diversi tipi di firewall classificati in base al loro utilizzo e funzionamento, di seguito ne verranno descritti alcuni.

1. **Packet filter**

- Fa riferimento alle intestazioni del pacchetto, come l’indirizzo IP e l’interfaccia di ingresso/uscita. In base a queste effettua un vero e proprio filtraggio del traffico.

- È situato fra la rete locale e internet.

- Filtra e scarta i datagrammi IP da trasferire sulle varie interfacce sulla base di:
  -interfaccia di riferimento e/o destinazione.
  -indirizzo MAC e/o IP di sorgente e di destinazione.
  -tipo di servizio.

2. Firewall proxy

- Funge da gateway fra le reti per una specifica applicazione.
  I server proxy possono offrire funzionalità aggiuntive come il caching e la protezione dei contenuti che impediscono connessioni dirette dall'esterno della rete.
  Tuttavia questa soluzione può avere ripercussioni sulla velocità di trasmissione e sulle applicazioni supportate.

- L'accesso alla rete esterna è consentito solo attraverso il server proxy.

- A differenza del Packet filter, il quale si limita a monitorare il traffico di rete, il firewall proxy interrompe il canale di comunicazione diretta tra due strumenti connessi e la riavvia prendendo il posto del mittente.

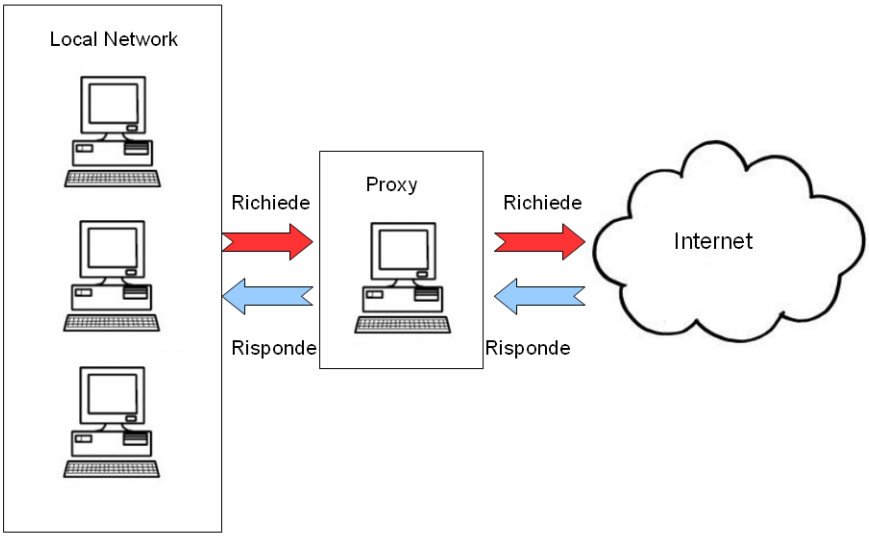


Fig. 2] Rappresentazione grafica di un firewall proxy [2]

3. Next-Generation Firewall (NGFW)

- Sono firewall di nuova generazione che non si limitano più al filtraggio dei pacchetti. La maggior parte delle aziende utilizza soluzioni Next-Generation Firewall per bloccare le minacce più recenti come il malware avanzato e gli attacchi a livello di applicazione.
- Funzionalità firewall standard, come la stateful inspection.
- Prevenzione delle intrusioni integrata.
- Riconoscimento e controllo delle applicazioni per individuare e bloccare le app pericolose.
- Percorsi di aggiornamento per includere feed di informazioni futuri.
- Tecniche per far fronte alle minacce alla sicurezza in costante evoluzione.

2.2 Packet filter

Nel corso di questo elaborato si farà riferimento alla configurazione del firewall come packet filter, in quanto tutt'ora si trovi nella maggior parte dei prodotti con firewall e routers ed è facilmente modificabile via software.
Questo tipo di firewall filtra le informazioni basandosi sull'header del livello di protocollo di rete. Il filtraggio si basa sull'**ACL** (Lista di Controllo degli Accessi) che decide il traffico permesso in una specifica rete, sia in ingresso sia in uscita.
I criteri su cui si basa tale lista sono i seguenti.

- Indirizzo IP sorgente e di destinazione.
- Numeri di porta sorgente e di destinazione.
- Tipo di protocollo.
- Interfaccia di ingresso e/o uscita.

Durante la comunicazione di rete, il pacchetto viene abbinato a delle regole specifiche. Una volta abbinato il pacchetto viene accettato o rifiutato a seconda che soddisfi o meno tali regole.
Di norma il controllo si basa sugli indirizzi IP di sorgente e destinazione, se entrambi gli indirizzi corrispondono il pacchetto viene considerato sicuro. Poiché il mittente può utilizzare diverse applicazioni e programmi, il filtraggio dei pacchetti controlla anche i protocolli di origine e destinazione, come UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). I filtri controllano anche il numero di porta della sorgente e della destinazione e le interfacce di ingresso e uscita.

Alcuni packet filter possono essere inoltre configurati in modalità **stateful**. Il firewall configurato in questa modalità tiene traccia dello stato di transizione dei pacchetti finché ogni specifica connessione non viene chiusa.

Lo stateful firewall controlla tutte le conversazioni tra i dispositivi connessi utilizzando una ***state table*** (tabella di stato) che ne schematizza i risultati. In questo modo se un pacchetto in ingresso o in uscita è relativo a una connessione già iniziata precedentemente e non ancora terminata esso viene automaticamente lasciato passare senza dover ripetere i controlli. L'abbinamento alle regole della lista viene fatto solo sul primo pacchetto, se esso supera i controlli allora tutti i pacchetti successivi e relativi alla stessa connessione sono accettati.
Il vantaggio di tale configurazione del firewall sta nel fatto che viene tenuta traccia di alcune relazioni fra i pacchetti che lo attraversano, dunque, ad esempio, è in grado di ricostruire lo stato delle connessioni TCP. Questo può permettere di riconoscere pacchetti TCP "malevoli" che non fanno parte di alcuna connessione.
Dopo l'ispezione iniziale dei pacchetti il firewall si limita al controllo della rete e delle porzioni di trasporto dell'intestazione: i valori di ogni header vengono comparati all'interno della state table e quest'ultima viene aggiornata per rendere conto del processo della comunicazione.

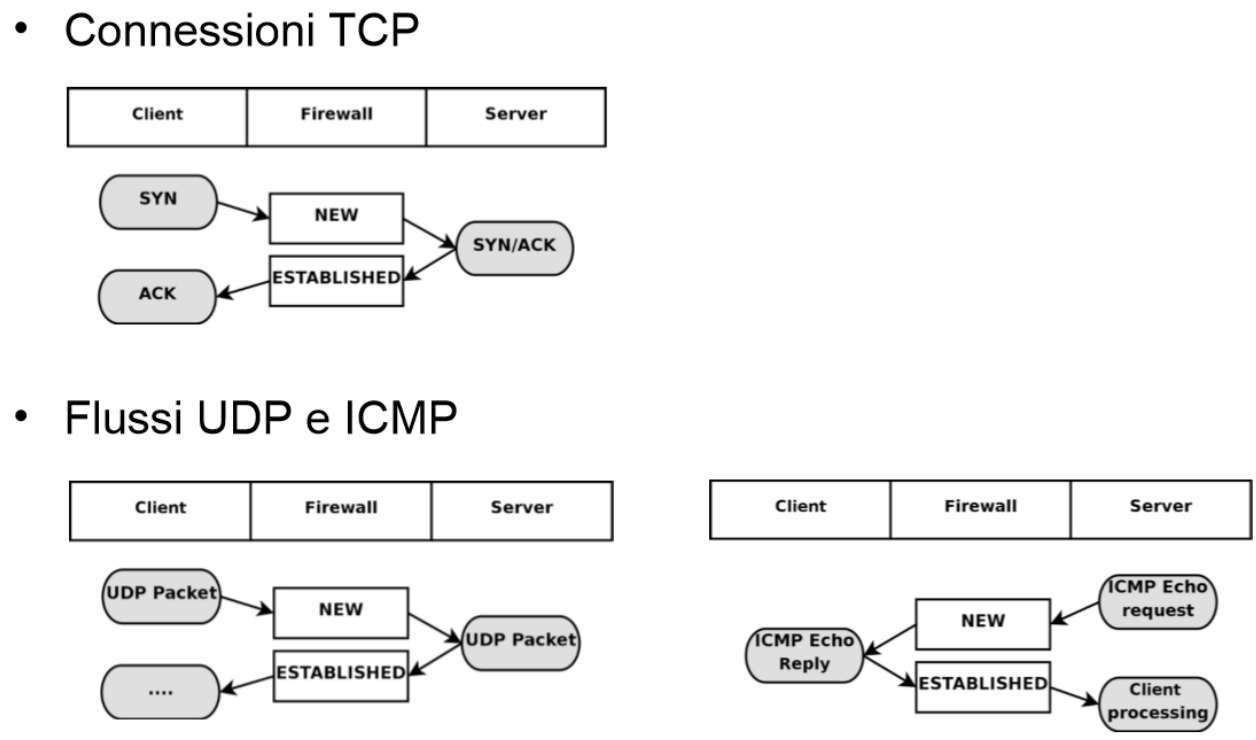


Fig. 3] Rappresentazione delle connessioni in uno stateful firewall

La Fig. 3] mostra la gestione degli stati di flussi/connessioni in uno stateful firewall impostato tramite il comando **iptables**. Tale comando verrà trattato nel seguito in quanto è lo strumento che verrà utilizzato per la configurazione del firewall.

In conclusione il packet filter è in genere una difesa efficace contro gli attacchi dei computer esterni a una rete locale (LAN).
Poiché la maggior parte dei dispositivi di routing ha funzionalità di filtro integrate, il filtraggio dei pacchetti è considerato un mezzo di sicurezza standard ed economico.

2.3 Vulnerabilità del packet filter

Come preannunciato nell'introduzione, il metodo migliore per difendersi è conoscere le proprie debolezze.

In questo paragrafo si vanno dunque ad elencare le principali debolezze del packet filter descritto nel paragrafo precedente.

Le principali vulnerabilità del packet filter sono le seguenti:

- L'impossibilità di prevenire degli attacchi che sfruttino specifiche vulnerabilità delle applicazioni.
- Molti packet filtering non riescono a rilevare ***IP address spoofing***, ovvero pacchetti inviati da falsi indirizzi IP.
- Potrebbero non essere in grado di rilevare degli ***IP fragmentation attacks***, che sono una sorta di attacco DDoS in cui viene frammentato un datagramma (o pacchetto) in multipli e più piccoli pacchetti "intossicando" il network.

Queste elencate sono le principali vulnerabilità del firewall in questione, tali debolezze possono facilmente essere sfruttate da un utente malevolo per causare danni al nostro sistema.
Nel capitolo successivo verrà analizzata la configurazione del firewall, in particolare si farà riferimento al comando iptables con il quale si potrà impostare il firewall in modo da far fronte alle problematiche sopra elencate.

3. CONFIGURAZIONE DEL FIREWALL

3.1 Default policy

È buona norma quando si configura il firewall impostare per prima cosa la ***default policy***, ovvero la politica che si decide di applicare a tutti quei servizi che non vengono esplicitamente permessi o negati.
In particolare esistono due tipi di default policy.

1. **Default deny**: tutti i servizi non esplicitamente permessi sono negati.

2. **Default allow**: tutti i servizi non esplicitamente negati sono permessi.

In particolare si farà riferimento alla default deny, ovvero alla politica che blocca tutto il traffico di default e fa passare solo il traffico specifico ai servizi noti.
Questa strategia offre un buon controllo del traffico e riduce la possibilità di una violazione a causa di un errore di configurazione del servizio.

Ciò che si fa dunque è impostare come ultima regola del firewall quella che nega tutto il traffico. Dunque tutti i servizi che non soddisfano le regole precedenti all'ultima, ovvero i servizi non esplicitamente permessi, verranno negati.
È evidente che servirà impostare delle regole specifiche che autorizzino il traffico che si vuole far passare.

3.2 Consentire il traffico specifico

La strategia utilizzata è il così detto ***principio del privilegio minimo***, ovvero le regole che si utilizzano per definire l'accesso alla rete devono essere le più specifiche possibili.
Nel capitolo precedente si è visto su quali parametri il packet filter basa il filtraggio, dunque specificare nelle regole che autorizzano il traffico il maggior numero di questi parametri fa si che esse siano le più selettive possibili e dunque rendano il firewall meno vulnerabile.
Di seguito si riportano i parametri su cui si basa il packet filter.

- Indirizzo IP sorgente e di destinazione.
- Numeri di porta sorgente e di destinazione.
- Tipo di protocollo.
- Interfaccia di ingresso e/o uscita.

Se il servizio deve essere accessibile a tutti su internet, allora qualsiasi indirizzo IP sorgente deve essere autorizzato e dunque nella regola non occorrerà specificare tale IP. In tutti gli altri casi, invece, è necessario specificare l'IP sorgente per evitare accessi da utenti non autorizzati.

L'indirizzo IP di destinazione è quello del server che esegue il servizio a cui si desidera consentire l'accesso. È dunque necessario specificare a quali server è possibile accedere in modo da evitare problemi futuri.

La porta di destinazione corrisponde invece al servizio a cui si vuole accedere. Consentire l'accesso a qualsiasi porta è un grave errore, solo la porta relativa al servizio che si vuole rendere disponibile deve essere accessibile.

3.3 Il comando iptables

In un sistema Linux le configurazioni di cui si è parlato nel paragrafo precedente possono essere fatte tramite iptables.
Iptables è un comando che agisce a livello del kernel e serve per realizzare il firewall stesso. Tale comando consente agli amministratori di gestire il traffico in entrata e in uscita tramite una serie di regole di cui si è già accennato nei paragrafi precedenti.
In particolare il comando iptables:

- Implementa funzionalità di stateful packet filter nei kernel di Linux 2.4 e successivi.
- Lavora a livello di kernel e ha il controllo dei pacchetti IP in transito sulle interfacce di rete.

I pacchetti IP processati da iptables sono soggetti a diverse modalità di elaborazione chiamate ***table***, ciascuna delle quali è composta da gruppi di regole dette ***chain***.

In particolare ci sono 4 table principali:

1. FILTER, per il filtraggio dei pacchetti.
2. NAT, per la sostituzione dell'IP.
3. MANGLE, per la manipolazione dei pacchetti.
4. RAW, per le eccezioni di configurazione.

Le funzionalità di firewall vere e proprie sono implementate dalla table FILTER.

Nella table FILTER sono presenti 3 chain predefinite.

- INPUT; contiene le regole di filtraggio da usare sui pacchetti in arrivo al firewall e destinati all'host locale.
- OUTPUT; contiene le regole di filtraggio da usare sui pacchetti in uscita dall'host locale.
- FORWARD; contiene le regole di filtraggio da usare sui pacchetti in transito nel firewall.

È inoltre possibile creare ulteriori chain oltre a quelle predefinite.

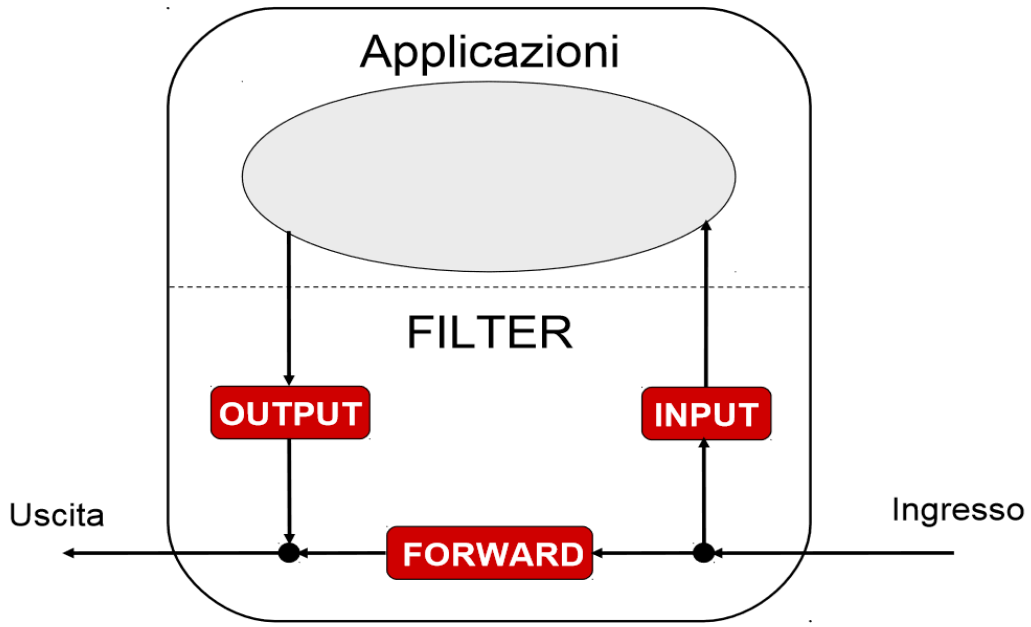


Fig. 4] Rappresentazione grafica della table filter di Iptable

La Fig. 4] mostra una rappresentazione della table FILTER e delle sue chain. Una regola può stabilire se scartare con DROP, rifiutare con un messaggio con REJECT o accettare con ACCEPT un pacchetto in ingresso/uscita al firewall.
Se un pacchetto non soddisfa nessuna regola viene applicata la default policy di quella chain.

3.4 Opzioni di Iptables

Innanzitutto è possibile visualizzare le regole in uso da ogni chain della table filter digitando:

```
iptables -L <chain-name> -t <table-name>
```

-L elenca tutte le regole contenute nella catena specificata dopo il comando. Per ottenere un elenco di tutte le regole di tutte le catene contenute nella tabella di default, FILTER, basta non specificare alcuna catena o tabella.

