Índex

Enunciat

L'activitat que us proposem es un resum de tots els temes tractats respecte seguretat i problemes en les xarxes. Es composara de dues parts:
- Una primera part pràctica en la que haureu de dissenyar un sistema de seguretat complex que incloura algunes de les eines explicades en blocs anteriors.
- Una segona part que consistirà en un conjunt de preguntes que us posarem al moodle on haureu de respondre temes relacionats amb l'activitat, de forma que sera necessari que tots els membres del grup hagueu comprovat el funcionament de tota la pràctica.

La practica es realitzara durant les 3 següents sessions de classe en el laboratori DEIM6, ja que alli tindreu el material necessari per a fer les proves convenients. La pràctica estructurada en 5 seccions, les 4 primeres es poden comprovar en els vostres ordinadors, ja que son connexions 1 a 1 (recordeu-vos després que heu de guardar-ne la configuració). La 5a part consistirà en integrar tot aixo en un entorn real, i realitzar les configuracions finals de regles per a que tot funcioni.

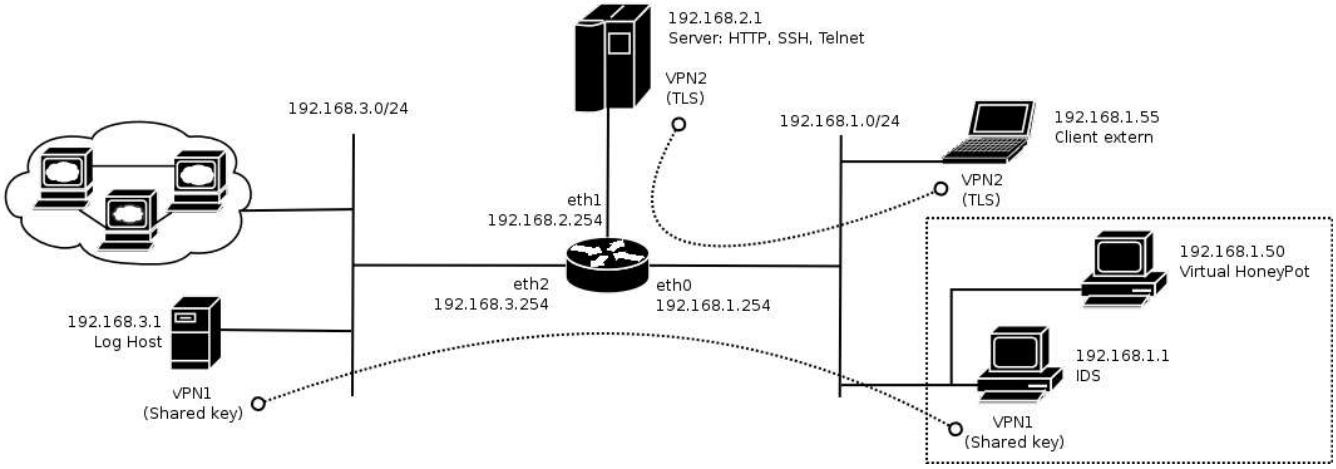
Les 5 parts en que esta estructurada la pràctica son:
- Configurar un servidor per a una DMZ.
- Configurar dues VPN, una de cada tipus.
- Configurar un honeyd monoposició amb diferents plugins per a fer-lo realista, i activar un snort perque en capturi tots els possibles atacs.
- Configurar que els logs del honeyd i del snort es guardin en un sistema remot.
- Connectar les 3 xarxes mitjançant un firewall i establir regles per a filtrar paquets.

Podeu trobar-ne mes detalls en el pdf de l'activitat. En el cas de que us surtin dubtes sobre la configuració d'alguna cosa, preneu voslatres mateixos les decisions adequades, sempre i quan siguin raonables.

Per a l'avaluació de la pràctica, ens haureu d'ensenyar en funcionament el vostre sistema muntat, i des d'un ordinador connectat a les diferents xarxes, comprovarem diferents elements:
- La sensació de realitat que doni el Honeypot.
- La correcta encriptació de le VPNs.
- L'accés unciament als serveis i ports que estan especificats.
- Que el sistema IDS capturi els possibles atacs i en guardi els logs a la màquina correcta.

