UNIVERSITÀ DEGLI STUDI DI CATANIA

Dipartimento di Matematica e Informatica

Corso di Laurea Triennale in Informatica

Rosario Scalia

(X81000374)

Progetto di Internet Security

Anno Accademico 2018 - 2019

Indice

Capitolo 1

Introduzione ad IPSec

In questo capitolo si vuole fornire una disamina sintetica dello standard IPSec, l'idea di base è delineare le caratteristiche salienti del protocollo.

1.1 Nozioni di base sul protocollo

IPSec è una suite di protocolli atta a garantire sicurezza al livello di rete dello stack TCP/IP.

Dal punto di vista delle proprietà di sicurezza, IPSec fornisce autenticazione, integrità e confidenzialità al livello di rete.

I principali strumenti utilizzati da IPSec ,per garantire le proprietà di sicurezza citate, sono la crittografia e le tecniche di MAC (Message Authentication Code).

I protocolli che compongono la suite sono tre:

- **AH**, è il protocollo che garantisce autenticazione e integrità dei pacchetti.
- **ESP**, è il protocollo che garantisce confidenzialità e integrità dei pacchetti, opzionalemnte può fornire anche autenticazione.
- **IKE**, è il protocollo che permette la creazione di una comunicazione "sicura" fra 2 entità; nello specifico ,in questa fase, le due entità si autenticano a vicenda e si scambieranno dei parametri crittografici non prima di aver negoziato un canale sicuro dove eseguire questo scambio.

Al netto degli aspetti tecnici del protocollo, è di cruciale importanza il fatto che IPSec porti sicurezza al livello di rete della pila TCP/IP.

La conseguenza di quest'ultima argomentazione è che gli strati superiori ,dello stack di rete, "ereditano" delle funzionalità di sicurezza.

A conferma del discorso appena trattato, anche gli applicativi più arretrati ,da questo punto di vista, potranno usufruire di queste proprietà senza modificare una linea di codice.

1.2 Modalità di utilizzo del protocollo

IPSec può essere utilizzato in due modalità: Trasporto e Tunnel.

La modalità trasporto collega sempre due host e in base alla scelta fra AH ed ESP può fornire autenticazione, integrità ed eventualmente confidenzialità del payload del pacchetto IP.

La modalità tunnel collega sempre due router oppure un host ed un router (configurazione chiamata *roadwarrior*); nello specifico ,a seconda della scelta fra AH ed ESP, si può garantire autenticazione, confidenzialità ed integrità dell'intero pacchetto IP.

Per ottenere ciò, l'intero pacchetto IP viene incapsulato nel campo payload di un nuovo pacchetto IP andando a creare un tunnel virtuale fra i due end-point della comunicazione.

Oltre alle considerazioni di sicurezza, resta da fare anche una breve riflessione funzionale in merito alle due modalità; infatti con IPSec in trasporto abbiamo bisogno che gli intermediari ,della comunciazione, supportino tale protocollo; invece con la modalità tunnel non c'è questa necessità, questo a causa dell'incapsulamento dei pacchetti IPSec in pacchetti IP classici.

1.3 Funzionamento di base del protocollo

Il concetto cardine di IPSec è la *Security Association* (abbreviato in SA).

Nello specifico, una Security Association è una connessione unidirezionale fra due entità che permette una comunicazione sicura.

Tutte le SA sono mantenute in un database chiamato SAD, questo database deve essere presente in tutti gli host compatibili con IPSec.

I campi chiave di una SA sono:

- **SPI**, è un indice scelto dal mittente della SA; in sostanza questo indice serve per identificare in maniera univoca una SA nel SAD.
- **Finestra Anti-Replay**, è un campo impiegato per discernere fra un pacchetto replicato ed uno non replicato.
- **AH**, contiene gli algoritmi di autenticazione ed altre informazioni utilizzate da AH.
- **ESP**, contiene gli algoritmi di autenticazione e cifratura usati da ESP assieme ad ulteriori informazioni utili a tale protocollo.
- **IPSec Protocol Mode**, indica se il pacchetto appartiene ad un flusso IPSec Tunnel o Trasporto.

Un'altro concetto alla base di IPSec sono le Security Policies.

Del resto, ogni pacchetto IP ,in un sistema dove è installato IPSec, ha applicata una security policy e sono proprio quest'ultime che determinano lo scarto del pacchetto, il processamento del pacchetto con IPSec oppure il processamento del pacchetto senza IPSec.

Tutte le security policy sono archiviate in un'ulteriore database chiamato SPD, presente in tutte le implementazione di IPSec.

Entrando nello specifico, una security policy permette di associare un sottoinsieme del traffico IP ad una Security Association, inoltre si può raffinare la selezione imponendo che l'associazione debba essere vincolata ad uno specifico protocollo di livello superiore.

Detto ciò, di seguito viene illustrato ,sommariamente, il funzionamento del protocollo dal punto di vista sia del mittente che del destinatario.