Per impostare la default policy della chain occorre digitare:

```
iptables -P <chain-name> <policy>
```

-P Imposta la policy di default per una determinata catena, così se un pacchetto arriva alla fine di una catena e nessuna delle regole è stata soddisfatta, il pacchetto stesso viene inviato al target specificato, come ad esempio ACCEPT o DROP.

Per aggiungere una nuova regola alla catena occorre digitare:

```
iptables -I <chain-name> <N> <rule specs> -j <policy>
```

-I Inserisce una nuova regola in un determinato punto specificato da un valore intero, N, definito dall'utente. Se non viene specificato alcun numero, iptables inserirà il comando all'inizio della catena.

-A Aggiunge la regola iptables alla fine della catena specificata. Questo comando viene utilizzato per aggiungere semplicemente una regola in coda alle altre.

Le **rule specs** servono per rendere la regola specifica, dunque sono dei parametri fondamentali:

```
-i <interface>                  -o <interface>
```

-i per specificare l'interfaccia di ingresso.

-o per specificare l'interfaccia di uscita.

```
-s <address>/<netmask>          -d <address>/<netmask>
```

-s per specificare l'indirizzo IP sorgente.

-d per specificare l'indirizzo IP di destinazione.

```
-p tcp|udp|icmp
```

-p per specificare il tipo di protocollo IP nella regola.

```
--sport <port>                  --dport <port>
```

Queste due opzioni possono essere inserite correttamente solo dopo avere specificato il tipo di protocollo con -p.

--sport per specificare la porta sorgente.

--dport per specificare la porta di destinazione.

È inoltre possibile configurare il firewall come stateful packet filter, in questo modo è possibile specificare pacchetti di connessioni TCP o flussi UDP nuovi o già attivi.
Come anticipato nel capitolo 2.2, questo è un modo utile per tenere traccia delle connessioni/flussi.
In particolare iptables definisce i seguenti stati di flussi/connessioni:

- NEW; relativo a un pacchetto appartenente a un flusso/connessione non presente nella *table conntrack*.
- ESTABLISHED; associato a flussi/connessioni dei quali sono già stati accettati i pacchetti precedenti, in entrambe le direzioni.
- RELATED; associato a flussi/connessioni non established ma correlati a flussi/connessioni established.
- INVALID; associato a pacchetti con uno stato non tracciabile.
- UNTRACKED; associato a pacchetti non soggetti alla modalità stateful (table raw*)*.

Dunque in una regola si può stabilire lo stato di una nuova connessione/flusso aggiungendo:

```
- m state –state NEW
```

Per fare si che i pacchetti di una connessione/flusso successivi al primo vengano accettati senza dover ripetere i controlli per ognuno di essi, basta aggiungere come prima regola:

```
iptables -I <chain-name> 1 -m state --state ESTABLISHED -j
ACCEPT
```

È inoltre molto utile visualizzare lo stato delle connessioni/flussi tramite il comando:

```
CONNTRACK –L
```

3.5 Configurazione di un semplice firewall

In questo paragrafo verrà mostrato un primo esempio di configurazione di un firewall molto semplice, ma in grado di offrire una buona protezione al proprio personal computer.

Si considera un terminale in cui si vuole configurare un firewall tale che ogni tentativo di accesso da parte di una rete esterna venga rifiutato.
Si considera inoltre che il terminale debba rendere disponibili i servizi SSH (porta 22 tcp), HTTP (porta 80 udp) e un servizio di filesharing (porta tcp 4662 e udp 4672).

Per prima cosa si imposta la default policy per la chain INPUT e per la chain FORWARD:

```
sudo iptables -P INPUT DROP
```

```
sudo iptable -P FORWARD DROP
```

Occorre consentire tutto il traffico interno al PC che passa per l'interfaccia di loopback (lo), dunque il successivo comando da digitare è:

```
sudo iptables -A INPUT -i lo -j ACCEPT
```

È inoltre necessario consentire tutto il traffico richiesto dal terminale, in questo modo è possibile accedere a internet:

```
sudo iptables -A INPUT -m state --state ESTABLISHED,
RELATED -j ACCEPT
```

I prossimi comandi che andranno inseriti sono relativi ai servizi che si vogliono rendere disponibili:

```
sudo iptables -A INPUT -p tcp --dport 22 -m state --state
NEW -j ACCEPT
```

```
sudo iptables -A INPUT -p udp --dport 80 -m state --state
NEW -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp --dport 4662 -m state --state
NEW -j ACCEPT
```

```
sudo iptables -A INPUT -p udp --dport 4672 -m state --state
NEW -j ACCEPT
```

```
iptables -I INPUT 1 -m state --state ESTABLISHED -j ACCEPT
```

I comandi utilizzati in questo esempio sono gli stessi descritti nel paragrafo precedente. Si può inoltre osservare che il firewall è stato configurato in modalità stateful, infatti l'ultima regola inserita (che in realtà viene messa in prima posizione) dice che tutti i pacchetti relativi a una connessione già stabilita devono essere automaticamente accettati.

4. PORT KNOCKING

4.1 Cos’è il Port knocking

Spesso una grande debolezza del server è il fatto che quest’ultimo per poter offrire dei servizi deve lasciare delle specifiche porte aperte per consentire agli altri utenti della rete di accedervi.
A volte, però, il servizio offerto non deve essere accessibile a tutti, basti pensare per esempio al sevizio SSH attraverso il quale si può accedere a un server da remoto.
Dunque nel caso di servizi riservati a uno stretto numero di host si potrebbe pensare di aprire le relative porte solo quando questi utenti autorizzati vogliono usufruire del servizio corrispondente e lasciarle chiuse il resto del tempo.
Lasciare queste porte sempre aperte è infatti rischioso, chiunque potrebbe accedervi. Esistono numerosi metodi di ***port scanning*** per individuare le porte aperte in un host della rete e il corrispondente servizio attivo.

7	ECHO
9	DISCARD
13	DAYTIME
17	QUOTD (Quote of the day)
19	CHARGEN (Character generator)
20	FTP-DATA (FTP data transfer)
21	FTP (File Transfer Protocol)
22	SSH (Secure Shell)
23	TELNET
25	SMTP (Simple Mail Transfer Protocol)
42	WINS (Windows Internet Naming Service)
53	DNS (Domain Name Server)
69	TFTP (Trivial File Transfer Protocol)
79	FINGER
80	HTTP (Hyper Text Transfer Protocol)
110	POP3 (Post Office Protocol 3)
113	IDENT/AUTH
119	NNTP (Network News Transfer Protocol)
135	EPMAP (DCE Endpoint Mapper)
137	NETBIOS-ns (name service)
138	NETBIOS-dgm (datagram service)
139	NETBIOS-ss (session service)
143	IMAP (Internet Message Access Protocol)
161	SNMP (Simple Network Management Protocol)
389	LDAP (Lightweight Directory Access Protocol)
443	HTTPS (Secure HTTP)
445	Microsoft-ds (Microsoft Directory Service)

Fig. 5] Lista porte del protocollo TCP-IP

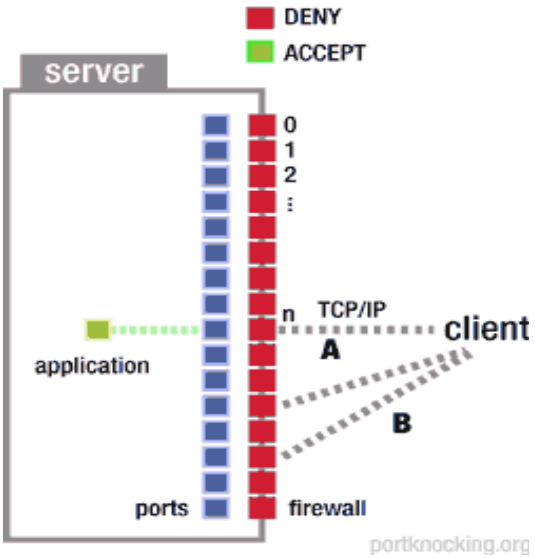
La Fig. 5] mostra un elenco di alcune delle porte del protocollo TCP-IP. Si può osservare che ogni servizio ha una corrispondente porta di default, tali porte sono dette “Well known ports”.
È però possibile assegnare a un servizio una porta diversa da quella di default, in modo da rendere quel servizio meno immediato da trovare per gli utenti non autorizzati. Nonostante ciò, come annunciato precedentemente, esistono numerosi metodi in grado di rilevare i servizi attivi su un host e le relative porte, anche se non sono quelle preimpostate.

Il ***Port knocking*** è un metodo per “oscurare” i servizi che sono in esecuzione sul proprio computer. Tale metodo consente al firewall di proteggere i servizi fino a quando non si chiede che la relativa porta venga aperta attraverso una sequenza specifica di traffico di rete.
Il Port knocking non si limita a cambiare la porta di default del servizio, ma grazie ad esso la porta risulterà chiusa a tutti gli utenti non autorizzati. I servizi che si intende nascondere saranno accessibili solo da chi conosce una determinata sequenza di porte da “bussare”, da cui il nome Port knocking. Infatti solo una volta completata la sequenza la porta relativa al servizio desiderato verrà aperta e quest’ultimo sarà accessibile.

Il Port knocking funziona configurando un servizio per guardare i log del firewall o le interfacce di acquisizione dei pacchetti per i tentativi di connessione. Se viene eseguita una sequenza specifica di tentativi di connessione predefiniti (o "colpi"), il servizio modificherà le regole del firewall per aprire le connessioni su una determinata porta.

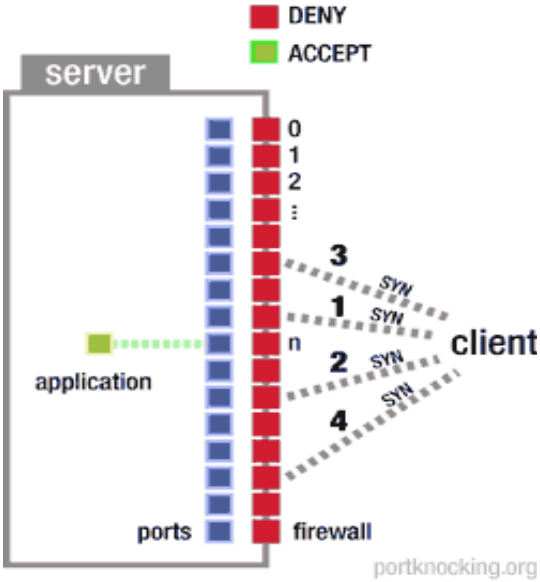
Questa tecnica aggiunge un ulteriore livello di sicurezza per i servizi che si vuole proteggere, ma non è l’unica contromisura adottata. Infatti se un utente riuscisse a completare la sequenza correttamente, sarà anche sottoposto alla autenticazione regolare (identificativo e password).
Inoltre un altro vantaggio di tale metodo è che non viene fornito alcun feedback sui tentativi effettuati per completare la sequenza, dunque chi tenta di accedere al servizio nascosto non saprà mai se la sequenza emessa sia anche solo in parte quella corretta, a meno che non ne sia già a conoscenza.
Questo è spesso sufficiente per dissuadere o vietare gli aggressori.

Gli steps fondamentali del Port knocking sono i seguenti;



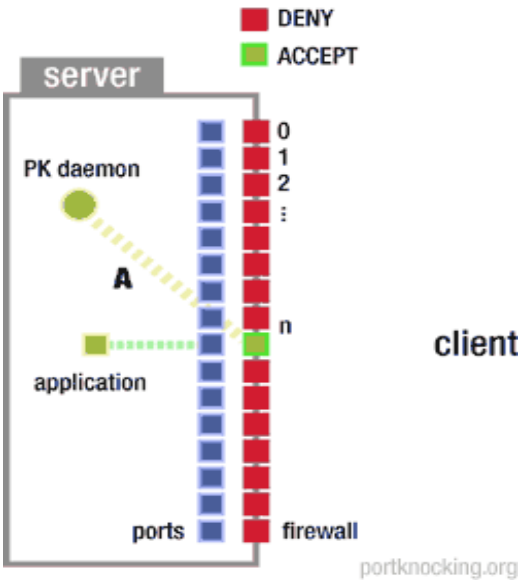
[3]

Step 1] Un utente client non può accedere alla applicazione in ascolto dietro la porta n. Lo stesso client non può accedere a nessuna porta protetta dal firewall.



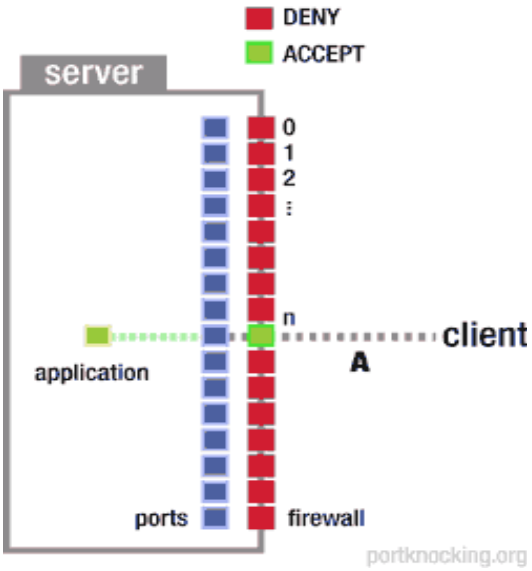
[3]

Step 2] Un utente client si connette a un insieme ben definito di porte in sequenza inviando pacchetti SYN. In questo caso il client è a conoscenza del demone di Port knocking e della sua configurazione, ma non riceve alcun riconoscimento in questa fase perché le regole del firewall precludono ogni risposta.



[3]

Step 3] Il processo del server (un demone di Port knocking) intercetta i tentativi di connessione e li decodifica interpretandoli come un “knock knock” autentico.
A questo punto il server esegue un'attività specifica in base al contenuto del knockout della porta, ad esempio aprendo la porta n al client.



[3]

Step 4] Il client adesso può accedere alla porta n e dunque si connette ad essa. Il client viene sottoposto alla regolare procedura di autenticazione in atto su tale applicazione.

4.2 I limiti del Port Knocking

la tecnica di Port knocking descritta nel paragrafo precedente ha delle limitazioni che la rendono impraticabile in alcuni casi.
Si consideri per esempio un server http, è evidente che per tale servizio si vorrebbero avere delle connessioni attive in qualsiasi momento, dunque in questo caso non è consigliabile applicare il Port knocking per tale servizio.
In casi come quello appena descritto il Port knocking fallirebbe, perché non ha senso nascondere un servizio che si vuole rendere pubblico, occorrerebbe rendere nota a tutti la sequenza di porte e ciò renderebbe il Port knocking pressoché inutile.
Inoltre tale metodo da solo non è un buon meccanismo di sicurezza, infatti il Port knocking si occupa solo di nascondere il servizio e non di proteggerlo. Chiunque sia a conoscenza, in modo legittimo o meno, della giusta sequenza di porte potrà accedere al servizio, infatti il Port knocking non imposta nessun meccanismo di autenticazione.

In conclusione, tale tecnica risulta efficace ogni volta che si desidera aggiungere un ulteriore livello di sicurezza a un servizio destinato a un ristretto numero di utenti noti, e che richiedono quel particolare servizio con una bassa frequenza. Mentre risulta inefficace ogni volta che il servizio in questione è destinato a un gran numero di utenti, se non addirittura a tutti gli utenti della rete.

4.3 Configurazione del servizio Knockd

Il servizio Knockd è ciò che implementerà la tecnica di Port knocking descritta precedentemente.
Per installarlo occorrerà digitare:

```
sudo apt-get install knockd
```

Questo comando installerà l'utility, ma non avvierà il servizio per impostazione predefinita.
È necessario configurare e abilitare manualmente questo servizio.