Disseny

En base al disseny establert a l'enunciat de la pràctica, he generat el següent esquema més elaborat:



- *Tots els arxius de configuració o logs es troben adjunts a aquest informe i catalogats per IPs segons a la màquina a la que corresponen.*
- *S'ha utilitzat sistemes Ubuntu GNU /Linux (basat en Debian)*

Instal·lació/Configuració

Servidor 192.168.2.1

Servei Web

```
aptitude install webfsd
```

Editem /etc/webfsd.conf per indicar quin directori volem fer servir per les webs i quin port ha d'utilitzar:

```
# document root
web_root=/var/www
...
# port to listen on (default: 8000)
web_port=80
```

Al directori /var/www creem una plana web principal anomenada “index.html”.

Reiniciem el servei:

```
/etc/init.d/webfsd restart
```

Servei SecureSHell

```
aptitude install openssh-server
```

Editem /etc/ssh/sshd_config i ens assegurem que no es permet fer login a root:

```
PermitRootLogin no
```

Reiniciem el servei:

```
/etc/init.d/ssh restart
```

Servei Telnet

```
aptitude install inetutils-telnetd
```

Editem el fitxer /etc/inetd.conf per fer que el superdimoni inetd accepti connexions pel port 23 i cridi als servei instal·lat:

```
telnet      stream      tcp nowait      root        /usr/sbin/telnetd /usr/sbin/telnetd
```

Reiniciem el superdimoni:

```
/etc/init.d/inetd restart
```

Servei OpenVPN

```
aptitude install openvpn
```

Per crear els certificats, primer s'ha de crear una entitat certificadora ficticia. Editem /etc/ssl/openssl.cnf:

```
[ ca ]
default_ca  = CA_default

[ CA_default ]
dir     = /etc/ssl/hackCA  # Where everything is kept
certs       = $dir/certs  # Where the issued certs are kept
crl_dir     = $dir/crl  # Where the issued crl are kept
database    = $dir/index.txt  # database index file.
new_certs_dir   = $dir/newcerts     # default place for new certs

certificate = $dir/HackCa.crt     # The CA certificate
serial      = $dir/serial       # The current serial number
crl     = $dir/crl.pem      # The current CR
private_key = $dir/private/HackCa.key # The private key
```

Executem:

```
mkdir /etc/ssl/hackCA/
mkdir /etc/ssl/hackCA/certs
mkdir /etc/ssl/hackCA/private
mkdir /etc/ssl/hackCA/newcerts
mkdir /etc/ssl/hackCA/crl
echo "01" > /etc/ssl/hackCA/serial
touch /etc/ssl/hackCA/index.txt

cd /etc/ssl/hackCA/
openssl req -nodes -new -x509 -keyout private/HackCa.key -out HackCa.crt -days 365
```

Ara ja disposem de l'entitat i anem a generar les claus privades, sol·licituds de certificat i certificats signats tant pel servidor VPN com per un client VPN:

```
openssl req -nodes -new -keyout privateNet.key -out privateNet.csr
openssl ca -out privateNet.crt -in privateNet.csr

openssl req -nodes -new -keyout publicNet.key -out publicNet.csr
openssl ca -out publicNet.crt -in publicNet.csr
openssl dhparam -out dh1024.pem 1024
```

La clau i certificat “publicNet” només es per un client, s'ha de fer una clau per a cada nou client que es vulgui connectar. El traspàs de la clau cap al client s'ha de fer mitjançant un medi segur (e.g. sftp).

Creem el directori /etc/openvpn/hack i fiquem dintre els arxius: dh1024.pem, HackCert.pem, privateNet.crt i privateNet.key. Creem el fitxer /etc/openvpn/hack.conf:

```
dev tap0
proto udp
up /etc/openvpn/hack.up
tls-server
dh /etc/openvpn/hack/dh1024.pem
ca /etc/openvpn/hack/HackCert.pem
cert /etc/openvpn/hack/privateNet.crt
key /etc/openvpn/hack/privateNet.key
```

Creem el fitxer /etc/openvpn/hack.up:

```
#!/bin/bash
ifconfig tap0 down
ifconfig tap0 192.168.200.1 up
```

Ens assegurem que es carrega el mòdul “tun” al arranc ficant-lo al /etc/modules i el carreguem en aquest moment:

```
modprobe tun
```