Funzionamento IPSec **mittente**:

1. I dati provenienti dai livelli superiori vengono confezionati in un pacchetto IP.
2. Successivamente, viene cercata una SPD corrispondente per il pacchetto in esame
   - Se la ricerca non restituisce risultati segue che il pacchetto viene scartato.
   - Se la ricerca va a buon fine, possono verificarsi i seguenti scenari:
     - Se la policy del pacchetto è `DISCARD` segue che il pacchetto verrà scartato.
     - Se la policy del pacchetto è `BYPASS` segue che il pacchetto verrà spedito come un classico pacchetto IP, questo implica che il pacchetto in esame non godra delle proprietà di sicurezza fornite da IPSec.
     - Se la policy del pacchetto è `PROTECT` segue che viene innescata una ricerca nel SAD, se la corrispettiva associazione viene trovata si passa alla fase successiva, altrimenti si invoca il protocollo IKE con l'obbiettivo di creare la Security Association mancante.
       Dopo questa fase, il pacchetto viene elaborato seguendo le indicazioni della SA e successivamente viene spedito al destinatario.

Funzionamento IPSec **destinatario**:

1. I livelli sottostanti consegnano al livello di rete un pacchetto.
2. Successivamente, viene effettuato un controllo sul tipo del pacchetto; del resto la computazione che subirà quest'ultimo sarà diversa a seconda che sia un pacchetto IPSec o un classico pacchetto IP.
   - Se il pacchetto non è di tipo IPSec viene innescata una ricerca nel database SPD, da questa ricerca possono venir fuori diversi scenari:
     - Se la ricerca non ottiene risultati oppure se la policy è `PROTECT` segue che il pacchetto viene scartato.
     - Se la ricerca va a buon fine e la policy è `BYPASS` segue che il pacchetto viene processato come un pacchetto IP canonico evitanto di applicargli le computazioni di IPSec.
   - Se il pacchetto in esame è un pacchetto IPSec viene innescata una ricerca nel SAD:

- Se la ricerca non ottiene risultati segue che il pacchetto viene scartato.
- Se la ricerca va a buon fine segue che il pacchetto viene computato secondo la Security Association; al termine di tale elaborazione vengono consegnati i dati del pacchetto ai livelli superiori.

1.4 Prevenzione agli attacchi di replica

Un'ulteriore caratteristica di IPSec è la tecnica di rilevazione degli attacchi di replica.

Il meccanismo anti replica di IPSec si snoda fra mittente e destinatario, nello specifico il mittente invia ogni pacchetto ,all'interno di una SA, con un numero progressivo che incomincia da uno; inoltre il mittente non può inviare più di $2^{32} - 1$ pacchetti al destinatario, infatti al superamento di questo limite viene chiusa l'associazione e ricreata con chiavi nuove.

Dal punto di vista del destinatario, il protocollo prescrive l'utilizzo di una finestra anti replica, tale finestra ha la seguente logica:

- Se il pacchetto ricevuto rientra all'interno della finestra, non è una replica ed è autenticato segue che il corrispettivo slot nella finestra viene marcato.
- Se il pacchetto è autenticato e cade alla destra della finestra segue che la finestra scorre a destra in modo tale che il pacchetto ricevuto rappresenti l'ultimo slot di quest'ultima, inoltre lo slot del pacchetto viene marcato.
- Se il pacchetto cade alla sinistra della finestra, non è autenticato oppure era già presente viene scartato.

1.5 Scenari di utilizzo di IPSec

Quest'ultima sezione vuole dare una panoramica sugli scenari reali di utilizzo di IPSec.

Nello specifico IPSec può essere usato nei seguenti contesti:

- Un host comunica in trasporto con un altro host presente su una LAN remota.
- Due gateway ,di LAN differenti, instaurano un tunnel in modo tale che gli host dietro tali gateway possano comunicare in maniera sicura.
- Un host remoto si connette in tunnel alla LAN aziendale.
- Due host della stessa rete comunicano in trasporto.

Capitolo 2

Implementazione di IPSec sui sistemi Linux

A questo punto della trattazione, si vuole fornire un minimo di conoscenza sulla strategia implementative di IPSec sui sistemi Linux, con questa strategia si vuole completare il pacchetto di conoscenza fornito dal capitolo 1 con una parta "pratica".

2.1 Strategia Implementativa su Linux

IPSec sui sistemi operativi GNU/Linux è implementato in larga parte lato kernel, ciò nonostante la gestione del protocollo IKE è lasciata ad un programma utente che comunica col kernel, la comunicazione avviene attraverso delle API messe a disposizione da quest'ultimo.

2.2 Stack Software Lato Kernel

Nel corso degli anni sono uscite varie implementazioni di IPSec lato kernel, nonostante ciò le più famose ed utilizzate restano KLIPS e Netkey.

Andando ad analizzare *KLIPS*, possiamo dire che questo software è la più diffusa implementazione di IPSec lato kernel, nonchè la più longeva.

Le peculiarità principali di KLIPS sono le seguenti:

- Device virtuali
- Caching dei pacchetti
- Path MTU Discovery

I *device virtuali* sono una tecnica usata da KLIPS per agganciare il suo funzionamento allo stack di rete del kernel Linux, in sostanza il software crea delle schede di rete virtuali dove vengono inoltrati i pacchetti da processore con IPSec; questa astrazione permette di definire in maniera chiara la pipeline di esecuzione del protocollo.

Il *caching dei pacchetti* è un ottima tecnica per ridurre le latenze, infatti KLIPS fa caching dei pacchetti che saranno coinvolti nella chiusura di un tunnel fra due entità.

In questo modo, gli utenti non noteranno alcun delay nella chiusura e riapertura del tunnel.

L'ultima caratteristica ,citata, di KLIPS è la tecnica che trova dinamicamente la dimensione massima dei frame ,del livello di collegamento, supportata da tutti gli intermediari della comunicazione, tale tecnica si chiama *Path MTU Discovery*.