Per configurare il servizio, occorre modificare il file di configurazione knocked.conf. Per far ciò occorre aprire il file knockd.conf con i privilegi di root:

```
sudo nano /etc/knockd.conf
```

Ciò che si dovrebbe vedere è qualcosa di simile a questo:

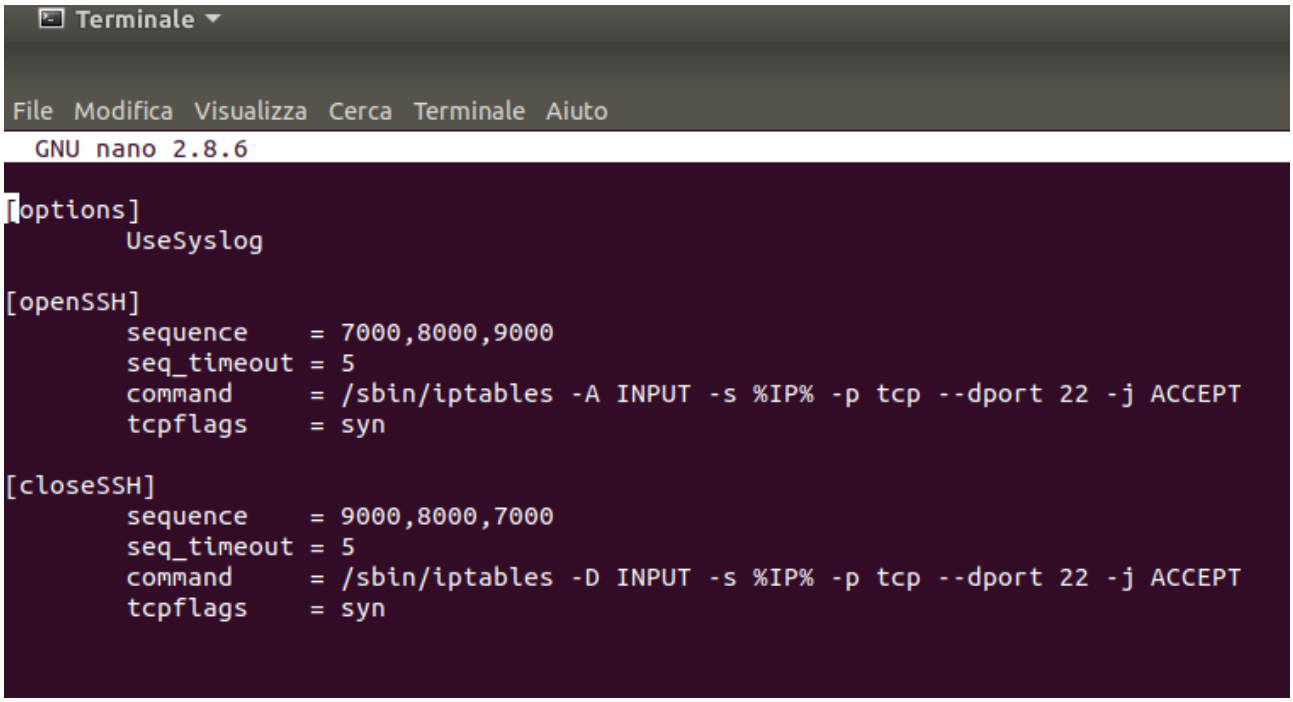


Fig. 6] File knockd.conf

In Fig. 6] si possono osservare tre diverse sezioni.
Nella sezione “options” si nota una direttiva denominata UseSyslog, questo dice a knockd che dovrebbe registrare le sue informazioni usando i normali metodi syslog. Dunque ciò inserirà i log in /var/log/messages.
Le due sezioni successive sono rispettivamente “openSSH” e “closeSSH”. In realtà i nomi di queste ultime possono essere qualsiasi e sono usati per raggruppare una serie di regole che corrisponderanno a un singolo evento ciascuna.
In questo caso si ha una sezione che aprirà la porta SSH e una che la chiuderà di nuovo.
Si possono osservare in queste ultime due sezioni i parametri su cui si baserà il Port knocking.

```
sequence = 7000,8000,9000
```

Questo parametro indica le porte a cui lo stesso IP dovrà connettersi in sequenza per far si che l’insieme di regole della corrispondente sezione venga applicato al firewall.

Altri due parametri impostano dei requisiti che la connessione deve rispettare:

```
seq_timeout = 5
tcpflags = syn
```

Il primo parametro specifica una quantità di tempo entro cui la sequenza deve essere completata.

Il secondo specifica un flag che deve essere presente nei pacchetti TCP per poter essere considerati validi. Il valore di syn qui è comunemente usato per distinguere i pacchetti che vogliamo da quelli creati in background da programmi come SSH.

Infine si ha la regola iptables:

```
command = /sbin/iptables -A INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
```

Come indicato dall'etichetta di sezione "openSSH", questa sezione aprirà una porta per le connessioni SSH quando viene completata la sequenza corretta.
La scritta %IP% verrà sostituita con l'indirizzo che ha completato la sequenza nel modo corretto.

La seconda sezione SSH fa quasi la stessa cosa, ma usa una sequenza diversa e rimuove la regola da iptables che ha aperto le connessioni a SSH. In questo modo è possibile una volta usufruito del servizio per il tempo necessario rimuovere la regola che dava accesso al servizio attraverso la corrispondente porta. Si osserva, infatti, nella riga "command" la stessa regola iptables della sezione precedente, ma con l'opzione -D che sta appunto per delate.

Dopo aver cambiato i parametri del file knockd.conf (in seguito verrà mostrato come scegliere i parametri in modo adeguato) sarà necessario abilitare il servizio Knockd andando a modificare un altro file.
Occorrerà aprire con i privilegi di root il seguente file:

```
sudo nano /etc/default/knockd
```

Di seguito la schermata del terminale apparirà simile alla seguente:

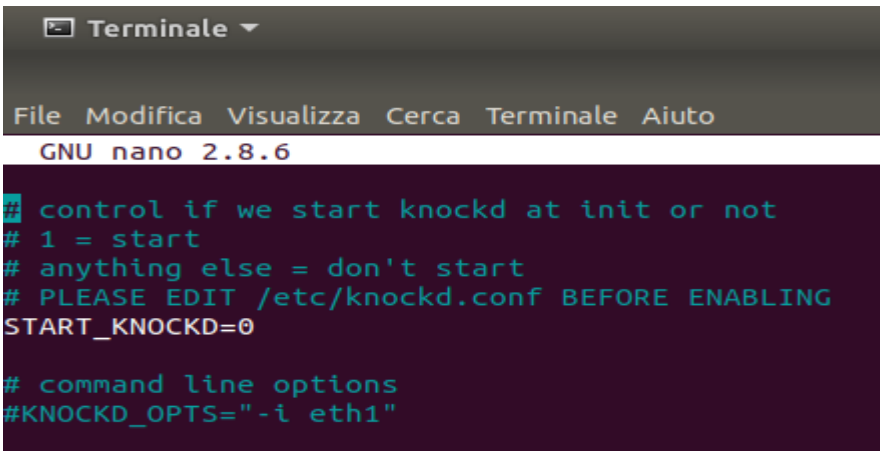


Fig. 7] File Knockd

Per avviare il servizio serve impostare START_KNOCKD=1.

Adesso è possibile avviare il servizio digitando:

```
sudo service knockd start
```

Fatto ciò il servizio Knockd sarà attivo e chiunque voglia accedere alla porta 22 relativa al servizio SSH, nell'esempio di FIG. 6], dovrà connettersi in sequenza alle porte 7000, 8000 e 9000, entro il limite di tempo "seq_timeout" impostato.

La configurazione in Fig. 6], però, è quella predefinita ed è poco sicura. Per rendere il Port knocking efficace occorrerebbe cambiare sempre la sequenza nelle due sezioni in modo casuale.
Nel prossimo paragrafo verrà mostrato come è facile bypassare il Port knocking quando la sequenza inserita è fissa, dunque verranno studiati alcuni possibili accorgimenti per rendere il Port knocking un ottimo sistema di sicurezza.

4.4 Esempio di cattiva configurazione del Port knocking e possibili soluzioni

In questo paragrafo verrà mostrato come una cattiva configurazione del servizio di Port knocking sia facile da aggirare.
Nel file Knock.conf di Fig. 6] ci sono tre parametri su cui l'utente può agire:

- Il numero di porte.
- Il numero delle porte, ovvero la sequenza.
- Il timeout consentito per completare la sequenza.

Su questi tre parametri si baserà il servizio di Port knocking.
Messi insieme tali parametri renderanno il compito a un possibile attaccante molto arduo, infatti quest'ultimo dovrebbe sapere il numero esatto di porte da "bussare", quali "bussare" nella giusta sequenza e sopratutto terminare le connessioni entro la scadenza del timeout impostato.
Nonostante ciò, una cattiva configurazione del file .conf renderà il servizio facilmente aggirabile.
Se la sequenza di porte impostata è sufficientemente vicina e con ordine dei numeri di porta crescenti o decrescenti, il servizio di Port knocking è pressoché inutile.

Si consideri per esempio la seguente sequenza di porte; 3000, 3050 e 3100. Tale sequenza presenta numeri vicini fra loro e in ordine crescente, un eventuale utente malintenzionato potrebbe tentare di accedere a tutte le porte comprese fra la numero 3000 e la numero 4000. Se il timeout impostato nel file di configurazione è sufficientemente grande l'utente che sta attaccando il server ha buone probabilità di riuscire ad aggirare il Port knocking. Infatti, il servizio rileva quando le porte della sequenza sono contattate nel giusto ordine, ma non prende provvedimenti se fra una porta e la successiva della sequenza vengono contattate altre porte differenti.

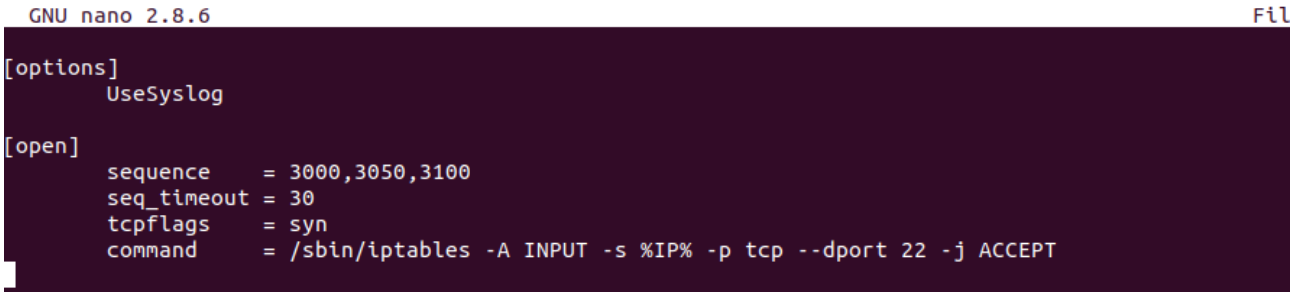


Fig. 8] Esempio di cattiva configurazione del Port knocking

In Fig. 8] è riportato un esempio di quanto detto precedentemente riguardo la cattiva configurazione del servizio knockd.
Oltre alla sequenza di porte un parametro importante è il *seq_timeout*. Nell'esempio riportato tale parametro è settato a 30, ovvero l'utente ha a disposizione ben 30 secondi per completare la sequenza.
In tale intervallo di tempo è possibile contattare ben 7500 porte con un intervallo di tempo di 4ms fra un collegamento e l'altro.
Nel caso predefinito il seq_timeout è di 5 secondi, dunque il range si restringe a 1250 porte, ipotizzando sempre di contattare le porte con un intervallo di 4ms fra una porta e la successiva.
Anche in quest'ultimo caso il range è molto elevato, tenendo conto che le porte totali disponibili sono 65335, basterebbero 52 iterazioni per esaurirle tutte.
È chiaro che l'attaccante avrà successo solo nel caso in cui una delle 52 iterazioni comprenda tutte le porte della sequenza, mentre se anche una sola porta risulterà fuori dal range della iterazione l'attacco fallirà.
Nel caso in questione, con la sequenza impostata come in Fig. 8], il numero di porte è molto basso (soltanto 3 porte) e sono sia molto vicine fra loro sia in ordine crescente, dunque in questo caso l'attaccante avrà buone probabilità che la sequenza faccia parte di una delle iterazioni.

Per porre rimedio a tale problematica è sufficiente scegliere una sequenza con porte molto distanti fra loro e possibilmente non in ordine crescente o decrescente, un esempio potrebbe essere 2500, 1000, 1500. In quest'ultimo caso l'attacco non potrà avere successo perché nessuna iterazione potrà contenere contemporaneamente tutte e tre le porte.

Tale soluzione adottata è certamente un passo in avanti, ma non consente ancora di sfruttare al massimo il servizio di Port knocking.

Un attaccante esperto potrebbe avere a sua volta degli strumenti detti di ***sniffing*** in grado di rilevare la sequenza di porte corretta senza andare per tentativi.

Uno *sniffer* è un tipo di software che registra tutto il traffico in entrata e in uscita da un computer connesso alla rete. Tali software non sono necessariamente dannosi, di norma si utilizzano per monitorare e analizzare il traffico di rete al fine di rilevare problemi e mantenere il sistema efficiente.
Un utente malintenzionato, però, con tale dispositivo sarà in grado di rilevare tutte le connessioni effettuate da un computer e dunque anche la sequenza di porte corretta per accedere a un servizio nascosto dal Port knocking. Infatti, se nel traffico di rete si dovesse presentare più volte una stessa sequenza di porte contattate e successivamente una connessione a un servizio che prima non era presente, è evidente che tale sequenza è quella necessaria per accedere a quel servizio.

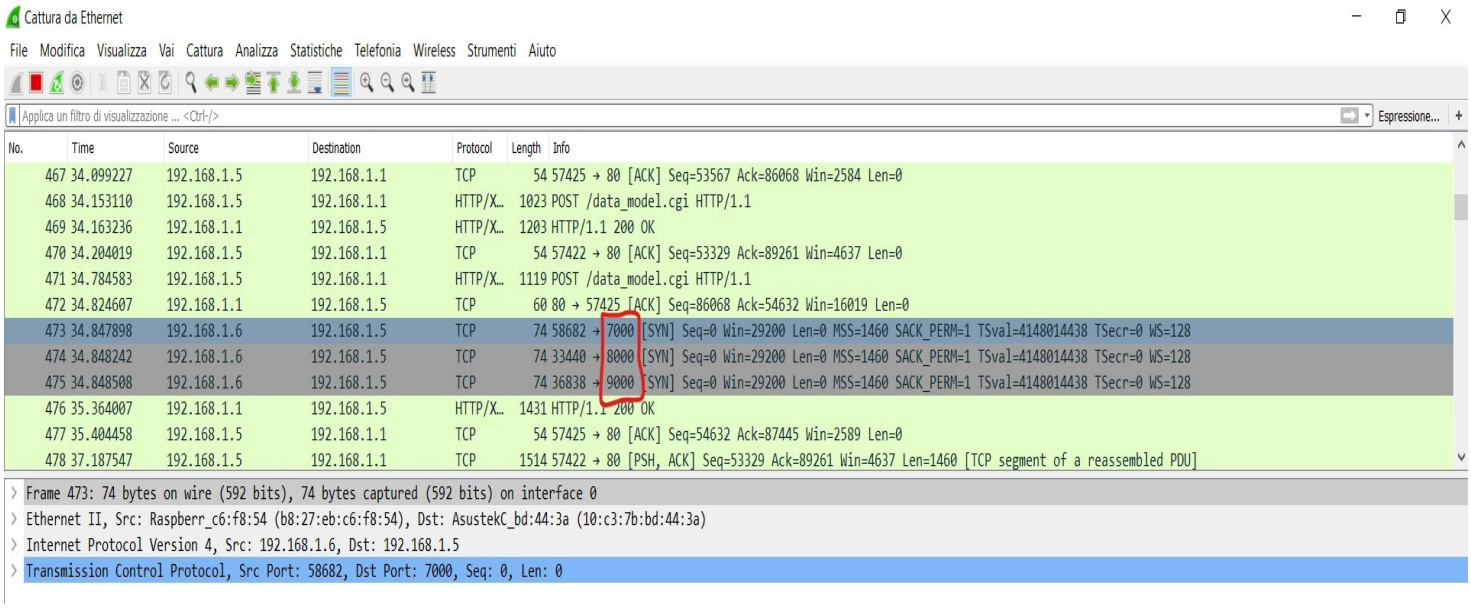


Fig.9] Visuale Wireshark del traffico della rete

La Fig. 9] mostra il traffico catturato da un software di sniffing chiamato Wireshark. Si possono osservare 3 connessioni tcp provenienti dall'indirizzo IP 192.168.1.6 e diretti verso l'indirizzo 192.168.1.5, in particolare alle sue porte 7000, 8000 e 9000, ovvero proprio alle porte impostate nel parametro *sequence* del file knockd.conf predefinito.

Per ovviare tale vulnerabilità del sistema l'idea è quella di cambiare ogni volta la sequenza di porte in modo casuale.
Così facendo anche catturando il traffico nella rete non sarà possibile in alcun modo ricondursi alla sequenza corretta in quanto quest'ultima cambierà ogni volta.

Dunque è possibile fare una lista di regole da rispettare per configurare in modo efficiente il servizio di Port knocking:

- Impostare il seq_timeout sufficientemente basso. Il valore predefinito di 5 secondi potrebbe essere superiore al necessario.
- Scegliere il maggior numero di porte possibile nella sequenza da contattare senza eccedere nel seq_timeout.
- Assicurarsi che la sequenza di porte non sia ascendente o discendente.
- Assicurarsi di configurare uno stop_command per chiudere la porta dopo alcuni secondi nel cmd_timeout.
- Eseguire qualsiasi servizio che si vuole proteggere su porte non standard, in questo modo anche se un utente malintenzionato dovesse avere successo, non troverà il servizio SSH sulla sua porta 22 standard. Ci vuole molto tempo per scansionare una macchina dopo ogni tentativo di knockout della porta, dunque usare porte non standard potrebbe salvare il sistema.
- Cambiare in modo casuale la sequenza di porte ad ogni accesso.
- Utilizzare le normali precauzioni di sicurezza (come login di sola chiave e login no-root sui daemon SSH, ad esempio).

5. TIPI DI ATTACCHI

5.1 Raccolta delle informazioni

Un buon firewall deve essere in grado di difendere il sistema da possibili attacchi esterni provenienti dalla rete. Ma quali sono questi tipi di attacchi da cui ci si deve difendere?
Un sistema di sicurezza efficiente deve tenere in considerazione qualsiasi possibile attacco, ma questo spesso può risultare molto complicato se non addirittura impossibile. Infatti, nonostante si possano suddividere in varie categorie, gli attacchi hacker sono sempre in evoluzione ed è dunque impossibile realizzare un firewall in grado di bloccarli tutti.
Dunque lo scopo di questo capitolo sarà analizzare alcuni dei più diffusi tipi di attacchi.

Prima di procedere sull'analisi degli attacchi più diffusi, occorre capire come avviene un attacco analizzandone le fasi principali.
Un attaccante spesso prima di colpire la vittima la analizza, ovvero cerca di prendere quante più informazioni possibili su di essa.
Occorre ricordare che nonostante la vasta gamma di attacchi esistenti, tutti hanno in comune il fatto di volere aggirare il sistema di sicurezza e per farlo occorre conoscere le sue vulnerabilità. Dunque la fase di ricerca delle informazioni sul sistema che si vuole compromettere è fondamentale ed è su essa che si baserà l'intero attacco.