Ja tenim preparat el servei:

```
/etc/init.d/openvpn start
```

La interfície virtual per on rebrem les dades del VPN serà “tap0”:

```
tap0      Link encap:Ethernet  HWaddr 00:FF:26:3D:47:F1
          inet addr:192.168.200.1  Bcast:192.168.200.255  Mask:255.255.255.0
          inet6 addr: fe80::2ff:26ff:fe3d:47f1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:238 (238.0 b)
```

Servidor de logs 192.168.3.1

OpenVPN

```
     aptitude install openvpn
```

Generem la clau compartida:

```
     openvpn --genkey --secret clau
```

Aquesta clau l'haurà d'utilitzar tot aquell que vulgui connectar amb el VPN, en el nostre cas només IDS/HoneyPod. S'haurà de traspasar la clau utilitzant un medi segur (e.g. sftp).

Creem /etc/openvpn/hack.conf:

```
dev tap0
proto udp
up /etc/openvpn/hack.up
secret clau
port 50000
comp-lzo
ping 15
verb 5
```

Creem /etc/openvpn/hack.up:

```
#!/bin/bash
/sbin/ifconfig tap0 192.168.100.1 netmask 255.255.255.0 broadcast 192.168.100.255
```

Ens assegurem que es carrega el mòdul “tun” al arranc ficant-lo al /etc/modules i el carreguem en aquest moment:

```
modprobe tun
```

Iniciem el servei:

```
/etc/init.d/openvpn start
```

Syslogd

Configurem el dimoni de logs per a que rebi logs remots.

Editem /etc/syslog.conf:

```
auth,authpriv.*         /var/log/auth.log
daemon.*            -/var/log/daemon.log
...
```

Editem /etc/init.d/syslogd (permetem la recepció de logs d'altres màquines):

```
SYSLOGD="-r"
```

Reiniciem el servei:

```
/etc/init.d/syslogd restart
```

IDS/HoneyPot 192.168.1.1/192.168.1.50

IDS Snort

Veure informe de l'activitat 5 (Detecció escaneig de ports). Es segueix el mateix procediment per la instal·lació i configuració, tenint en compte que considerem xarxa local 192.168.1.0/24.

Es important crear un usuari específic amb el qual s'executarà snort, d'aquesta forma si resulta vulnerable, l'atacant no aconseguirà automàticament permisos de root.

En aquest cas s'indicarà al fitxer /etc/snort/snort.conf que es vol que els logs passin a mans de syslog:

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

HoneyPot

Veure informe de l'activitat 6 (Activitat Honeyd). En aquest cas només es crea un HoneyPot virtual amb adreça 192.168.1.50 que emula un sistema GNU/Linux (concretament SuSE 8.0):

```
create suseLinux
```

```
set suseLinux personality "Linux Kernel 2.4.20"
add suseLinux tcp port 80 "sh /etc/honeypot/scripts/unix/linux/suse8.0/apache.sh"
add suseLinux tcp port 23 "sh /etc/honeypot/scripts/unix/linux/suse8.0/telnetd.sh"
add suseLinux tcp port 21 "sh /etc/honeypot/scripts/unix/linux/suse8.0/proftpd.sh"
add suseLinux tcp port 25 "sh /etc/honeypot/scripts/unix/linux/suse8.0/sendmail.sh"
set suseLinux default tcp action reset
set suseLinux default udp action reset
bind 192.168.1.50 suseLinux
```

Les versions dels serveis corresponen a una SuSE 8.0 real, per tant no dona lloc a pensar que no es tracta d'un servidor real.

Es important fer corre Honeyd de la forma més segura, per exemple utilitzant un chroot de forma que si l'aplicació es vulnerable, l'atacant no tindrà accés a tot l'arbre de directoris de la màquina.

Honeyd envia automàticament els seus logs a syslog sota la categoria “daemon”.

OpenVPN

Copiem a /etc/opevpn/ la clau compartida generada al servidor de logs i creem / etc/openvpn/hack.conf:

```
dev tap0
remote 192.168.3.1
proto udp
up /etc/openvpn/hack.up
secret clau
port 50000
comp-lzo
```

Creem /etc/openvpn/hack.up:

```
#!/bin/bash
/sbin/ifconfig tap0 192.168.100.2 netmask 255.255.255.0 broadcast 192.168.100.255
```

Ens assegurem que es carrega el mòdul “tun” al arranc ficant-lo al /etc/modules i el carreguem en aquest moment:

```
modprobe tun
```

Ja podem establir la connexió amb el servidor de logs:

```
/etc/init.d/openvpn start
```

Syslogd

Configurem el dimoni de logs per a que envii alguns al servidor de logs de la xarxa privada (sempre utilitzant el VPN).