Sostanzialmente, tale tecnica prevede l'invio di pacchetti inizialmente piccoli, all'aumentare dei pacchetti inviati correttamente segue anche un aumento della dimensione dei frame inviati (e di conseguenza anche dei pacchetti IP), questa ascesa della dimensione si ferma quando un nodo ,all'interno della catena, non supporta la dimensione dei frame ricevuti.

In tal caso, il nodo che non supporta la dimensione invierà un pacchetto ICMP al mittente; quest'ultimo ,a cascata, abbasserà la dimensione dei frame inviati (e quindi anche dei pacchetti IP) in modo tale che vengano soddisfatti tutti gli intermediari della comunicazione.

In ultima analisi, è giusto citare il mancato supporto ad IPv6 da parte di KLIPS, questa è una delle differenze principali con Netkey.

La seconda implementazione di IPSec ,lato kernel, citata è *Netkey*, questo software è molto più recente di KLIPS e per tanto contiene ancora qualche problematica in più.

Dal punto di vista del funzionamento, Netkey gestisce in maniera diversa IPSec rispetto a KLIPS; infatti qua non esistono device virtuali e l'astrazione è demandata ad un livello più basso.

La conseguenza di questo discorso è il "tipo di visibilità" che avra il sistema operativo nei confronti dei pacchetti IPSec inviati/ricevuti.

Più precisamente, il sistema operativo "vedrà" i pacchetti IPSec inviati in forma cifrata e i pacchetti IPSec ricevuti in chiaro.

2.3 Stack Software Lato Utente

Andando ad analizzare lo stack IPSec lato utente, possiamo dire che i due demoni più diffusi ,che gestiscono il protocollo IKE, sono *Pluto* e *Racoon*.

Pluto è il più rodato fra i due dato che ha subito più test e in generale si è dimostrato più affidabile nella gestione delle Security Association.

Oltre a ciò, Pluto ha una configurazione più semplice di Racoon; tutto ciò in virtù del supporto ,da parte di Pluto, alla mappatura automatica del file di configurazione alle due strutture dati fondamentali di IPSec, ovvero i database SPD e SAD.

2.4 Soluzioni IPSec Complete

Dal punto di vista delle soluzioni software complete, esistono vari pacchetti che combinano le componenti software lato kernel e lato utente viste nelle sezioni precedenti.

Il pacchetto più "storico" è *FreeSwan*, infatti il suo sviluppo iniziò nel 1997.

Più recentemente, FreeSwan è stato abbandonato e dalle sue ceneri ,più o meno in contemporanea, sono nati *OpenSwan*, *LibreSwan* e *StrongSwan*.

Capitolo 3

Attacco ad IPSec - Descrizione teorica

Nei capitoli precedenti sono stati analizzati gli aspetti teorici e pratici dello standard IPSec, in questo capitolo e nel successivo si andrà a delineare un attacco ad IPSec.

Nello specifico, nel capitolo corrente si andrà ad esemplificare l'attacco dal punto di vista teorico, invece nel successivo verrà mostrata la realizzazione pratica dell'attacco.

3.1 Spiegazione Attacco

Lo *scenario d'attacco* è un server IPSec connesso con l'attaccante attraverso una LAN, l'obbiettivo di quest'ultimo è instaurare una Security Association col server impersonando un altro host della LAN, assente al momento dell'attacco.

I *prerequisiti* dell'attacco sono i seguenti:

1. Il server deve utilizzare il protocollo IKE ,in versione 1, assieme alla modalità *Aggressive* del suddetto protocollo.

2. Il server deve utilizzare la *Pre-Shared Key* come tecnica di autenticazione nella *Fase 1* del protocollo IKE; la conseguenza di questo discorso è la presenza di un file nel server (protetto dalla policy del sistema operativo) contenente gli indirizzi IP dei mittenti che possono instaurare Security Association con quest'ultimo, accanto a questi mittenti ci sarà anche la pre-shared key, in chiaro.

3. Il server instaura Securiy Association esclusivamente con la vittima dell'attacco, questo significa che la vittima è l'unico mittente presente nel file descritto al punto precedente.

4. Il server risponde a richieste di connessioni IPSec (più precisamente richieste ISAKMP SA per negoziare una Security Association) da qualunque indirizzo IP, per generare la sua risposta userà la stessa chiave usata per la vittima.

5. Un'ulteriore conseguenza del punto 3 è la condivisione di un segreto fra il server e la vittima, ovvero la Pre-Shared Key.

   Del resto, la Pre-Shared Key è indispensabile alla vittima per instaurare l'associazione sicura col server.

Per riuscire a concretizzare l'attività malevola, l'attaccante ha bisogno di apprendere la Pre-Shared Key condivisa fra il server e la vittima.

Ebbene, verrà mostrato in seguito come l'attaccante ,sfruttando una debolezza del protocollo IKE, riesca a ricavare la chiave condivisa e di conseguenza instaurare una comunciazione sicura col server impersonando la vittima.

3.2 Spiegazione Tecnica Attacco

In questa sezione, si andrà ad esemplificare il funzionamento della parte del protocollo IKE coinvolta nell'attacco; successivamente si andrà ad illustrare la vulnerabilità che ha portato a quest'ultimo.

3.2.1 Funzionamento IKEv1 in Aggressive Mode con Pre-Shared Key

Come già detto nel primo capitolo, IKE è il protocollo ,appartenente ad IPSec, che si occupa di instaurare in maniera "sicura" le Security Association.

