IPS E PENETRATION TEST

Indice

1. INTRODUZIONE

1.1. **COS'È VMWARE E VSPHERE.** VMware è un software che permette di creare macchine virtuali, può essere utilizzato per creare un ambiente per effettuare test di sicurezza. L'intera struttura virtuale viene gestita tramite browser grazie allo strumento VSphere Client. Per questa simulazione la versione VMware utilizzata è la 5.5.

1.2. **CONFIGURAZIONE INIZIALE.** La rete ha questa struttura:

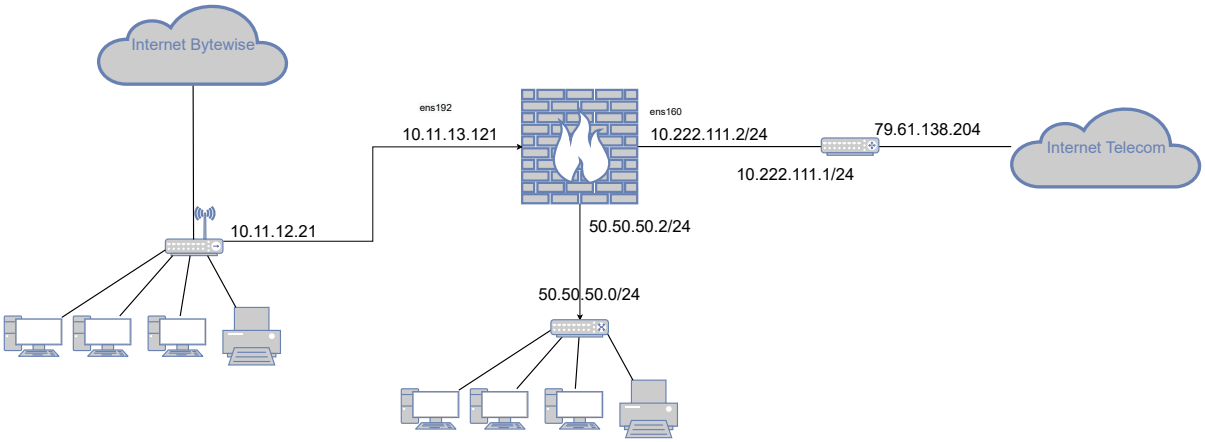

FIGURA 1.1. La rete

Un firewall con tre schede di rete:

- la 10.11.13.121 che si affaccia sulla rete bytewise con default gateway 10.11.12.21. L'interfaccia è ens192.
- la 10.222.111.2 che si affaccia sulla rete telecom con default gateway 10.222.111.1. Questo è il router che viene usato dal firewall quando l'host da raggiungere non è direttamente connesso. L'interfaccia è ens160.
- la 50.50.50.1 sulla lan 50.50.50.0/24 (*sul disegno compare l'IP sbagliato). L'interfaccia è ens224.
- la vm è visibile dall'esterno tramite l'IP pubblico 79.61.138.204

La macchina virtuale che ospita il **firewall** è stata installata con questa configurazione:
Le risorse assegnate alla vm sono:

- 8 core
- 32 GB ram
- 250 GB hdd

1.3. **FIREWALL.** Il firewall viene configurato manualmente, quindi è stato disabilitato firewalld (systemctl disable firewalld –now) e SELinux.
Il firewall si trova in /root/fwdir.

Algoritmo 1 vi /root/fwdir/flush

```
#!/bin/bash

iptables –F #flush delle catene di forward e input output
iptables –F –t nat #flush delle catene di pre–post routing
iptables –X #delete catene di forward

# politica
iptables –P INPUT ACCEPT
iptables –P OUTPUT ACCEPT
iptables –P FORWARD ACCPT
```

Algoritmo 2 vi /root/fwdir/fw

```
#!/bin/bash

#———————— FLUSH FIREWALL ————————#
/root/fwdir/flush
echo 1 > /proc/sys/net/ipv4/ip_forward

#————————POLICY————————#
iptables –P INPUT DROP
iptables –P OUTPUT DROP
iptables –P FORWARD DROP

#———————— FORWARD FILE————————#
/root/myfirewalldir/myfirewallForward

#———————— I/0 FILE ————————#
/root/myfirewalldir/myfirewall_IO
service iptables save
```

Questo file esegue prima il flush, poi i file in cui sono memorizzate le regole di forward e di input output. I file sono visibili su github nel folder personale di Filippo.

1.4. **SERVIZI ATTIVI SULLA MACCHINA.** I servizi attivi sulla macchina sono:

- Un server http apache con vulnerabilità remote code execution.
- MariaDB Version 5.5.68
  - passwd: mnkjoi09
- Samba Version 4.10.16
  - Guida: https://linuxize.com/post/how-to-install-and-configure-samba-on-centos-7/
  - TCP porte 139, 445
  - utente: josh
  - passwd: johnlennon
  - utente: sadmin
  - passwd: ringostarr

1.5. **COS'È SNORT.** Snort è un software di rilevazione e prevenzione delle intrusioni. I sistemi di rilevazione delle intrusioni (Intrusion Detection Systems, o IDS) individuano pacchetti sospetti, li registrano e ne

segnalano la presenza all'amministratore di rete. La loro naturale evoluzione consiste nei sistemi di prevenzione delle intrusioni (Intrusion Prevention Systems, o IPS), che intraprendono azioni attive sui pacchetti sospetti eliminandoli, resettando connessioni e/o bloccando interamente il traffico proveniente da determinati indirizzi IP.