Fondamentalmente le fasi di raccolta delle informazioni sono tre:

1. **Footprinting**
   Questa è la fase in cui l'attaccante raccoglie informazioni riguardanti il possibile obiettivo da attaccare e che permettono di determinare il footprint (letteralmente "impronta"), ovvero il profilo del sistema di sicurezza utilizzato dalla vittima; presenza di internet, tecnologia per l'accesso remoto e eventuali Intranet/Extranet.
   Gli hacker che seguono una metodologia ben strutturata riescono a raccogliere informazioni da una serie di fonti e a ricostruire l'impronta di qualsiasi organizzazione che gli permette di ricreare il profilo del suo livello di sicurezza.

Tecnologia	Informazioni
Internet	Nomi di dominio Blocchi della rete Indirizzi IP dei sistemi raggiungibili da Internet Servizi TCP e UDP in esecuzione su ciascuno dei sistemi identificati Architettura di sistema (per esempio, SPARC piuttosto che X86) Meccanismi di controllo degli accessi e relativi elenchi (ACL, *Access Control List*) Sistemi di intercettazione delle intrusioni (IDSes) Enumerazione del sistema (nomi utente e di gruppo, messaggi di sistema, tabelle di instradamento, informazioni SNMP)
Intranet	Protocolli di rete utilizzati (per esempio, IP, IPX, DecNET ecc.) Nomi di dominio interni Blocchi di rete Indirizzi IP specifici di sistemi raggiungibili sulla Intranet Servizi TCP/UDP in esecuzione su ciascuno dei sistemi individuati Architettura di sistema (per esempio SPARC, piuttosto che X86) Meccanismi di controllo degli accessi e relativi elenchi (ACL, *Access Control List*) Sistemi di intercettazione delle intrusioni (IDS) Enumerazione del sistema (nomi utente e di gruppo, messaggi di sistema, tabelle di instradamento, informazioni SNMP)
Accesso remoto	Numeri delle linee telefoniche analogiche/digitali Tipo di sistema remoto Meccanismi di autenticazione VPNs e protocolli correlati (IPSEC, PPTP)
Extranet	Fonte e destinazione della connessione Tipo di connessione Meccanismi di controllo degli accessi

Fig. 10] Lista delle tecnologie ed informazioni critiche [4]

La tabella mostrata in Fig. 10] è una lista di informazioni utili e ricercate per ricostruire il footprint. In particolare il footprinting concentra la ricerca sulle risorse informatiche possedute dal bersaglio seguendo tale tabella.

2. Scanning

Una volta conclusa l'analisi preliminare è possibile stilare un elenco delle macchine più esposte ad un attacco. Dunque questa fase consiste in una analisi più approfondita dei dispositivi selezionati al fine di individuarne le configurazioni e le eventuali vulnerabilità presenti. Infatti una volta a conoscenza del tipo di sistema operativo con cui si ha a che fare, dei servizi in esso attivi e delle debolezze del firewall, sarà possibile sfruttare le informazioni ricavate per mettere in atto un vero e proprio attacco.

Per eseguire tale analisi l’attaccante si serve per esempio del comando **Ping** che permette di sapere se una certa macchina è al momento collegata, inviandole dei pacchetti al suo indirizzo IP.
Un altro comando utilizzato è **Nmap** con il quale è possibile sapere i servizi attivi su una determinata macchina in un dato momento e dunque risalire a quali sono le porte in ingresso aperte.

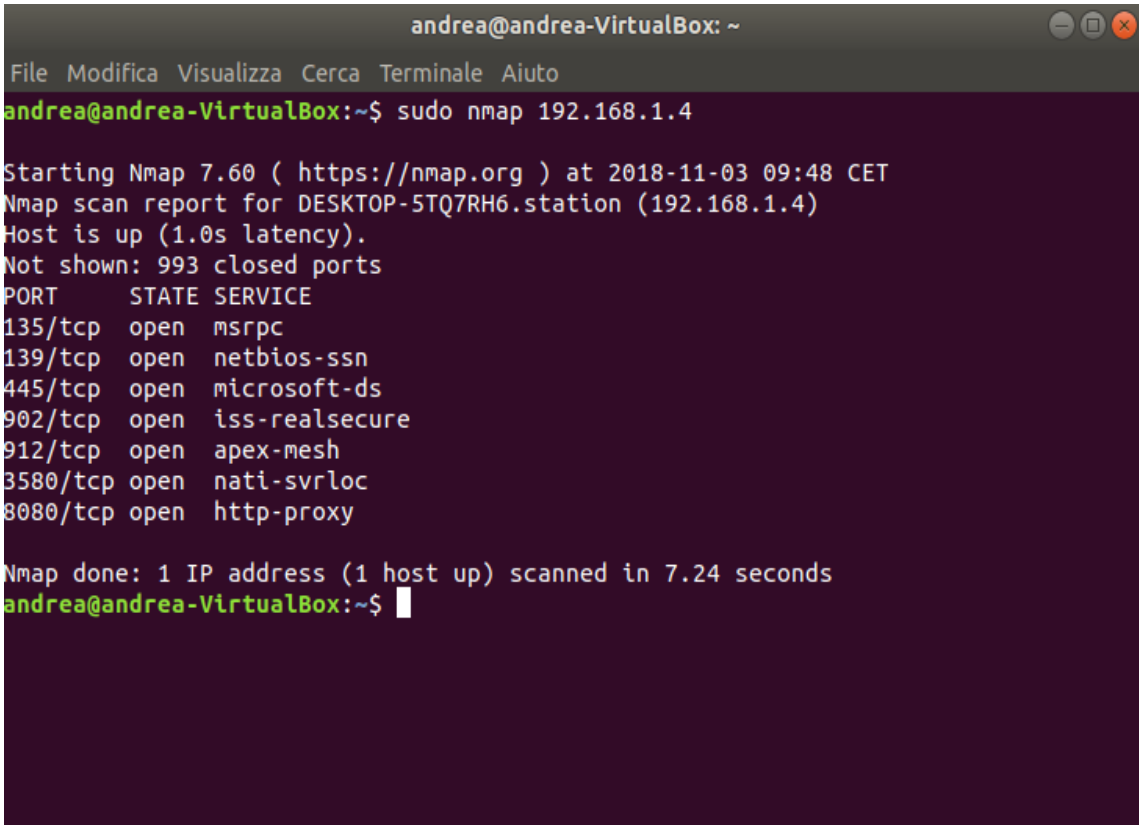


FIG. 11] Esempio di utilizzo del comando Nmap

3. Enumeration

E’ la fase intrusiva di determinazione degli account attivi, delle risorse accessibili come file condivisi in modo insicuro e di individuazione di vecchie versioni del software che contengono vulnerabilità note. Una volta ottenuti gli account attivi, infatti, l’attaccante può tentare di indovinare le password per ottenere l’accesso al sistema.

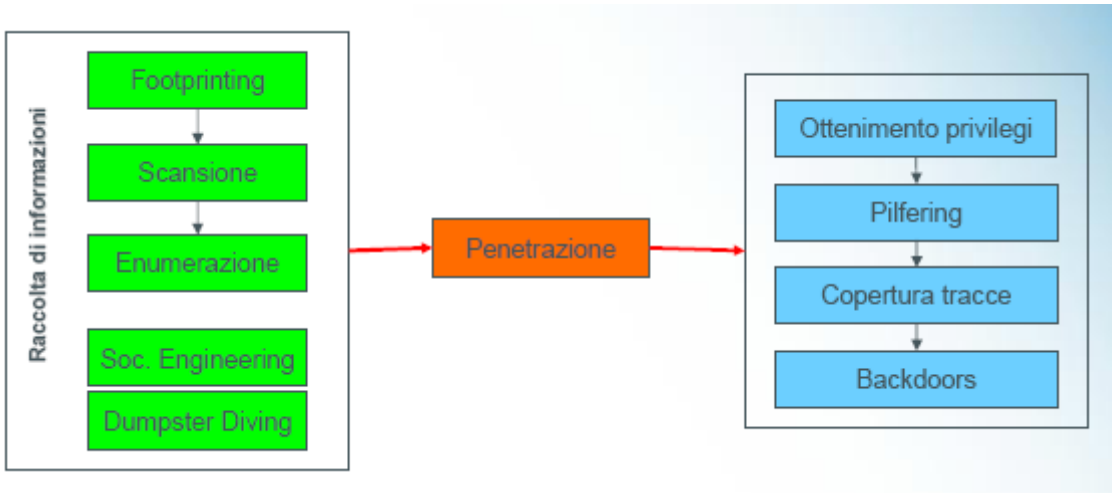


Fig. 12] Schema relativo alle fasi di un attacco informatico [5]

Raccolte le informazioni sulla vittima la fase successiva è quella dell'attacco vero e proprio.
L'attaccante entra nel sistema sfruttando le vulnerabilità individuate nelle fasi precedenti, una volta penetrato avrà accesso a tutte le informazioni e arrivati a questo punto sarà molto più difficile fermarlo.

5.2 Tipologie di attacchi

Sniffing

Lo sniffing è una tecnica che consiste nel catturare il traffico all'interno di una rete. Di norma è utilizzato per individuare eventuali problemi nella rete di comunicazione, dunque per scopi legittimi. Però a volte viene utilizzato per scopi illegali, come per intercettare password o altre informazioni sensibili.

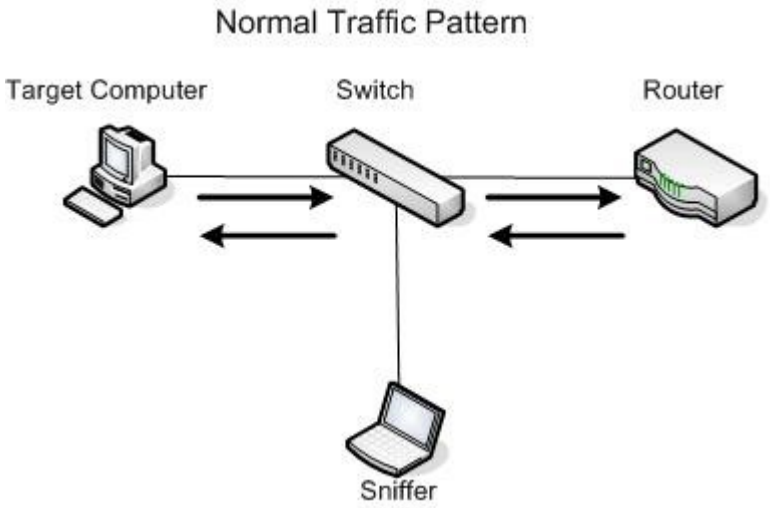


Fig.13] Rappresentazione grafica di uno sniffer [6]

Una volta catturato il traffico, l'attaccante può leggere i dati contenuti nei pacchetti di rete, fra essi possono essere presenti password trasmesse dalla vittima per autenticarsi nei confronti di un servizio a cui sta cercando di accedere.
Gli strumenti utilizzati per mettere in atto questa pratica sono detti *sniffer*, fra essi il già citato Wireshark.

A difesa dell'utente molti servizi utilizzano algoritmi per cifrare il traffico in modo da rendere inutilizzabili le informazioni catturate dallo sniffer. Esistono inoltre molti software che rilevano gli sniffer presenti nella rete.

Attacchi DOS

Un attacco DoS (Denial of Service) ha come scopo il rendere inutilizzabile un determinato servizio in ascolto sul web tramite il continuo invio di richieste fittizie. Qualsiasi servizio esposto su internet che fornisce servizi di rete basati sul protocollo TCP/IP è soggetto al potenziale pericolo di attacchi DoS.

Le comunicazioni su internet avvengono tramite il protocollo TCP/IP che prevede una sincronizzazione della connessione fra sorgente e destinazione realizzata tramite il ***Three Way Handshaking***; il client inizia la connessione verso il server inviando un pacchetto TCP contrassegnato con il flag SYN e con un certo numero di sequenza x, il server risponde alla richiesta tramite una notifica di avvenuta ricezione generano un pacchetto con il flag SYN e ACK=x+1 e un diverso numero di sequenza y. Concordate le sequenze il prossimo pacchetto che invierà il client conterrà ACK=y+1, seq=x+1 e i dati veri e propri.

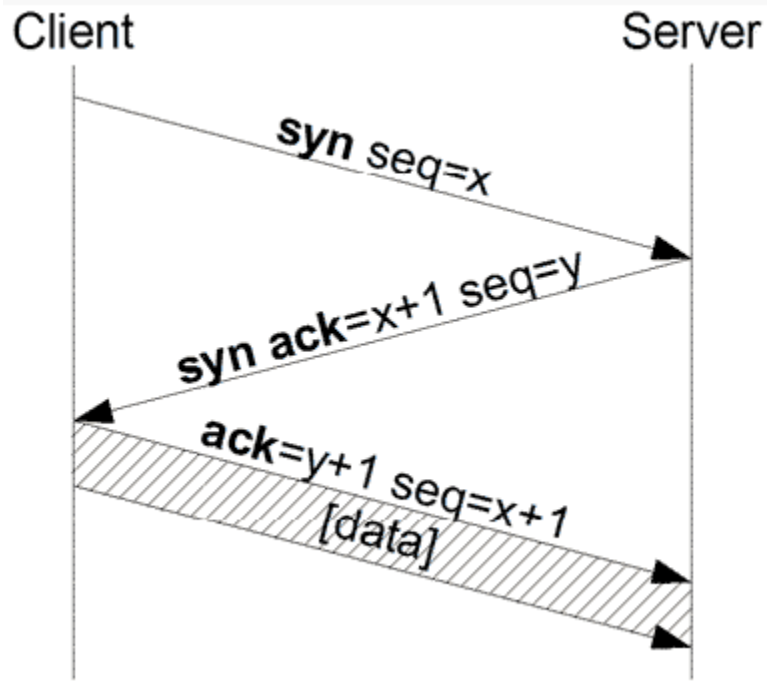


FIG. 14] Protocollo Handshake [7]

L’attacco al server può dunque essere realizzato inviando contemporaneamente un elevato numero di richieste di sincronizzazione al secondo. Il server risponderà inviando SYN/ACK finché le sue possibilità hardware e software lo permetteranno, ovvero finché l’utilizzo della CPU non arriverà al 100%. Se l’utilizzo della CPU dovesse saturare, si avrà uno stallo della architettura con conseguenza l’immediato arresto del server che

non permetterà nessun tipo di operazione, sia essa di amministrazione locale della macchina o in remoto.
Anche tutte le connessioni dall'esterno non saranno più possibili e dunque tutti i servizi non risulteranno più disponibili.
Siccome questa particolare tecnica di attacco DoS prevede l'invio di una grande quantità di pacchetti verso una unica destinazione, viene anche definita **Flooding** (inondamento).

Attacchi DDOS

Gli attcchi DDoS (Distribuited Denial of Service) sono una evoluzione degli attacchi DoS precedentemente descritti.
La differenza sta nel numero di macchine utilizzate per eseguire l'attacco. L'insieme di computer disseminati per la rete compongono una botnet a disposizione dell'attaccante. Chi esegue l'attacco non si espone direttamente, per evitare di essere individuato, ma "infetta" un numero elevato di computer utilizzando virus di varia specie che permettono di lasciare aperte delle backdoor a loro riservate, in questo modo l'attaccante riesce a prendere il controllo remoto della macchina senza il consenso del proprietario.

Man-in-the-Middle

Nella forma più semplice questi attacchi prevedono che l'attaccante si inserisca fra due entità che stanno cercando di comunicare fra loro, "avvelena" la comunicazione e intercetta i messaggi.
La presenza dell'attaccante non è percepita né dalla vittima (il client), né dalla fonte (il server o il router) che la prima sta cercando di contattare.
L'attaccante utilizza una rete Wi-Fi per intercettare le comunicazioni dell'utente. Questo avviene di solito sfruttando una falla nel setup del router con il fine di intercettare le sessioni degli utenti sul router stesso.
Chi esegue l'attacco potrebbe utilizzare un hotspot Wi-Fi gratuito, ovvero creare un nodo Wi-Fi falso che simula un legittimo punto di accesso dandogli uno di quei nomi comunemente usati nelle aree Wi-Fi pubbliche.
Quando un utente si connette al "falso" router e cerca di accedere a siti sensibili, l'attaccante identifica una debolezza nella configurazione o nel sistema di criptografia del router e la sfrutta per intercettare la comunicazione fra l'utente e il router compromesso.

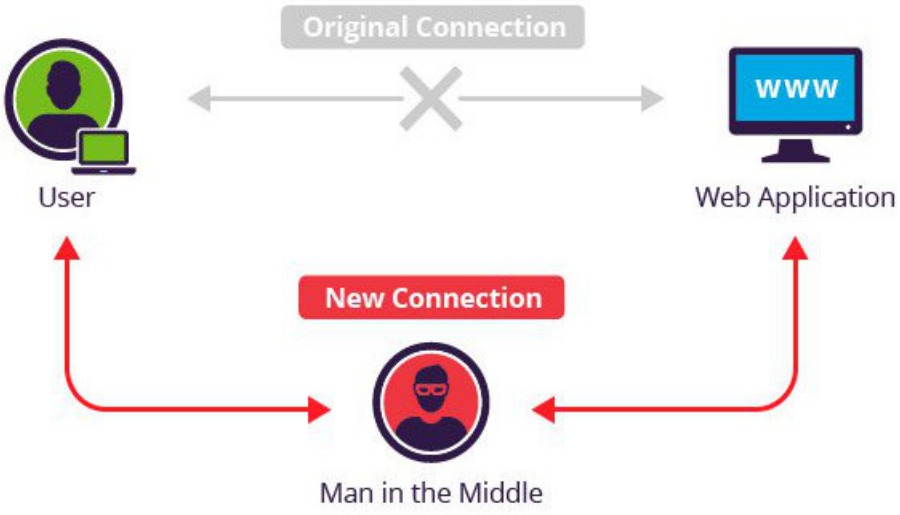


FIG. 15] Attacco Man-in-the-Middle [8]

Web spoofing

Tale attacco consiste nella falsificazione di un sito web. Un utente (vittima) crede di navigare all'interno del sito web richiesto mentre in realtà è connesso a un server malevolo.