Nello specifico, questo protocollo ha due fasi:

- **Fase 1**, in questa fase i 2 host si autenticano ,a vincenda, e negoziano un canale sicuro sul quale successivamente "trattare" i parametri della futura Security Association, più tecnicamente i due peer vogliono instaurare una *ISAKMP SA*.
- **Fase 2**, in questa fase vengono negoziati i parametri della Security Association.

L'autenticazione reciproca fra gli host può avvenire in vari modi, i più diffusi sono la certificazione oppure la Pre-Shared Key che non è altro che un segreto condiviso fra 2 entità.

In questo contesto, si vuole porre l'attenzione sul funzionamento della fase 1 del protocollo dato che è proprio là che verrà effettuato l'attacco.

Detto ciò, la fase 1 della prima versione di IKE può essere configurata in 2 modi: *Main Mode* e *Aggressive Mode*.

La Main Mode è la modalità più sicura fra le due appena citate, di contro ,al livello di rapidità d'esecuzione, la Main Mode è più lenta rispetto all'Aggressive Mode; infatti la Fase 1 in Main Mode conta 6 messaggi ,scambiati fra i due peer, contro i 3 della Aggressive Mode.

Come già detto in precedenza, la debolezza sta nella combinazione di IKEv1 con l'Aggressive Mode e l'autenticazione con Pre-Shared Key.

Prima di descrivere la vulnerabilità, si vuole delineare il funzionamento della Fase 1 di IKE con la suddetta configurazione vulnerabile:

1. I -> R: (CKY_i , SA_i , g^i, N_i , ID_i)
2. R -> I: (CKY_r , SA_r , g^r , N_r , ID_r , h_r)

3. I -> R: (h_i)

In sostanza, il mittente invia al destinatario un messaggio con il seguente contenuto:

- Un Cookie ($\mathbf{CKY}_i$), è un numero generato randomicamente dal suo header ISAKMP.
- Una ISAKMP SA ($\mathbf{SA}_i$), ovvero un insieme di parametri di sicurezza proposti, questi parametri servono a costruire il canale sicuro fra le due entità.
- Un parametro derivato dall'esponenziazione di Diffie-Helmann ($\mathbf{g}^i$), questo valore è calcolato come α^{X_i} mod β.

  Più precisamente, α e β sono due parametri publici mentre X_i è un numero random tenuto segreto dal mittente.
- Una nonce ($\mathbf{N}_i$), ovvero un numero generato randomicamente.
- L'id del mittente ($\mathbf{ID}_i$), è un valore che identifica il payload.

Alla ricezione del messaggio *1*, il destinatario invierà un messaggio speculare a quello del mittente con l'aggiunta di un "nuovo" parametro h_r.

Il parametro h_r serve al mittente nella verifica d'integrità dei dati ricevuti dal destinatario, in oltre questo parametro permette l'autenticazione del destinatario col mittente.

Nello specifico, il calcolo del parametro h_r coinvolge una certa funzione hash f e viene eseguito dal destinatario con la seguente metodologia:

$$\begin{aligned} s &= f(pre\_shared\_key, (N_i, N_r)) \\ h_r &= f(s, (g^r, g^i, CKY_r, CKY_i, SA_r, ID_r)) \end{aligned} \tag{3.1}$$

Alla ricezione del messaggio *2* segue la verifica d'integrità di quest'ultimo da parte del mittente.

Più precisamente, il mittente deriva il parametro h'_r ,con lo stesso calcolo descritto sopra, e lo confronta con l'hash ricevuto.

Se gli hash sono uguali il messaggio non è stato alterato e si continua nella computazione; altrimenti il mittente fa un *abort* e conseguentemente interrompe la comunicazione col destinatario.

A questo punto, il mittente ,a meno che il messaggio non sia stato alterato, effettuerà 2 ulteriori azioni:

1. **Generazione della chiave finale**, il mittente genera la chiave finale k atta a cifrare i dati scambiati nella fase 2 di IKE.

   La procedura per generare la chiave è esemplificata di seguito:

$$\begin{aligned} s &= f(pre\_shared\_key, (N_i, N_r)) \\ sd &= f(s, (g^{ir}, CKY_i, CKY_r, 0)) \\ sa &= f(s, (sd, g^{ir}, CKY_i, CKY_r, 1)) \\ k &= f(s, (sa, g^{ir}, CKY_i, CKY_r, 2)) \end{aligned} \tag{3.2}$$

2. **Invio del messaggio 3**, il mittente genera h_i in modo speculare alla (3.1) e lo invia al destinatario in modo tale da potergli permettere la verifica d'integrità dei dati ricevuti al messaggio *1*.

Il destinario riceverà l'hash contenuto nel messaggio *3* e farà operazioni speculari a quanto fatto dal mittente alla ricezione del messagio *2*, ovvero verificherà l'integrita del messaggio ricevuto.

Se la verifica è andata a buon fine segue la generazione della chiave k ,da parte del destinatario, usando la (3.2).

Prima di chiudere questa sezione, si vuole sottolineare la totale assenza di crittografia nei messaggi scambiati nel protocollo appena descritto, sarà proprio questa mancanza a portare all'attacco.

Un'ulteriore sfaccettatura da sottolineare è il punto ,del protocollo appena descritto, in cui avviene la mutua autenticazione degli agenti, più precisamente quest'ultimi si autenticano ,a vicenda, solamente nei messaggi *2* e *3* (questo in virtù del segreto condiviso contenuto nel calcolo degli hash); di conseguenza non esiste nessuna forma di autenticazione nel messaggio *1*.