1.6. **INSTALLAZIONE E CONFIGURAZIONE.** Snort è stato installato da sorgente:

- snort-2.9.19.tar.gz
- daq-2.0.7.tar.gz (data acquisition library)

La guida è disponibile tramite il link:
https://upcloud.com/community/tutorials/installing-snort-on-centos/

1.7. **TEST PRELIMINARE.** Un test preliminare è stato rilevare tutti i tentativi di connessione sulla porta 22. Bisogna aggiungere una regola nel file **/etc/snort/rules/local.rules**.

La regola è:

- **alert tcp any any -> $HOME_NET 22 (msg:"ssh test"; sid:10000001; rev:001;)**
  - snort manda un alert per i pacchetti che hanno source-ip: **any,** source-port: **any,** destination-ip: **10.222.111.2** (memorizzato nella variabile **$HOME_NET),** destination-port: **22**

Se un host esterno cerca di connettersi tramite la porta 22, il tentativo viene rilevato da snort.

È possibile far partire un'istanza di snort e far stampare a schermo i log digitando il comando:

- **snort -A console -i ens160 -u snort -g snort -c /etc/snort/snort.conf**
  - Nota: ens160 è l'interfaccia che corrisponde all'ip 10.222.111.2

Grazie ai log di snort e ai log di ssh su **/var/log/secure**, è possibile osservare come ogni secondo ci siano tentativi di login su ssh (ovviamente falliti) dovuti a script automatizzati. Per eliminare il problema, ssh è stato reso disponibile sulla porta 65022.

2. TENTATIVO 2

Il precedente tentativo è andato a mignotte senza nessun motivo apparente, quindi si è reso necessario procedere con una seconda installazione.

2.1. **INSTALLAZIONE E CONFIGURAZIONE.** Snort è stato installato tramite yum:

- yum install snort.x86_64

Per configurare Snort è stato modificato il file /etc/snort/snort.conf

In questo modo è stato possibile attivare il demone snortd.

È stato installato PulledPork package che permette di avere accesso alle community rules offerte da snort.

Guida:
https://support.redborder.com/hc/en-us/articles/209057125-Snort-on-CentOS-7-with-redBorder-Live-

Informazioni su DAQ, pcap:
http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node7.html

2.2. **CREAZIONE DI WEB SERVER NELLA DMZ.** È stato creato un web server nella DMZ. Questo rende necessario una modifica del firewall.

Sul web server sono installati i servizi:

- Httpd 2.4.6
- Tomcat 7.0.76
- MariaDB
- Samba

2.3. **UN CINESE IN CONNESSIONE CON SAMBA.** Dopo l'installazione e la configurazione di Samba è stato digitato il comando **netstat -tulnap** ed è stato visto che un IP address sconosciuto (proveniente dalla Cina) è riuscito ad instaurare una connessione con Samba non autorizzata. Questo perchè la versiona installata è vecchia e vulnerabile.

2.4. **INSTALLAZIONE KALI E OPENVAS.** È stata creata una macchina virtuale kali su cui è stato installato openvas. Openvas è un framework che include servizi e strumenti per la scansione e la gestione completa delle vulnerabilità (vulnerability assessment).

Openvas è in grado di fornire un report delle vulnerabilità dell'host, e un primo scan ha segnalato http come il servizio a maggiore rischio (vulnerabilità SSL-TLS):

Una possibile guida per installare Openvas:
https://www.geeksforgeeks.org/installing-openvas-on-kali-linux/

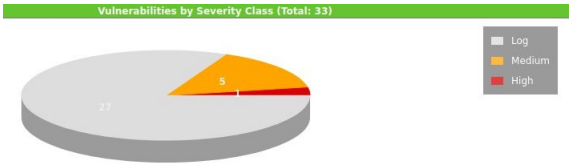

FIGURA 2.1. Vulnerabilities

2.5. **METASPLOIT E ATTACCO DOS.** Metasploit è un framework open source per lo sviluppo e l'esecuzione di exploits ai danni di un host remoto. Sulla vm Kali è possibile accedere alla console tramite il comando **msfconsole**.

Cercando la vulnerabilità rilevata da openVAS con il comando **search ssl tls** è possibile trovare uno script che effettua un attacco dos ai danni di un host.

Una possibile guida per i comandi base di metasploit:
https://www.makeuseof.com/beginners-guide-metasploit-kali-linux/

2.6. **IPS.** Snort **inline mode** permette di proteggere il sistema. Quando viene generato del traffico:

- legge il contenuto di pacchetti memorizzati in una coda
- legge una serie di regole iptables e in base ad esse decide se lasciarli passare o meno.

La guida usata è:
http://sublimerobots.com/2017/06/snort-ips-with-nfq-routing-on-ubuntu/
Questa guida è stata usata ma non funzionava... in the end è stato sfruttato FWsnort invece di nfqueue

2.7. **FILIPPO LAVORA SU IPS CON FWSNORT.**

- Snort ha le community rules,
- FWsnort è un progetto opensource che permette di tradurre le regole di snort in iptables eseguendo uno script,

Tra le guide usate:

(1) https://www.cipherdyne.org/fwsnort/
(2) https://linux.die.net/man/8/fwsnort

2.8. **CROWDSEC.** Per configurare crowdsec è stato disabilitato snort e fatto il flush del firewall.
La versione installata è 1.3.4
Guida per installare:
https://docs.crowdsec.net/docs/getting_started/install_crowdsec/