L'attaccante inganna la vittima servendosi di hyperlink fasulli che indirizzano l'utente che li seleziona non al sito web desiderato, ma a quello creato dall'attaccante.

Il falso sito web assomiglia completamente a quello vero, in modo che la vittima non si accorga dell'inganno. In questo modo l'attaccante potrà impadronirsi del traffico fra il browser web dell'utente e il web permettendogli così di ottenere anche dati sensibili dell'utente, come per esempio le sue credenziali.

Se per esempio la vittima decidesse di effettuare un ordine on-line, l'attaccante potrebbe modificare l'indirizzo e inviare a sé stesso l'ordine.

6. ESEMPI DI ATTACCO E DIFESA

6.1 Attacco DoS

In questo capitolo verranno eseguiti veri attacchi verso un dispositivo vittima, in particolare un raspberry pi. In particolare verrà analizzata la dinamica dell'attacco e successivamente si analizzerà una possibile configurazione del firewall per difendersi da esso.
Come primo attacco si analizza il Denial of Service e gli strumenti utilizzati sono i seguenti:

- Un PC con installato il sistema operativo Linux Ubuntu su macchina virtuale.
- Un Raspberry pi, utilizzato come computer vittima.
- Un software per realizzare l'attacco DoS chiamato PyLoris.
- Il software Wireshark, per monitorare il traffico durante l'attacco.
- Una rete Wi-Fi.

Per prima cosa si imposta il raspberry, su cui è installato il sistema operativo Raspbian, come server web.
Per installare il server web occorre collegarsi al raspberry tramite SSH, per farlo si utilizza il seguente comando:

```
Sudo ssh pi@ <ip>
```

In questo caso l'indirizzo IP del raspberry è 192.168.1.5, mentre quello del PC utilizzato per eseguire l'attacco 192.168.1.4.
Una volta connessi per installare il server web Apache si utilizza il seguente comando:

```
sudo apt-get install apache2
```

Giunti a questo punto per attivare il servizio Apache occorre digitare:

```
sudo service apache2 start
```

Fatto ciò il server è reso funzionante e, collegandosi in locale tramite un qualunque browser web all'indirizzo 192.168.1.5/index.html, la pagina mostrata è quella di default del server Apache.

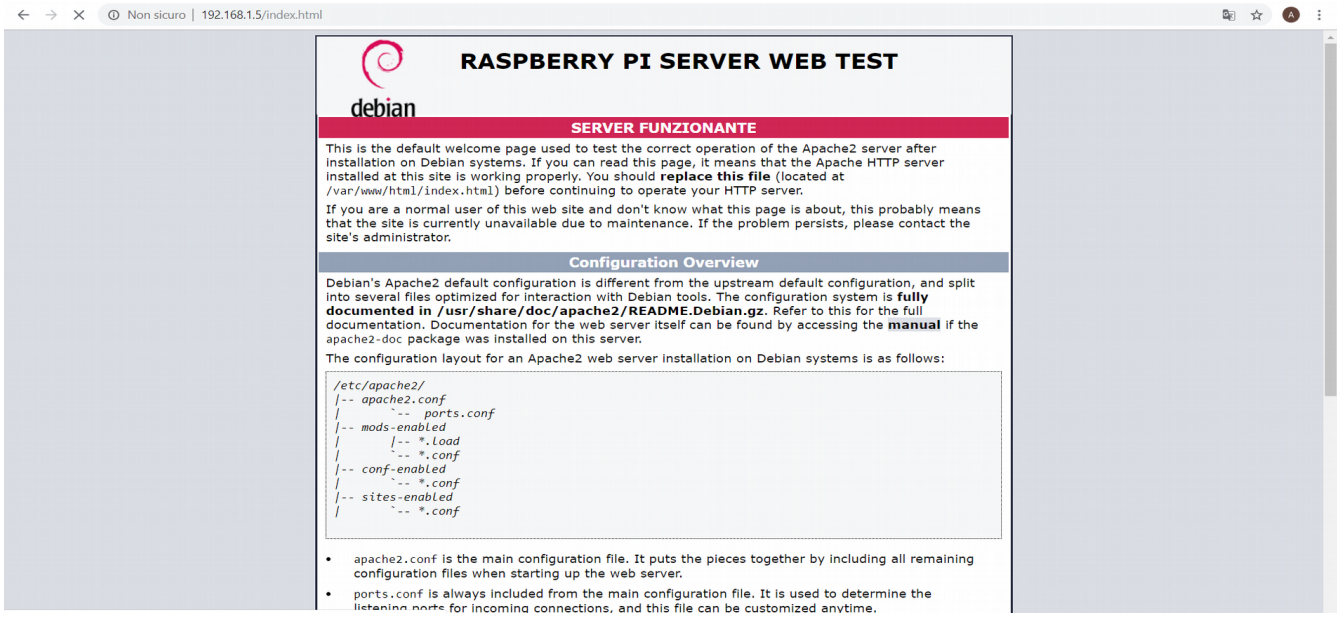


FIG. 16] Pagina web Raspberry pi

FASE DI ATTACCO

Fatto ciò è possibile procedere all'attacco. Per prima cosa tramite comando Nmap, digitato dal computer attaccante, si analizzano le porte aperte sul dispositivo vittima.

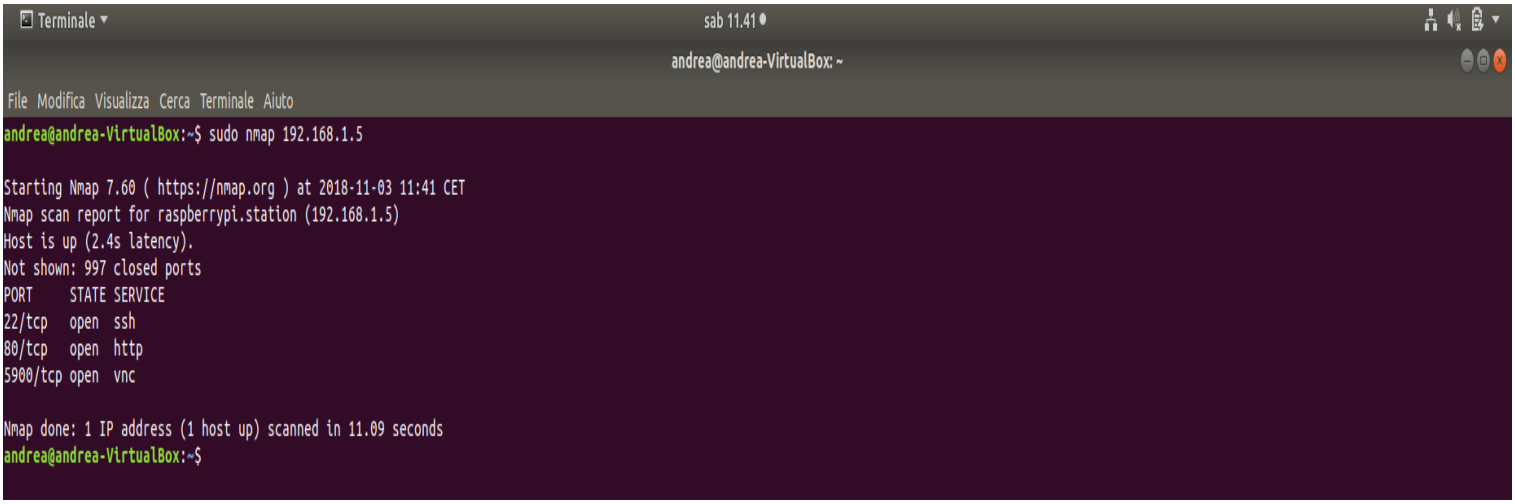


FIG.17] Porte attive sul Raspberry

Dal controllo delle porte attive si nota in FIG. 17] la numero 80 e il relativo servizio http, ovvero la pagina web precedentemente configurata.

Individuata la porta è possibile mandare in esecuzione l'attacco tramite il software PyLoris.

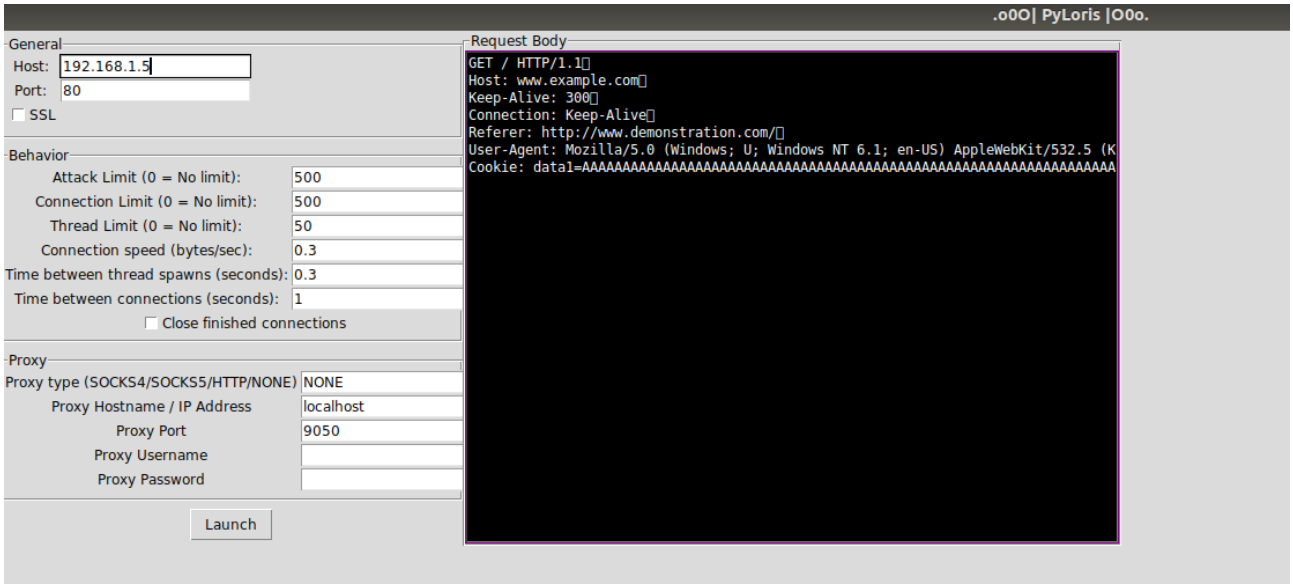


FIG.18] Software PyLoris per attacco DoS

Tramite questo software è possibile, una volta selezionata la vittima e la relativa porta, decidere il numero di attacchi, il numero di connessioni, la velocità di ogni connessione e il tempo fra una connessione e l'altra.
In FIG.18] si possono osservare i parametri utilizzati.
Selezionando "Launch" l'attacco viene eseguito.
Per rendere l'attacco più efficace si mandano in esecuzione più sessioni alla volta.

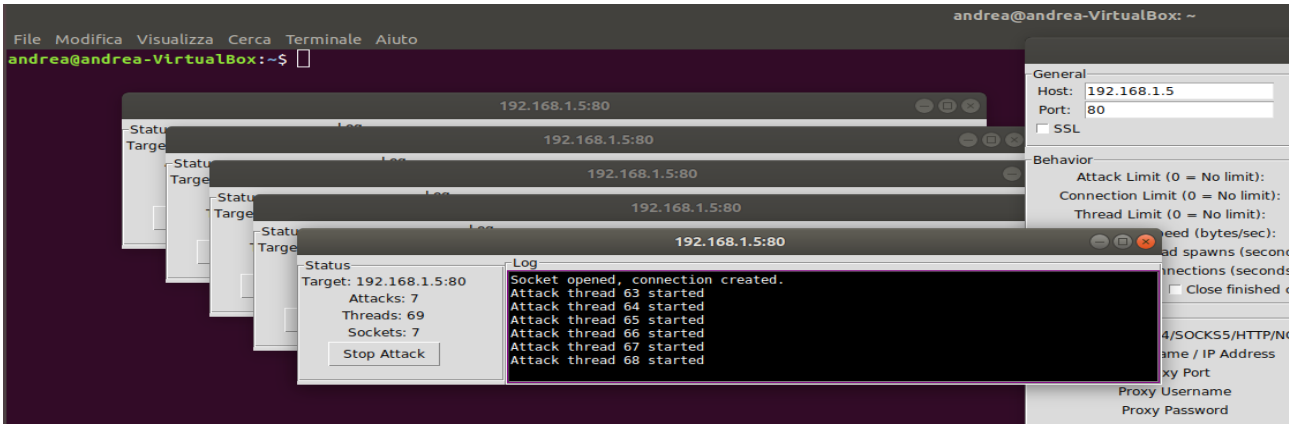


FIG.19] Sessioni attive di PyLoris

Infine è viene mandato in esecuzione il software Wireshark per monitorare il traffico nella rete durante l'attacco.

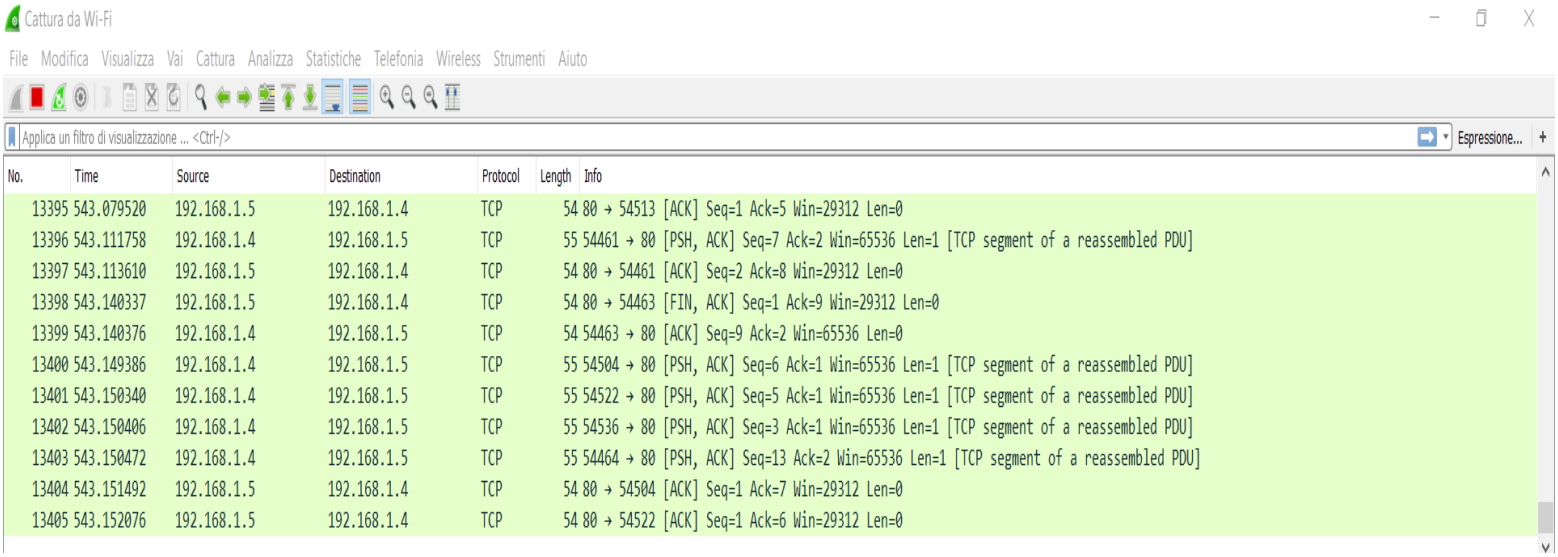


FIG.20] Visuale Wirshark del traffico nella rete durante l'attacco

Il risultato finale dell'attacco è un evidente rallentamento della pagina web.
Impostando parametri più "aggressivi" e eseguendo il medesimo attacco da più dispositivi (attacco DDoS), probabilmente si sarebbe riusciti a mandare offline completamente il server.

FASE DI DIFESA
Adesso si analizzano alcuni possibili modi per difendersi da attacchi di questo genere.
In particolare si considerano tre linee di difesa contro gli attacchi DoS e DDos:

1. **Prevenzione e cognizione**
   Questa fase avviene prima dell'attacco e prevede tutti quei meccanismi che consentono alla vittima di resistere al tentativo di negazione del servizio e mantenere quest'ultimo disponibile per gli utenti legittimi.

2. **Rilevazione degli attacchi e filtraggio**
   Questa fase prevede di individuare gli attacchi analizzando i servizi presi di mira e il numero di connessioni attive, successivamente si cerca di mitigare gli attacchi tramite filtraggi da parte del firewall.

3. **Risalire alla sorgente dell'attacco**
   Una volta difeso il sistema, il passo successivo consiste nel individuare la sorgente dell'attacco per poi bloccarla in modo definitivo. Quest'ultima fase non sempre è possibile da realizzare in quanto spesso l'attaccante riesce a rendersi irrintracciabile.

Dunque il primo passo è configurare il firewall in modo da mitigare l'attacco.

```
iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --set

iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 60 --hitcount
5 -j DROP
```

Con questa regola associata alla catena **INPUT** un utente non potrà effettuare più di 5 connessioni alla porta 80. Superato tale limite l'indirizzo IP associato a tale utente verrà bloccato per 60 secondi.