3.2.2 Vulnerabilità IKEv1 in Aggressive Mode con Pre-Shared Key

L'ultima sezione di questo capitolo illustra la tecnica usata dall'attaccante per impersonare un altro host nella connessione al server IPSec.

Nello specifico, l'attaccante, supponendo che la vittima sia offline, effettua le seguenti operazioni malevole:

- Imposta il suo indirizzo IP uguale a quello della vittima.
- Fa *port scanning* per rilevare la presenza del server IPSec.
- Assumendo che il server sia attivo, l'attaccante invia ,a quest'ultimo, una richiesta per instaurare una Security Association; nello specifico viene eseguito il protocollo IKE in Aggressive Mode e di conseguenza viene inviato al server il primo messaggio della fase 1 di tale protocollo.
- Il server ,come da policy, restituisce il classico messaggio *2* di IKE in Aggressive Mode.
- A questo punto, l'attaccante evita di rispondere al server ed estrae l'hash h_r dal messaggio; dopo di che effettua un attacco a dizionario ,offline, su quest'ultimo provando tutte le pre-shared key del dizionario.

  Nello specifico, l'attaccante effettua la seguente computazione:

```
1         for word in dict:
2             s = hash_function (word, (N_i , N_r ))
3             hypthetic_hash = hash_function (s, (g_r , g_i ,
                  CKY_r , CKY_i , SA_r , ID_r ))
4             if (hypothetic_hash == h_r):
5                 real_psk = word
6                 break
7             else:
8                 real_psk = -1
```

Se l'attaccante trova una pre-shared key che genera ,attraverso la computazione descritta sopra, un hash uguale a quello che ha ricevuto allora l'attaccante avrà scoperto la pre-shared key condivisa fra la vittima e il server.

La conseguenza del cracking della password è che l'attaccante ha guadagnato la possibilità di instaurare un canale sicuro col server impersonando la vittima.

A conferma dell'argomentazione descritta sopra, l'attaccante ,dopo aver appreso la chiave, reinoltrerà una richiesta ISAKMP SA al server; la differenza rispetto a prima è che ,ora, l'attaccante risponderà al server, più precisamente l'attaccante costruirà il messaggio *3* della fase 1 di IKE e di conseguenza porterà avanti l'instaurazione del canale sicuro col server, tutto ciò in virtù della chiave appresa.

Capitolo 4

Attacco ad IPSec - Demo

In questo capitolo verrà illustrato l'attacco dal punto di vista pratico.

4.1 Configurazione delle macchine

Ai fini della demo si è sfruttata la virtualizzazione utilizzando *Virtual Box*, nello specifico sono state create 2 macchine virtuali, una è dell'attaccante e l'altra è del server IPSec.

La macchina dell'attaccante monta *Kali Linux 2019.2* mentre la macchina del server monta *Ubuntu 12.04.02 LTS*.

Inoltre, le due macchine sono connesse alla stessa LAN; per poter ottenere questa proprietà è necessario impostare la stessa scheda di rete per entrambe le macchine, inoltre bisogna selezionare anche lo stesso nome dell'interfaccia di rete.

La configurazione di rete appena raccontata è descritta nelle seguenti immagini:

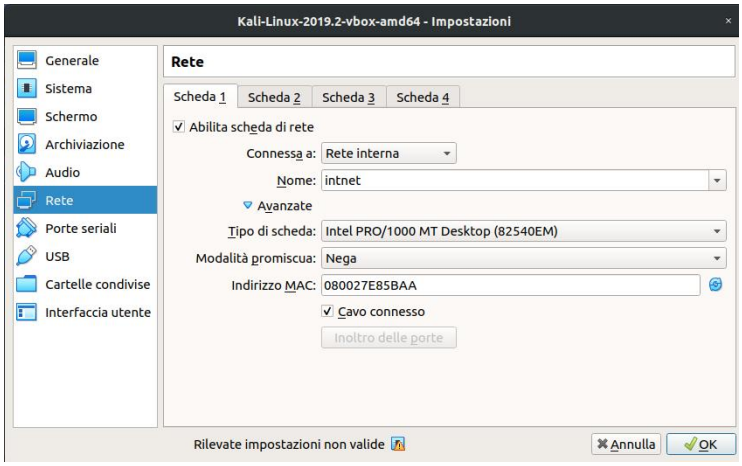

Figura 4.1: Configurazione di Rete Attaccante

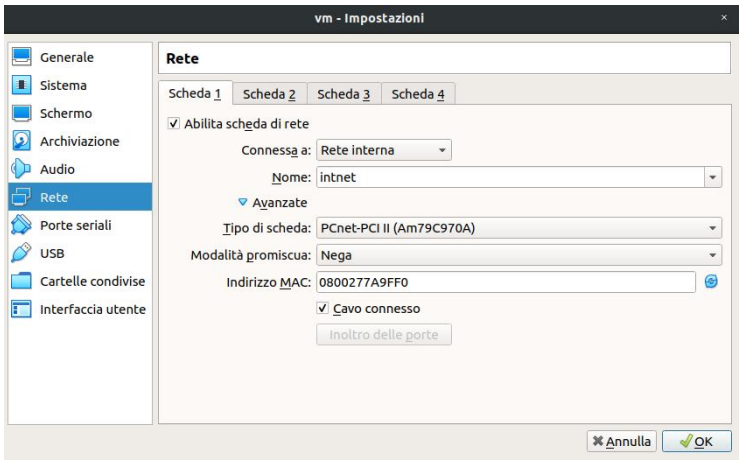

Figura 4.2: Configurazione di Rete Server IPSec

Dal punto di vista della configurazione del server, si è sfruttata la macchina virtuale scaricabile da [8], questa macchina virtuale porta in dote una versione di IPSec vulnerabile, nello specifico la suddetta macchina contiene *OpenSwan v1.2.6.37.1*.

Analogamente al server, anche la macchina dell'attaccante non è stata creata da zero ma si è usufruito della macchina virtuale messa a disposizione sul sito di Kali Linux.

A differenza del server, l'attaccante monta l'ultima versione di *OpenSwan* disponibile nel momento in cui si sta scrivendo, ovvero la versione *1.2.6.51.5*.

Detto ciò, i tool che hanno permesso di realizzare l'attacco sono 3: *nmap, ike-scan e psk-crack*.

Nmap permette di eseguire un *port scanning* su un host, invece *ike-scan* permette di inviare una richiesta ISAKMP SA ad un host e catturare la risposta di quest'ultimo; in sostanza questo tool ci è servito per catturare il messaggio *2* dell'Aggressive Mode di IKE.

L'ultimo tool mensionato è *psk-crack*, tale software è stato utile nell'esecuzione dell'attacco a dizionario.

Di seguito, vengono presentate le configurazioni di OpenSwan sulle due macchine virtuali:

Script 4.1: Configurazione OpenSwan Attaccante - file ipsec.conf

```
1     config setup
2         nat_traversal=yes
3         virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4
              :172.16.0.0/12
4         oe=off
5
6     conn vpn
7         authby=secret
8         pfs=no
9         auto=add
10        keyingtries=3
11        rekey=no
12        ikelifetime=8h
13        keylife=1h
14        type=transport
```

```
15          leftprotoport=17/1701
16          left=192.168.0.11
17          right=192.168.0.10
18          rightprotoport=17/1701
19          ike=aes256
20          esp=aes256-sha1
21          aggrmode=yes
```

Script 4.2: Chiavi segrete dell'attacante all'inizio - file ipsec.secrets

```
1 %any :    PSK ENTER_PSK_HERE
```

Script 4.3: Configurazione OpenSwan Server - file ipsec.conf

```
1     config setup
2         nat_traversal=yes
3         virtual_private=%v4:10.0.0.0/8,%v4:192.168.0.0/16,%v4
              :172.16.0.0/12
4         oe=off
5         protostack=netkey
6
7     conn vpn
8         authby=secret
9         pfs=no
10        auto=add
11        keyingtries=3
12        rekey=no
13        ikelifetime=8h
14        keylife=1h
15        type=transport
16        left=192.168.0.10
17        leftprotoport=17/1701
18        right=%any
19        rightprotoport=17/any
20        ike=aes256
21        esp=aes256-sha1
22        aggrmode=yes
```

Script 4.4: Chiavi segrete del server - file ipsec.secrets

```
1 192.168.0.10 192.168.0.11 :   PSK 123456
```

Prima di passare alla descrizione dell'attacco è giusto elencare i punti salienti delle configurazioni delle due macchine:

- In entrambi è impiegala l'aggressive mode di IKE 1.
- Entrambi usano IKE ed ESP usando AES-256 come algoritmo di cifratura.
- Entrambi usano SHA-1 come algoritmo di autenticazione, questo accade sia in IKE che in ESP.
- Come già detto in precedenza, il server risponde a richieste ISAKMP SA provenienti da qualunque indirizzo IP, la chiave usata nella risposta è la stessa della vittima.
- Entrambi hanno impostato la modalità trasporto di IPSec.

- Sia server che attaccante usano la Pre-Shared Key come tecnica di autenticazione.
- Il server instaura le Security Association solamente con l'host che possiede l'indirizzo IP *192.168.0.11/24* e la pre-shared key *123456*.

  Di conseguenza, server e host vittima condividono un segreto.
- Inizialmente, l'attaccante non conosce la Pre-Shared Key.

4.2 Descrizione attacco

Prima di descrivere l'attacco, si vuole mostrare lo stato delle interfacce di rete a macchine virtuali appena avviate:

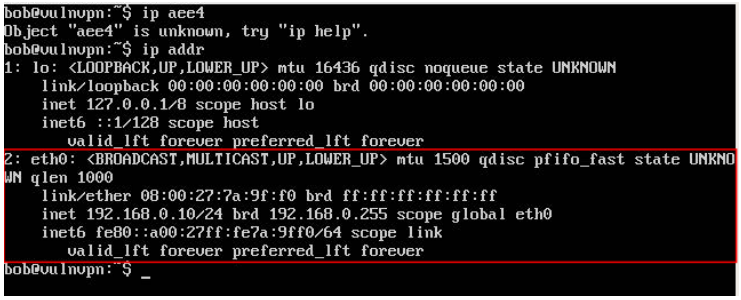

Figura 4.3: Interfacce di rete server all'avvio

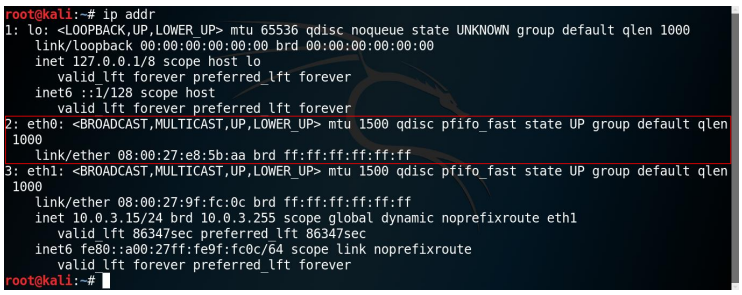

Figura 4.4: Interfacce di rete attaccante all'avvio

Come si può vedere dalle immagini, il server ha già un indirizzo IP preimpostato (*192.168.0.10/24*) a differenza dell'attacante.