Inoltre siccome molti attacchi si basano sul port-scan, è utile limitare tali tentativi tramite:

```
iptables -A INPUT -p tcp --tcp-flags SYN,ACK,FIN,RST SYN -m limit --limit 1/s -j
DROP
```

Con questa regola verranno eliminati tutti i pacchetti con i flag SYN, ACK, FIN e RST ricevuti dal firewall nel caso in cui la loro frequenza sia superiore a 1 al secondo. Tali flag sono quelli usati dal comando Nmap.

Infine è utile salvare il traffico in file log. Tali file verranno salvati in /var/log/syslog e saranno identificati dal prefisso che si desidera assegnargli.

FIG.21] Chain PORT_SCAN e relative regole

In particolare il comando che salva i file log è il seguente:

```
iptables -A PORT-SCAN -j LOG --log-prefix "PORT_SCAN: "
```

La fase successiva consiste nel individuare l'attacco.
In particolare se l'attacco è, come nel caso in questione, di tipo DoS, risulta semplice identificare l'IP dell'attaccante.
Utilizzando il comando **netstat** si possono visualizzare tutte le connessioni attive sul dispositivo.

```
pi@raspberrypi:~ $ sudo netstat -an -t
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0    172 192.168.1.5:22          192.168.1.4:60776       ESTABLISHED
tcp        0      0 127.0.0.1:42869         127.0.0.1:33422         ESTABLISHED
tcp        0      0 127.0.0.1:33422         127.0.0.1:42869         ESTABLISHED
tcp6       0      0 :::5900                 :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
pi@raspberrypi:~ $
```

FIG.22] Lista connessioni attive

Durante un attacco di tipo Denial of Service si rileverà un numero anomalo di connessioni a una determinata porta.

```
pi@raspberrypi:~ $ sudo netstat -an -t
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 192.168.1.5:22          192.168.1.4:60776       ESTABLISHED
tcp        0      0 127.0.0.1:42869         127.0.0.1:33422         ESTABLISHED
tcp        0      0 127.0.0.1:33422         127.0.0.1:42869         ESTABLISHED
tcp6       0      0 :::5900                 :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 192.168.1.5:80          192.168.1.4:60969       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60939       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60997       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:61004       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60991       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60961       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60989       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60912       FIN_WAIT2
tcp6       0      0 192.168.1.5:80          192.168.1.4:60915       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60932       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60906       FIN_WAIT2
tcp6       0      0 192.168.1.5:80          192.168.1.4:60987       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60985       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60963       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60972       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60933       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60919       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60977       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:61009       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60938       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60970       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60998       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60935       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60913       FIN_WAIT2
tcp6       0      0 192.168.1.5:80          192.168.1.4:61003       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:61001       SYN_RECV
tcp6       0      0 192.168.1.5:80          192.168.1.4:60929       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60973       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60926       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60943       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60921       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60978       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60955       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60974       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60947       ESTABLISHED
tcp6       0      0 192.168.1.5:80          192.168.1.4:60999       SYN_RECV
```

FIG.23] Lista connessioni attive durante l'attacco

In questo caso si osserva in FIG.23] un numero elevato di connessioni alla porta 80 provenienti dall'indirizzo IP 192.168.1.4.
È dunque facile concludere che la sorgente dell'attacco sia proprio tale dispositivo.
Dunque inserendo una specifica regola nella chain INPUT si bloccherà in maniera definitiva l'attacco:

```
iptables -A INPUT -s <ip sorgente dell'attacco> -j DROP
```

CONCLUSIONI

In un caso reale risalire all'IP dell'attaccante potrebbe risultare molto più difficile, sopratutto se l'attacco è di di tipo DDoS e dunque i dispositivi utilizzati per l'attacco sono un numero elevato. Questo è dovuto al fatto che il vero attaccante non si espone mai direttamente, ma usa tutta una serie di tecniche di bouncing, basate su proxy anonimi, che gli consentono di mantenere un certo anonimato.
Nonostante ciò, configurando il firewall come nell'esempio è stato possibile mitigare l'attacco senza dover mandare offline il server, dunque consentendo agli utenti legittimi di usufruire del servizio.

Di seguito verranno mostrati alcuni test sul tempo di caricamento della pagina web eseguiti con il comando cURL.

```
curl -s -w 'Test del tempo di risposta per :%{url_effective}\n\nTotal Time:\t\t%{time_total}\n'
-o /dev/null 172.20.10.3/index.html
```

In questo caso l'indirizzo ip del raspberry è 172.20.10.3.

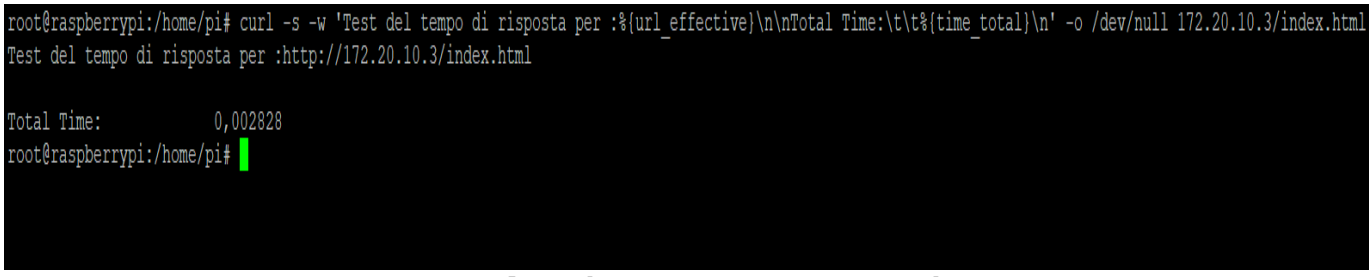


Fig. 24] Test di accesso in situazioni normali

In situazioni normali il tempo di caricamento della pagina web è molto basso come mostrato in Fig. 24].

TEST 1]
Si considera il tempo necessario per accedere alla pagina web sotto attacco nel caso in cui non sia stato impostato alcun firewall.

```
root@raspberrypi:/home/pi# curl -s -w 'Test del tempo di risposta per :%{url_effective}\n\nTotal Time:\t\t%{time_total}\n' -o /dev/null 172.20.10.3/index.html
Test del tempo di risposta per :http://172.20.10.3/index.html

Total Time:             21,660854
root@raspberrypi:/home/pi# curl -s -w 'Test del tempo di risposta per :%{url_effective}\n\nTotal Time:\t\t%{time_total}\n' -o /dev/null 172.20.10.3/index.html
Test del tempo di risposta per :http://172.20.10.3/index.html

Total Time:             34,971922
root@raspberrypi:/home/pi# curl -s -w 'Test del tempo di risposta per :%{url_effective}\n\nTotal Time:\t\t%{time_total}\n' -o /dev/null 172.20.10.3/index.html
Test del tempo di risposta per :http://172.20.10.3/index.html

Total Time:             20,550338
```

Fig. 25] Test 1 senza firewall

Dalla Fig. 25] si può osservare come i tempi di caricamento della pagina web siano aumentati notevolmente durante l'attacco, con un picco di 34,9 secondi.

TEST 2]
Si considera adesso la regola IPTABLES che limita gli accessi.

```
iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --set

iptables -I INPUT -p tcp --dport 80 -m state --state NEW -m recent --update --seconds 60 --hitcount
5 -j DROP
```

```
Total Time:             0,002221
root@raspberrypi:/home/pi# curl -s -w 'Test del tempo di risposta per :%{url_effective}\n\nTotal Time:\t\t%{time_total}\n' -o /dev/null 172.20.10.3/index.html
Test del tempo di risposta per :http://172.20.10.3/index.html

Total Time:             0,002214
root@raspberrypi:/home/pi# curl -s -w 'Test del tempo di risposta per :%{url_effective}\n\nTotal Time:\t\t%{time_total}\n' -o /dev/null 172.20.10.3/index.html
Test del tempo di risposta per :http://172.20.10.3/index.html

Total Time:             0,002156
root@raspberrypi:/home/pi#
```

Fig.26] Test 2 con firewall

In quest'ultimo caso si può osservare come i tempi di accesso per gli utenti legittimi siano pressoché inalterati dall'attacco.

6.2 IP spoofing

In questo esempio si vuole simulare un attacco di tipo IP spoofing.
Si considerano 3 reti diverse collegate fra loro attraverso un router; rete1, rete2, rete3. Nel router viene impostata una regola iptables che consente il traffico fra rete1 e rete3 e nega a rete2 di connettersi a rete1. Si considera come regola la seguente:

```
Iptables -A FORWARD -s 10.0.7.1/28 -d 10.0.9.1/28 -j ACCEPT

Iptables -A FORWARD -s 10.0.9.1/28 -d 10.0.7.1/28 -j ACCEPT
```

La defoult policy della chain FORWARD è di tipo DROP, dunque tutto il resto del traffico non verrà accettato.

L'obiettivo dell'attaccante, situato nella rete2, è di connettersi alla rete1 utilizzando un IP "fasullo".
Infatti, si può osservare che la regola impostata nel router è estremamente poco specifica e si limita a consentire il traffico sulla base degli indirizzi IP della sorgente e della destinazione. Dunque per un attaccante risulterà facile appropriarsi di un IP a cui l'accesso è consentito e aggirare il firewall.

Per realizzare tale esempio è stato prima necessario creare 3 reti virtuali all'interno di una singola macchina fisica.

Per prima cosa si realizzano le reti e il router virtuali tramite il seguente comando:

```
ip netns add <netns name>
```

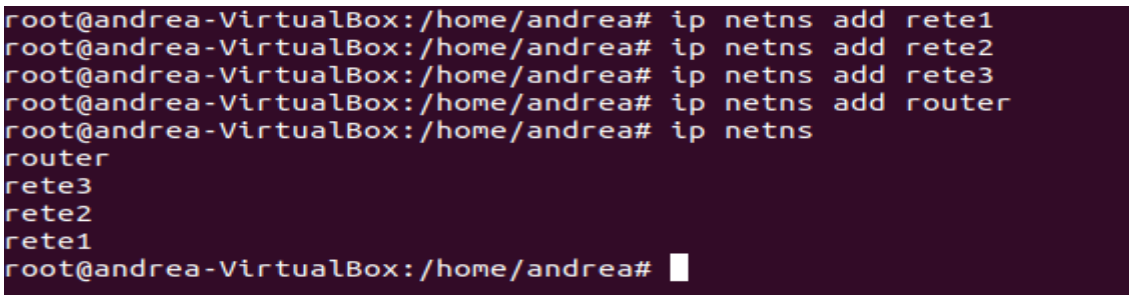


Fig. 27] Creazioni delle reti e del router virtuali

Vengono poi realizzate le interfacce di rete virtuali con il seguente comando:

```
ip link add <one side veth name> type veth peer name <other side veth name>
```

```
root@andrea-VirtualBox:/home/andrea# ip link add veth1 type veth peer name veth1-R
root@andrea-VirtualBox:/home/andrea# ip link add veth2 type veth peer name veth2-R
root@andrea-VirtualBox:/home/andrea# ip link add veth3 type veth peer name veth3-R
root@andrea-VirtualBox:/home/andrea#
```

Fig. 28] Creazione delle interfacce di rete virtuali

Si assegnano alle varie reti le corrispondenti interfacce:

```
ip link set <interface name> netns <netns name>
```

```
root@andrea-VirtualBox:/home/andrea# ip link add veth1 type veth peer name veth1-R
root@andrea-VirtualBox:/home/andrea# ip link add veth2 type veth peer name veth2-R
root@andrea-VirtualBox:/home/andrea# ip link add veth3 type veth peer name veth3-R
root@andrea-VirtualBox:/home/andrea# ip link set veth1-R netns router
root@andrea-VirtualBox:/home/andrea# ip link set veth2-R netns router
root@andrea-VirtualBox:/home/andrea# ip link set veth3-R netns router
root@andrea-VirtualBox:/home/andrea# ip link set veth1 netns rete1
root@andrea-VirtualBox:/home/andrea# ip link set veth2 netns rete2
root@andrea-VirtualBox:/home/andrea# ip link set veth3 netns rete3
root@andrea-VirtualBox:/home/andrea#
```

Fig. 29] Assegnazione delle interfacce ai namespace corrispondenti

Si abilitano le interfacce di rete virtuali:

```
Ip netns exec <netns name> ip link set dev <interface name> up
```

```
root@andrea-VirtualBox:/home/andrea# ip netns exec rete1 ip link set dev veth1 up
root@andrea-VirtualBox:/home/andrea# ip netns exec rete2 ip link set dev veth2 up
root@andrea-VirtualBox:/home/andrea# ip netns exec rete3 ip link set dev veth3 up
root@andrea-VirtualBox:/home/andrea# ip netns exec router ip link set dev veth1-R up
root@andrea-VirtualBox:/home/andrea# ip netns exec router ip link set dev veth2-R up
root@andrea-VirtualBox:/home/andrea# ip netns exec router ip link set dev veth3-R up
root@andrea-VirtualBox:/home/andrea#
```

Fig. 30] Abilitazione delle interfacce virtuali

Infine si assegnano gli indirizzi IP alle varie interfacce di rete virtuali:

```
Ip netns exec <netns name> ip address add <ip/mask> dev <interface name>
```

```
root@andrea-VirtualBox:/home/andrea# ip netns exec rete1 ifconfig veth1 10.0.7.1/28
root@andrea-VirtualBox:/home/andrea# ip netns exec rete2 ifconfig veth2 10.0.8.1/23
root@andrea-VirtualBox:/home/andrea# ip netns exec rete3 ifconfig veth3 10.0.9.1/28
root@andrea-VirtualBox:/home/andrea# ip netns exec router ifconfig veth1-R 10.0.7.2/28
root@andrea-VirtualBox:/home/andrea# ip netns exec router ifconfig veth2-R 10.0.8.2/23
root@andrea-VirtualBox:/home/andrea# ip netns exec router ifconfig veth3-R 10.0.9.2/28
root@andrea-VirtualBox:/home/andrea#
```

Fig. 31] Assegnazione degli indirizzi IP

Giunti a questo punto è stato necessario configurare le tabelle di route per le 3 reti e abilitare il forward nel router.
Per configurare le tabelle di route si utilizza il seguente comando:

```
Route add -net <ip_destinazione/mask> gw <ip_gw>
```

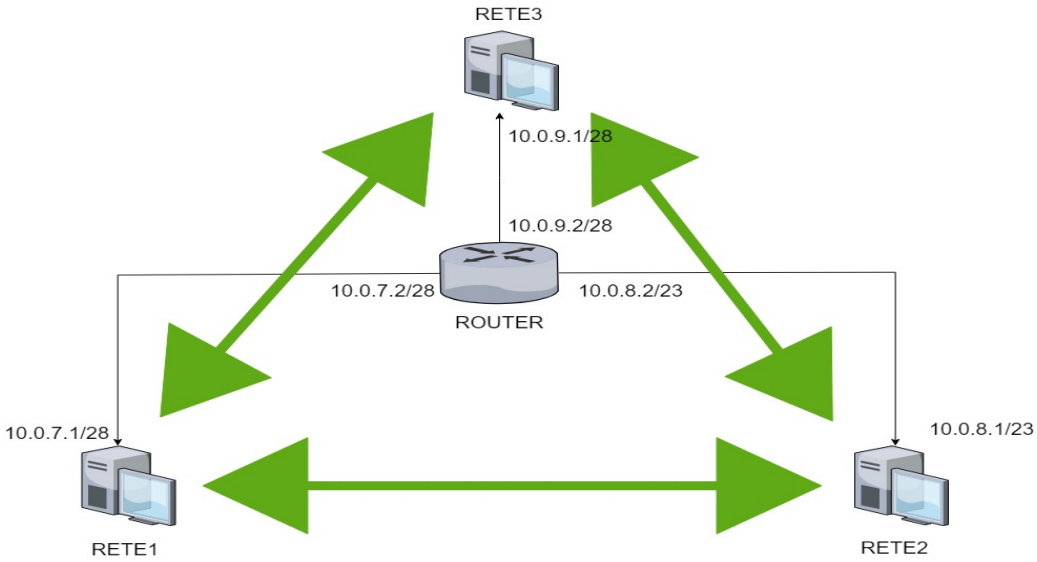
```
root@andrea-VirtualBox:/home/andrea# ip netns exec rete1 bash
root@andrea-VirtualBox:/home/andrea# route add -net 10.0.8.0/23 gw 10.0.7.2
root@andrea-VirtualBox:/home/andrea# route add -net 10.0.9.0/28 gw 10.0.7.2
root@andrea-VirtualBox:/home/andrea# ip netns exec rete2 bash
root@andrea-VirtualBox:/home/andrea# route add -net 10.0.7.0/28 gw 10.0.8.2
root@andrea-VirtualBox:/home/andrea# route add -net 10.0.9.0/28 gw 10.0.8.2
root@andrea-VirtualBox:/home/andrea# ip netns exec rete3 bash
root@andrea-VirtualBox:/home/andrea# route add -net 10.0.7.0/28 gw 10.0.9.2
root@andrea-VirtualBox:/home/andrea# route add -net 10.0.8.0/23 gw 10.0.9.2
root@andrea-VirtualBox:/home/andrea# ip netns exec router bash
root@andrea-VirtualBox:/home/andrea# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@andrea-VirtualBox:/home/andrea#
```

Fig. 32] Tabelle di routing e abilitazione del forward

Per attivare il passaggio del traffico attraverso il router occorre utilizzare il seguente comando:

```
Sysctl -w net.ipv4.ip_forward=1
```

A questo punto tutte le reti sono fra loro connesse e eseguendo il comando PING da una qualsiasi di esse verso un’altra il traffico verrà lasciato passare.




```
root@andrea-VirtualBox:/home/andrea# ip netns exec rete2 bash
root@andrea-VirtualBox:/home/andrea# ping 10.0.7.1
PING 10.0.7.1 (10.0.7.1) 56(84) bytes of data.
64 bytes from 10.0.7.1: icmp_seq=1 ttl=63 time=0.240 ms
64 bytes from 10.0.7.1: icmp_seq=2 ttl=63 time=0.040 ms
64 bytes from 10.0.7.1: icmp_seq=3 ttl=63 time=0.040 ms
64 bytes from 10.0.7.1: icmp_seq=4 ttl=63 time=0.039 ms
64 bytes from 10.0.7.1: icmp_seq=5 ttl=63 time=0.041 ms
64 bytes from 10.0.7.1: icmp_seq=6 ttl=63 time=0.039 ms
64 bytes from 10.0.7.1: icmp_seq=7 ttl=63 time=0.041 ms
64 bytes from 10.0.7.1: icmp_seq=8 ttl=63 time=0.056 ms
^C
--- 10.0.7.1 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7173ms
rtt min/avg/max/mdev = 0.039/0.067/0.240/0.065 ms
root@andrea-VirtualBox:/home/andrea#
```

Fig. 33] Ping dalla rete2 verso la rete1

Il passo successivo è dunque l’inserimento del firewall nel router.
Una volta inserito il firewall non sarà più concesso il traffico proveniente dalla rete2 verso la rete1.