A questo punto, l'attaccante imposta l'indirizzo IP della vittima (*192.168.0.11/24*) sulla sua scheda di rete:

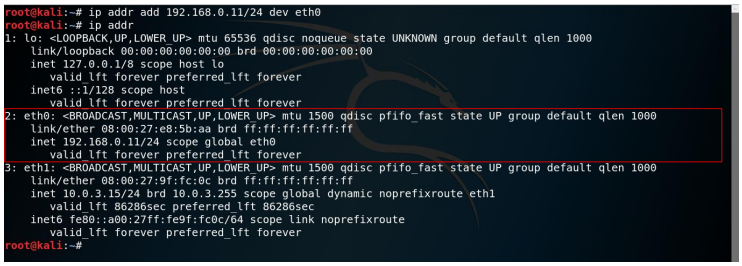

Figura 4.5: Impostazione indirizzo IP attaccante

Da questo momento, attaccante e server sono connessi alla stessa LAN.

Il passo successivo ,per l'attaccante, è effettuare un port scanning sul server in modo tale da capire se il servizio IKE è attivo su qualche porta:

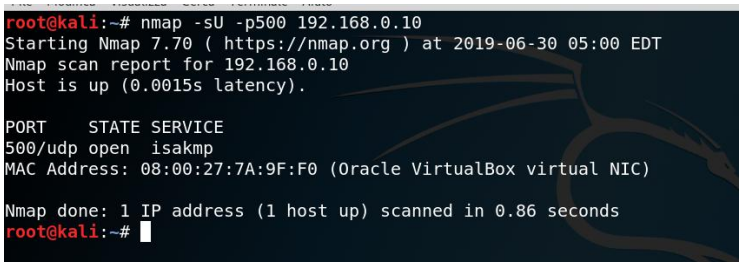

Figura 4.6: Port scanning attaccante

Come si può notare dall'immagine, il server ha attivo ,sulla porta 500, un servizio IKE.

Dopo aver appreso quest'informazione, l'attaccante invia al server il primo messaggio del protocollo IKE ,in Aggressive Mode, e cattura la risposta al messaggio:

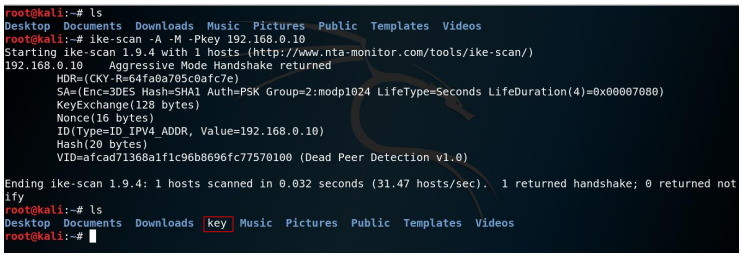

Figura 4.7: Messaggio catturato attaccante

Come si può notare dall'immagine, il tool *ike-scan* restituisce un file denominato "key" contenente l'hash inviatogli dal server, più precisamente ecco l'hash in questione:

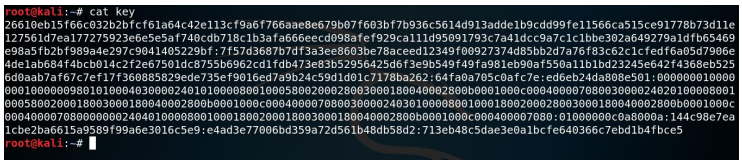

Figura 4.8: Hash catturato dall'attacante

A questo punto, l'attaccante monta un attacco dizionario ,offline, sull'hash appena discusso:

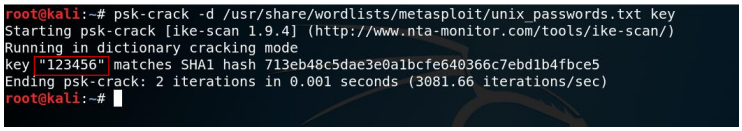

Figura 4.9: Attacco a dizionario sull'hash

Come si evince dall'immagine, la pre-shared key fra il server e la vittima era "123456".

Dopo aver appreso la chiave, non resta all'attaccante che inserirla nel file */etc/ipsec.secrets* e riavviare il servizio ipsec:

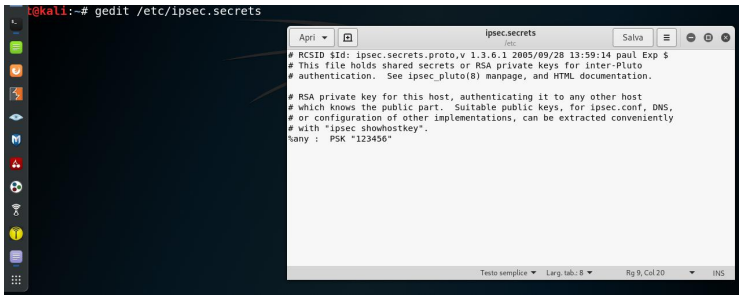

Figura 4.10: Inserimento pre-shared key vittima

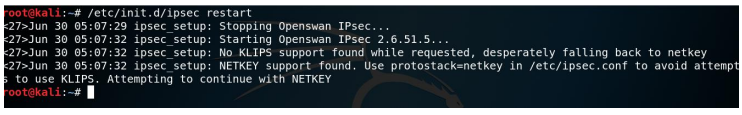

Figura 4.11: Riavvio servizio IPSec

L'attacco si conclude con la creazione ,indebita, della Securiy Association fra l'attaccante e il server, tutto ciò a discapito della vittima:

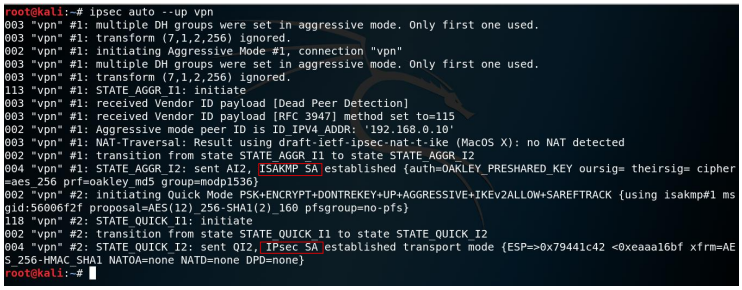

Figura 4.12: Connessione indebita attaccante

Capitolo 5

Riflessioni e Conclusioni

Questo capitolo porta in dote delle riflessioni in merito all'attacco, le riflessioni sono sia dal punto di vista delle conseguenze dell'attacco che dal punto di vista delle misure per arginarlo senza dimenticare di fornire una contestualizzazione di quest'ultimo.

5.1 Contestualizzazione e Conseguenze dell'attacco

L'attacco appena mostrato è figlio di una cattiva configurazione, nonostante ciò resta ancora realistico per una serie di motivi che per lo più hanno una matrice socio tecnologica.

Il primo motivo, è che la pre-shared key di IKE viene spesso e volentieri creata dagli amministratori di sistema anzichè farla creare ad un tool; tutto ciò porta spesso e volentieri ad una chiave debole.

Il secondo motivo, è che l'Aggressive Mode ,impiegata nell'attacco, è più rapida al livello di tempo di esecuzione rispetto alla "più sicura" Main Mode, per tanto qualche amministratore di sistema potrebbe essere tentato dall'usarla.

Inoltre, alcuni router Cisco ,con determinate versioni del software, possono essere forzati da un client ad utilizzare l'Aggressive Mode, come raccontato da [4].

Il terzo motivo, è che la configurazione delle Pre-Shared Key è più semplice da gestire rispetto a quella delle certificazioni ,che per inciso sono più sicure, e per tanto un amministratore di sistema potrebbe essere tentato da questa maggior semplicità.

Le conseguenze dell'attacco sono tanto ovvie quanto pericolose, infatti la chiave pre-condivisa ricavata dall'attaccante permette a quest'ultimo di impersonare la vittima nella comunicazione sicura col server.

Prima di chiudere questa sezione è importante puntualizzare in merito ad un'ulteriore vulnerabilità di IKEv1 in Aggressive Mode, la vulnerabilità in questione potrebbe permettere all'attaccante un attacco di negazione del servizio distribuita (DDOS) nei confronti di un host IPSec, tutto ciò a patto che tale host abbia IKE ,in versione 1, impostato in Aggressive Mode e risponda a richieste ISAKMP SA provenienti da qualunque indirizzo IP.

Del resto, il server risponde al messaggio iniziale ,della fase 1 di IKE, con un messaggio contenete l'esponenziazione di Diffie-Helmann, di conseguenza un numero elevato di richieste ISAKMP SA costringerebbe il server a fare tante esponenziazioni, operazione tutt'altro che leggera.

Tutto ciò ,alla lunga, porterebbe ad una negazione del servizio.

5.2 Riflessione sulle contromisure

La prima contromisure possibile ,all'attacco mostrato, è scegliere una pre-shared key robusta, magari generandola con un tool.

Un'ulteriore contromisura, potrebbe essere passare alla Main Mode di IKE, infatti questa modalità prevede la cifratura dei messaggi che contengono l'hash e per tanto risulterebbe più difficile all'attacante ricavare la pre-shared key.

Al netto delle contromisure appena dette, l'alternativa più efficace è l'utilizzo di uno schema di autenticazione alternativo come la certificazione.

5.3 Conclusioni

Questo progetto ha fortemente evidenziato l'importanza della scelta dei parametri di sicurezza e contestualmente ha permesso di esplorare la teoria oggetto del corso senza dimenticare la conoscenza pratica acquisita in merito ad IPSec.

Bibliografia

[1] William Stallings - Sicurezza delle reti, Applicazioni e standard

[2] Paul Wouters - OpenSwan

[3] John Pliam - Authentication Vulnerabilities in IKE and Xauth with Weak Pre-Shared Secrets

[4] Michael Thumann - PSK Cracking using IKE Aggressive Mode

[5] Steve Pitts - VPN Aggressive Mode Pre-shared Key Brute Force Attack

[6] Robert W. Beggs - Mastering Kali Linux for Advanced Penetration Testing

[7] Shoichi Sakane - Simple Configuration Sample of IPsec/Racoon

[8] Rebootuser - VulnVPN