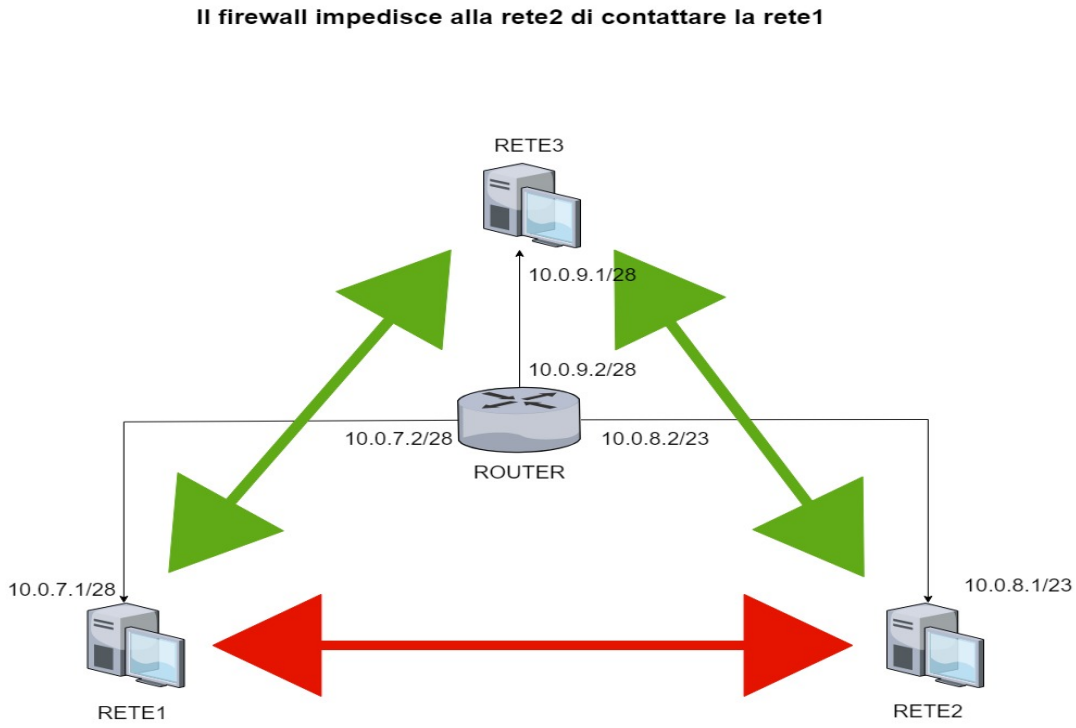
```
root@andrea-VirtualBox:/home/andrea# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  10.0.8.0/23          10.0.9.0/28
ACCEPT     all  --  10.0.7.0/28          10.0.9.0/28
ACCEPT     all  --  10.0.9.0/28          10.0.8.0/23
ACCEPT     all  --  10.0.9.0/28          10.0.7.0/28

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@andrea-VirtualBox:/home/andrea#
```

Fig. 34] Firewall sul router

A questo punto la rete2 può contattare la rete3 ma non la rete1.



```
root@andrea-VirtualBox:/home/andrea# ip netns exec rete2 bash
root@andrea-VirtualBox:/home/andrea# ping 10.0.9.1
PING 10.0.9.1 (10.0.9.1) 56(84) bytes of data.
64 bytes from 10.0.9.1: icmp_seq=1 ttl=63 time=0.057 ms
64 bytes from 10.0.9.1: icmp_seq=2 ttl=63 time=0.060 ms
64 bytes from 10.0.9.1: icmp_seq=3 ttl=63 time=0.041 ms
64 bytes from 10.0.9.1: icmp_seq=4 ttl=63 time=0.042 ms
64 bytes from 10.0.9.1: icmp_seq=5 ttl=63 time=0.042 ms
^C
--- 10.0.9.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4110ms
rtt min/avg/max/mdev = 0.041/0.048/0.060/0.010 ms
root@andrea-VirtualBox:/home/andrea# ping 10.0.7.1
PING 10.0.7.1 (10.0.7.1) 56(84) bytes of data.
^C
--- 10.0.7.1 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4099ms

root@andrea-VirtualBox:/home/andrea#
```

Fig. 35] tentativi di ping da parte della rete2 prima verso rete3 poi verso rete1

FASE DI ATTACCO

Una volta configurato l'intero sistema, la fase di attacco risulta in questo caso molto semplice.

Utilizzando un software di spoofing, come Wireshark, un utente della rete2 può facilmente individuare gli indirizzi che hanno accesso alla rete1 e assegnarsene uno.

In questo caso tutti gli indirizzi appartenenti alla rete 10.0.9.0/28 hanno accesso alla rete1, dunque l'attaccante può assegnarsi per esempio l'indirizzo IP 10.0.9.6/23 e avere di conseguenza accesso a sua volta alla rete1.

```
root@andrea-VirtualBox:/home/andrea# ifconfig
veth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.9.6  netmask 255.255.254.0  broadcast 10.0.9.255
        inet6 fe80::ccf7:54ff:fea3:2df  prefixlen 64  scopeid 0x20<link>
        ether ce:f7:54:a3:02:df  txqueuelen 1000  (Ethernet)
        RX packets 12  bytes 936 (936.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 936 (936.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@andrea-VirtualBox:/home/andrea#
```

Fig. 36] Assegnazione del nuovo IP alla rete2

A questo punto l’utente nella rete2 possiede un indirizzo IP che soddisfa le regole impostate nel firewall.
Nonostante ciò la funzionalità di reverse path filtering del kernel Linux è attiva e impedisce all’attaccante di raggiungere la rete1. Infatti tale funzione ha proprio lo scopo di filtrare attacchi di tipo IP spoofing; il kernel Linux elimina tutti i pacchetti relativi a un indirizzo IP sorgente che vengono ricevuti su un’interfaccia di rete diversa da quella che permette di raggiungere la destinazione secondo le sue tabelle di routing.
Dunque per proseguire con l’attacco è necessario disattivare il reverse path filtering, per farlo occorre utilizzare i seguenti comandi sul router:

```
sysctl -w net.ipv4.conf.all.rp_filter=0

sysctl -w net.ipv4.conf.veth1-R.rp_filter=0

sysctl -w net.ipv4.conf.veth2-R.rp_filter=0

sysctl -w net.ipv4.conf.veth3-R.rp_filter=0
```

Adesso l’utente nella rete2 riesce nuovamente a contattare la rete1 nonostante il firewall attivo nel router.

Adesso l’attaccante può trasmettere i pacchetti con un IP falso appartenente alla rete3 verso la rete1. Dunque effettuando il ping dalla rete2 alla rete1 i pacchetti attraverseranno il router e arriveranno alla rete1, quest’ultima risponderà con messaggi icmp indirizzati all’IP fasullo e dunque alla rete3.

Dunque a questo punto l’attaccante può eseguire un attacco DoS e colpire in modo del tutto anonimo sia la rete1 sia la rete3.
La Fig.37] mostra l’esecuzione dell’attacco da parte dell’utente nella rete2 verso la rete1 e la rete3.

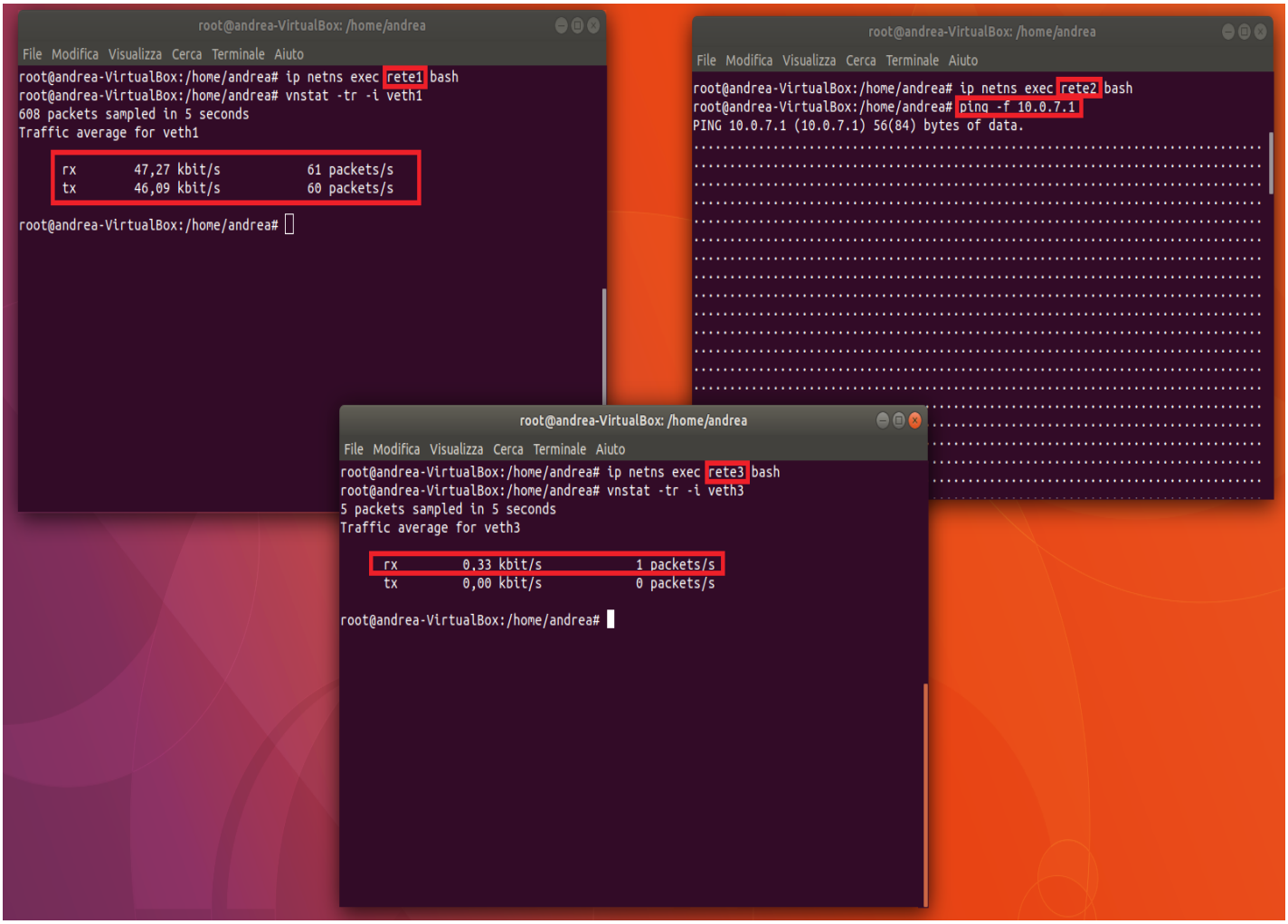


Fig. 37] Simulazione di un attacco DoS da parte di un utente della rete2 verso la rete1 e la rete3

FASE DI DIFESA

In questo caso la vulnerabilità del firewall stava proprio nella regola iptables poco specifica. Infatti, specificando solo l'indirizzo IP nella regola, è stato facile per l'attaccante aggirare il sistema di sicurezza cambiando il proprio IP con uno a cui era concesso accedere alla rete1.

Una regola più specifica avrebbe evitato ciò, per esempio aggiungendo da quale interfaccia di rete deve provenire il traffico.
Si può realizzare tale regola aggiungendo l'opzione **-i veth3-R** per il traffico proveniente dalla rete3 verso la rete1, e **-i veth1-R** per quello proveniente dalla rete1 e diretto verso la rete3. In questo modo il traffico proveniente dalla rete2 verso la rete1 non verrà accettato in quanto proveniente dalla interfaccia di rete veth2-R.

Dunque si può applicare tale soluzione modificando il firewall nel router nel seguente modo.

```
root@andrea-VirtualBox:/home/andrea# ip netns exec router bash
root@andrea-VirtualBox:/home/andrea# iptables -F
root@andrea-VirtualBox:/home/andrea# iptables -A FORWARD -s 10.0.8.0/23 -d 10.0.9.0/28 -j ACCEPT
root@andrea-VirtualBox:/home/andrea# iptables -A FORWARD -s 10.0.7.0/28 -i veth1-R -d 10.0.9.0/28 -j ACCEPT
root@andrea-VirtualBox:/home/andrea# iptables -A FORWARD -s 10.0.9.0/28 -d 10.0.8.0/23 -j ACCEPT
root@andrea-VirtualBox:/home/andrea# iptables -A FORWARD -s 10.0.9.0/28 -i veth3-R -d 10.0.7.0/28 -j ACCEPT
root@andrea-VirtualBox:/home/andrea#
root@andrea-VirtualBox:/home/andrea# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  10.0.8.0/23          10.0.9.0/28
ACCEPT     all  --  10.0.7.0/28          10.0.9.0/28
ACCEPT     all  --  10.0.9.0/28          10.0.8.0/23
ACCEPT     all  --  10.0.9.0/28          10.0.7.0/28

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@andrea-VirtualBox:/home/andrea#
```

Fig. 38] Impostazione del firewall sul router con le nuove regole

A questo punto i pacchetti con falso IP provenienti dall'interfaccia veth2-R e diretti verso la rete1 non verranno accettati e l'attacco non avrà successo.

È possibile verificare ciò effettuando nuovamente l'attacco dalla rete2 verso la rete1.
Come si può osservare nella Fig. 39], i pacchetti provenienti dalla rete2 non arrivano alla rete1 e questo è dovuto alle modifiche effettuate nel firewall posto sul router.

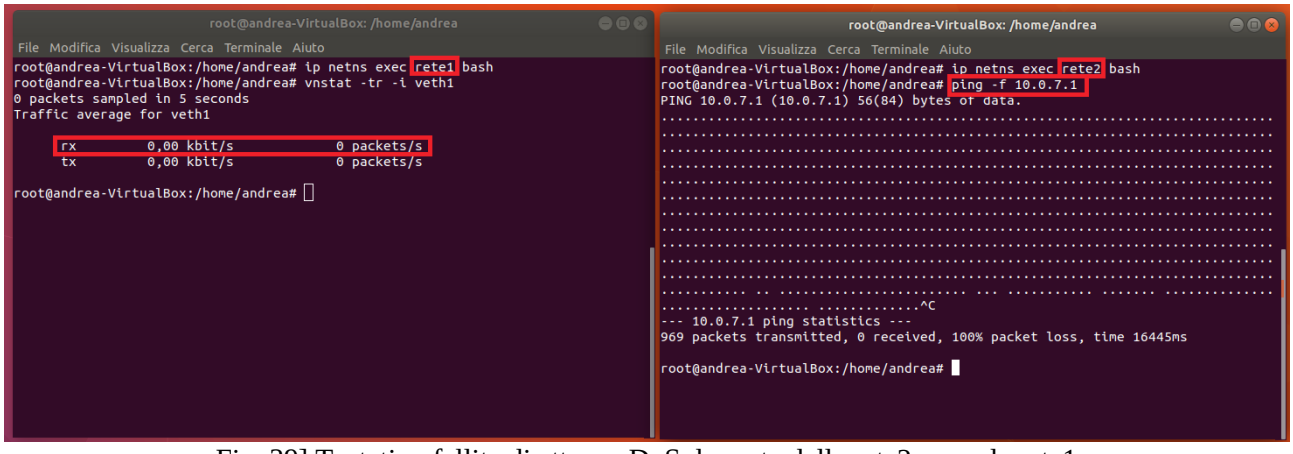


Fig. 39] Tentativo fallita di attacco DoS da parte della rete2 verso la rete1

ELENCO COMANDI USATI

%%Creazione delle reti e del router
ip netns add rete1
ip netns add rete2
ip netns add rete3
ip netns add router

%%Creazione delle interfacce di rete virtuali
ip link add veth1 type veth peer name veth1-R
ip link add veth2 type veth peer name veth2-R
ip link add veth3 type veth peer name veth3-R

%%Assegnazione delle interfacce viruali ai Namespace corrispondenti
ip link set veth1-R netns router
ip link set veth2-R netns router
ip link set veth3-R netns router
ip link set veth1 netns rete1
ip link set veth2 netns rete2
ip link set veth3 netns rete3

%%Abilitazione delle interfacce virtuali
ip netns exec rete1 ip link set dev veth1 up
ip netns exec rete2 ip link set dev veth2 up
ip netns exec rete3 ip link set dev veth3 up
ip netns exec router ip link set dev veth1-R up
ip netns exec router ip link set dev veth2-R up
ip netns exec router ip link set dev veth3-R up

%%Assegnazione degli indirizzi Ip
ip netns exec rete1 ifconfig veth1 10.0.7.1/28
ip netns exec rete2 ifconfig veth2 10.0.8.1/23
ip netns exec rete3 ifconfig veth3 10.0.9.1/28
ip netns exec router ifconfig veth1-R 10.0.7.2/28
ip netns exec router ifconfig veth2-R 10.0.8.2/23
ip netns exec router ifconfig veth3-R 10.0.9.2/28

%%Creazione delle tabelle di routing
ip netns exec rete1 bash
route add -net 10.0.8.0/23 gw 10.0.7.2
route add -net 10.0.9.0/28 gw 10.0.7.2
ip netns exec rete2 bash
route add -net 10.0.7.0/28 gw 10.0.8.2
route add -net 10.0.9.0/28 gw 10.0.8.2
ip netns exec rete3 bash
route add -net 10.0.7.0/28 gw 10.0.9.2
route add -net 10.0.8.0/23 gw 10.0.9.2

%%Abilitazione del forward nel router e configurazione del firewall
ip netns exec router bash
sysctl -w net.ipv4.ip_forward=1

```
iptables -P FORWARD DROP
iptables -A FORWARD -s 10.0.8.0/23 -d 10.0.9.0/28 -j ACCEPT
iptables -A FORWARD -s 10.0.7.0/28 -d 10.0.9.0/28 -j ACCEPT
iptables -A FORWARD -s 10.0.9.0/28 -d 10.0.8.0/23 -j ACCEPT
iptables -A FORWARD -s 10.0.9.0/28 -d 10.0.7.0/28 -j ACCEPT
```

CONCLUSIONI

L’esempio appena descritto mostra una applicazione pratica del principio del privilegio minimo trattato nel paragrafo 3.2.
È infatti buona norma utilizzare nel firewall delle regole specifiche, in modo che solo il traffico desiderato attraversi il router.
In particolare un firewall di tipo packet filter gestisce il filtraggio sulla base di; indirizzi IP, numeri di porta, tipo di protocollo utilizzato e interfacce di rete. L’ideale sarebbe inserire tutti questi parametri nelle varie regole del firewall.
In questo esempio ciò che rendeva il firewall vulnerabile era proprio la mancanza di uno di questi parametri, ovvero l’interfaccia di rete dalla quale dovevano provenire i pacchetti indirizzati alla rete1. Aggiungendo nella regola tale opzione tale vulnerabilità del firewall è stata risolta.

6.3 Attacco a sistema con Port knocking

In quest'ultimo esempio verrà mostrato come una cattiva configurazione del servizio di Port knocking sia vulnerabile ad attacchi esterni.
In particolare verrà eseguito il servizio knockd su un raspberry con IP: 172.20.10.3, mentre il dispositivo attaccante sarà un PC con indirizzo IP: 170.20.10.2.
L'obbiettivo dell'attaccante sarà quello di accedere al servizio SSH del raspberry, il quale è "nascosto" dal Port knocking.

Innanzitutto occorre configurare il servizio knockd.
Per farlo sono stati realizzati due script, uno per aprire la sessione ssh e uno per chiuderla:

```
#/bin/knock-open
iptables -I INPUT 1 -s $1 -p tcp -m tcp --dport 22 -j ACCEPT
```

```
#/bin/knock-close
iptables -D INPUT -s $1 -p tcp -m tcp --dport 22 -j ACCEPT
```

Tali script verranno eseguiti solamente una volta completata la sequenza di porte corretta.

Dunque il passo successivo consiste nel configurare i parametri del servizio knockd.

```
[options]
        logfile = /var/log/knockd.log
[opencloseSSH]
        sequence      = 7000,8000,9000
        seq_timeout   = 10
        tcpflags      = syn
        start_command =sh /bin/knock-open.sh %IP%
        cmd_timeout   = 20
        stop_command  =sh /bin/knock-close.sh %IP%
```

Fig. 40] Configurazione di knockd.conf

In questo caso la sequenza scelta è 7000, 8000, 9000 e il tempo massimo per completarla è di 10 secondi. Una volta completata correttamente la sequenza verrà eseguito lo script knock-open e la porta 22, relativa al servizio SSH, verrà aperta tramite l'inserimento dinamico di una nuova

regola iptables. Passati 20 secondi la porta 22 verrà nuovamente chiusa tramite l'esecuzione del secondo script che elimina la regola precedentemente inserita.

FASE DI ATTACCO

Durante la cosiddetta fase di scanning, l'attaccante esegue il comando nmap sul dispositivo vittima in cerca della porta ssh a cui desidera accedere.

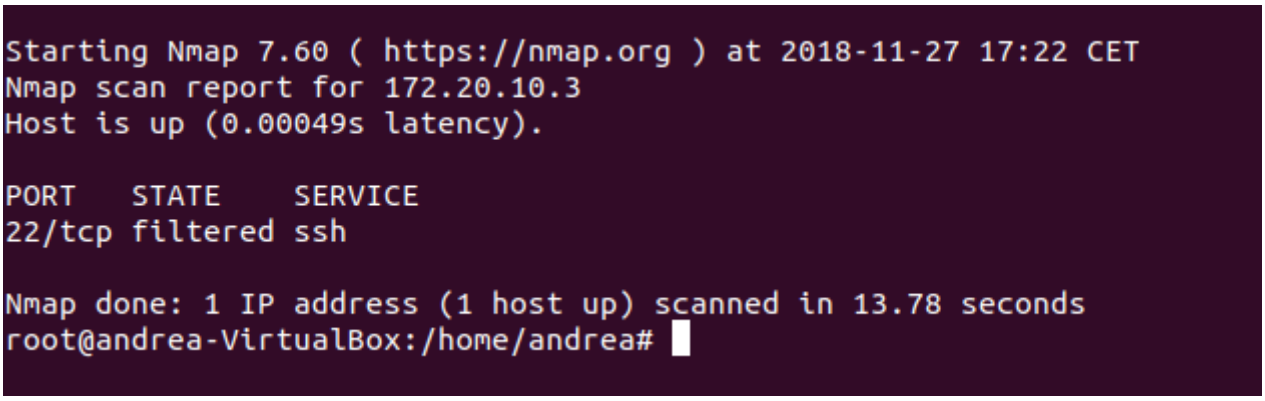


Fig. 41] Fase di scanning con il comando nmap

L'attaccante osserva che la porta 22 è filtrata e dunque inaccessibile. Nonostante ciò l'attaccante è a conoscenza del servizio di Port knocking attivo su tale porta (per realizzare tale esempio si è supposto essere l'attaccante a conoscenza di tale servizio), dunque decide di tentare l'accesso indovinando la sequenza corretta di porte.

Siccome l'utente che sta eseguendo l'attacco non è a conoscenza di tale sequenza decide di contattare le porte dalla 6000 alla 9000 con un passo di 250 fra una porta e la successiva.

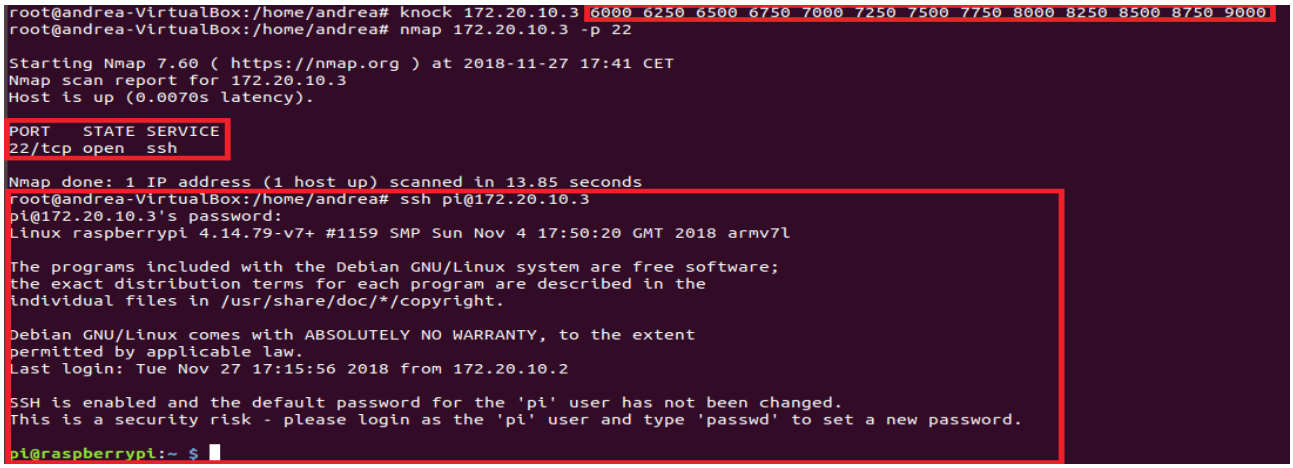


Fig. 42] Tentativo riuscito di aggirare il port knocking

La Fig. 42] mostra che nonostante solo 3 delle 13 porte contattate dall’attaccante siano effettivamente quelle della sequenza corretta, l’utente è riuscito ad aggirare il Port knocking e ad accedere al servizio SSH e dunque al raspberry.

Una volta penetrato nel sistema, l’attaccante ha pieno controllo della macchina vittima e può dunque facilmente disabilitare il servizio knockd attivo sulla porta 22.

FASE DI DIFESA

La fase di difesa consiste nel prevenire tale attacco. Infatti se l’attaccante fosse già entrato nel sistema potrebbe essere ormai troppo tardi per rimediare, infatti l’attaccante avrebbe ormai pieno controllo del dispositivo.

Dunque l’idea potrebbe essere quella di configurare meglio il servizio knockd. Infatti si può osservare che il successo dell'attacco è avvenuto grazie alla cattiva configurazione del file knockd.conf, in particolare il parametro sequence presentava 3 soli numeri e perlopiù in ordine crescente. L’attaccante ha sfruttato tale vulnerabilità contattando diverse porte entro un certo range e, pur non conoscendo l’esatta sequenza, è riuscito ad aprire la porta 22. Infatti il servizio di Port knocking è in grado di riconoscere quando la corretta sequenza viene emessa, ma non prende provvedimenti se fra una porta e la successiva della sequenza vengono contattate altre porte differenti.

Dunque il primo accorgimento da prendere consiste nell’impostare una sequenza che non sia né in ordine crescente né in ordine decrescente, per esempio; 6553, 5010, 9000, 7503, 8055.

Inoltre per evitare di concedere un numero troppo elevato di tentativi all’attaccante si può pensare di ridurre il tempo massimo, seq_timeout, per l’inserimento della sequenza a 1 secondo.

Configurando in questo modo il servizio knockd un utente non autorizzato avrà meno probabilità di accedere al servizio SSH.

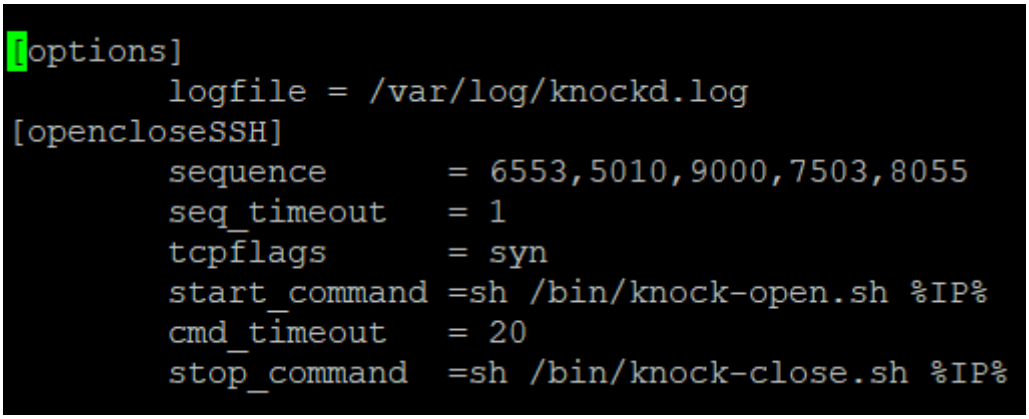


Fig. 43] Nuova configurazione del file knockd.conf

Tale configurazione del file knockd.conf renderà il servizio di port knocking più efficiente e meno facile da aggirare.

CONCLUSIONI

Grazie alla configurazione appena descritta del servizio knockd, eventuali tentativi di intrusione da parte di utenti della rete poco esperti non avranno senz'altro successo. Nonostante ciò l'ideale sarebbe modificare la sequenza di porte ogniqualvolta si effettui un accesso, in tal modo anche utenti più esperti che tentano di individuare la sequenza tramite l'utilizzo di software di sniffing non saranno in grado di effettuare un accesso illegittimo, questo in quanto la sequenza individuata non sarà più valida per l'accesso successivo.
Per rendere la sequenza di porte diversa ad ogni accesso il servizio knockd dà la possibilità di realizzare un file di testo contenente una lista di sequenze che si intende usare. Inserendo al posto del parametro sequence un nuovo parametro chiamato **one_time_sequences** e assegnandogli il percorso del file di testo contenente la lista di sequenze, ogni volta che si effettuerà un accesso la sequenza utilizzata verrà sostituita con quella successiva nel file di testo. In questo modo tale parametro cambierà dinamicamente e non sarà più possibile per un eventuale attaccante individuare la sequenza corretta.

```
one_time_sequences = /etc/knockd/smtp_sequences
```

L’unico inconveniente di tale soluzione è dovuto al fatto che ogni PC dovrà avere una copia di tale file di testo, senza di essa risulterebbe impossibile ricondursi alla sequenza corretta, specialmente se la lista è stata creata in modo casuale.

7. CONCLUSIONI E RINGRAZIAMENTI

Con il passare degli anni e con la continua espansione di internet, gli attacchi hacker diventeranno inevitabilmente sempre più insidiosi e numerosi, ma per questo motivo anche le tecniche di Cyber security saranno a loro volta sempre più all'avanguardia.
Con l'elaborato di tesi appena concluso spero di aver dato un'idea, seppur minima, dell'importanza della sicurezza informatica e dunque della protezione dei dati personali tramite software come i firewall.
Gli esempi del capitolo 6 in particolare avevano lo scopo di mostrare come semplici impostazioni del firewall possano fare la differenza e rendere il nostro sistema sicuro e difficilmente accessibile da utenti non autorizzati.

Il Port knocking, uno degli argomenti centrali di questa tesi, è un servizio con grandi potenzialità e il suo utilizzo può avere anche scopi che esulano dall'apertura e chiusura di porte. Infatti, una corretta sequenza eseguita da un utente remoto verso una macchina su cui è attivo il servizio di Port knocking potrebbe far compiere qualsiasi azione su tale macchina, per esempio avviare un programma.
Dunque il futuro di tale tecnica si prospetta molto interessante.

Infine concludo con dei doverosi ringraziamenti.
In particolare volevo ringraziare la mia famiglia per il grande sostegno che mi ha dato in questi anni di studio. Un ringraziamento va anche al mio relatore e professore Walter Cerroni, che mi ha seguito durante lo svolgimento di questo elaborato e mi ha aiutato a portarlo a termine.

8. BIBLIOGRAFIA

ONLINE

INTRODUZIONE
Che cos'è la cyber security:
http://www.businesspeople.it/Hi-Tech/Che-cos-e-la-cyber-security-103276

TIPI DI FIREWALL
What is a firewall:
https://www.cisco.com/c/it_it/products/security/firewalls/what-is-a-firewall.html

PACKET FILTER
Firewall filtering cos'è, come funziona e le sue debolezze:
https://codexsprawl.wordpress.com/2016/11/14/firewall-packet-filtering-cose-come-funziona-e-le-sue-debolezze/

Packet filtering:
https://www.techopedia.com/definition/4038/packet-filtering

STATEFUL FIREWALL
Statefull inspection firewalls come funzionano e le loro debolezze:
https://codexsprawl.wordpress.com/2017/03/06/stateful-inspection-firewalls-come-funzionano-e-le-loro-debolezze/

CONFIGURAZIONE DEL FIREWALL
Best pratices for firewall rules configuration:
https://support.rackspace.com/how-to/best-practices-for-firewall-rules-configuration/

PORT KNOCKING
Portknocking:
http://www.portknocking.org/

How to use port knocking to hide your ssh daemon from attackers on ubuntu:
https://www.digitalocean.com/community/tutorials/how-to-use-port-knocking-to-hide-your-ssh-daemon-from-attackers-on-ubuntu

TIPI DI ATTACCHI
La sicurezza informatica tutte le tipologie di attacchi esterni:
https://vitolavecchia.altervista.org/la-sicurezza-informatica-tutte-le-tipologie-attacchi-esterni/

IMMAGINI

[1]https://steemit.com/technology/@skcbappy/what-is-a-firewall-why-should-you-use-a-firewall-1bfb2bf69022

[2]http://www.tecnes.com/tecnologie/Networking/Networking+e+sicurezza%3A+proxy%2C+inverse+proxy+e+firewall.html

[3]http://www.portknocking.org/

[4]https://vitolavecchia.altervista.org/la-sicurezza-informatica-tutte-le-tipologie-attacchi-esterni/

[5]https://vitolavecchia.altervista.org/la-sicurezza-informatica-tutte-le-tipologie-attacchi-esterni/

[6]https://www.theprohack.com/2009/07/getting-max-internet-speed-using-arp.html

[7]https://vitolavecchia.altervista.org/la-sicurezza-informatica-tutte-le-tipologie-attacchi-esterni/

[8]http://www.multimac.it/soluzioni_scheda_ita.php/nomeProdotto=Man_in_the_Middle/idcat=1/idsottocat=154/idprodotto=1655