Editem /etc/syslog.conf:

```
auth,authpriv.*         @192.168.100.1
daemon.*                @192.168.100.1
...
```

Editem /etc/init.d/syslogd (permetem l'enviament de logs a altres màquines):

```
SYSLOGD="-h"
```

Reiniciem el servei:

```
/etc/init.d/syslogd restart
```

Firewall

El firewall disposa de 3 interfícies de xarxa:

eth0 (192.168.1.254) – Connecta amb la xarxa pública (192.168.1.0/24)
eth1 (192.168.2.254) – Connecta amb la DMZ (192.168.2.0/24)
eth2 (192.168.3.254) – Connecta amb la xarxa privada (192.168.3.0/24)

Seguirem una política de negació per defecte, només es permeten les connexió especificades explícitament. Com a característiques generals:

- Acceptem connexions en estat ESTABLISHED o RELATED, d'aquesta forma acceptem automàticament el tràfic de resposta dels serveis permesos.
- Activem les següents opcions del kernel:
  - Kernel anti-spoofing protection
  - Ignore broadcast pings
  - Do not accept ICMP redirect
  - Ignore bogus ICMP errors
  - TCP SYN Cookies

Interfície eth0

- Denegem qualsevol tipus de tràfic que tingui com a font la IP del HoneyPot (192.168.1.50), aquesta màquina es virtual i no ha d'arribar tràfic seu a cap altra xarxa.
- Acceptem connexions des de la IP del IDS (192.168.1.1) al servidor de logs (192.168.3.1) port UDP 5000 del VPN.
- Acceptem connexions al servidor (192.168.2.1) si van dirigides al port 80 (HTTP), 22 (SSH) i 5000 UDP (VPN).
- Denegem la resta.

Interfície eth1

- Denegem tot el tràfic, des del servidor no s'han de realitzar connexions cap a l'exterior. Només es útil el tràfic ja acceptat a la interfície eth0 i les corresponents respostes.

Interfície eth2

- Acceptem tràfic procedent de la xarxa privada (192.168.3.0/24) que vagi dirigit al servidor als ports 80 (HTTP) i 22 (SSH).
- Acceptem tràfic procedent de la xarxa privada (192.168.3.0/24) que vagi dirigit a qualsevol IP però només dirigit al port 80 (HTTP). Probablement a la xarxa pública hi haurà un gateway amb

sortida a Internet, per tant permetem la navegació web des de la xarxa privada. No es permet cap altre mena de connexions de del interior, amb el pas del temps es pot adaptar a les necessitats de la xarxa habilitant nous serveis (e.g. FTP) sempre i quan siguin estrictament necessaris.
- Denegem la resta.

Utilitzant firewall builder 2.0.5 he creat el projecte corresponent i he generat l'arxiu necessari per configurar un sistema GNU/Linux amb iptables.

Proves

Client OpenVPN cap al DMZ

Des d'un client a la xarxa pública amb IP 192.168.1.55, creem el directori /etc/openvpn/hack i fiquem dintre els arxius: dh1024.pem, HackCert.pem, publicNet.crt i publicNet.key. Creem el fitxer /etc/openvpn/hack.conf:

```
dev tap0
remote 192.168.2.1
proto udp
up /etc/openvpn/vpn2/hack.up
tls-client
dh /etc/openvpn/hack/dh1024.pem
ca /etc/openvpn/hack/HackCert.pem
cert /etc/openvpn/hack/publicNet.crt
key /etc/openvpn/hack/publicNet.key
port 50000
comp-lzo
ping 15
verb 5
```

Creem el fitxer /etc/openvpn/hack.up:

```
#!/bin/bash
ifconfig tap0 down
ifconfig tap0 192.168.200.1 up
```

Ens assegurem que es carrega el mòdul “tun” al arranc ficant-lo al /etc/modules i el carreguem en aquest moment:

```
modprobe tun
```

Connectem el VPN:

```
/etc/init.d/openvpn start
```

Podem accedir correctament als serveis utilitzant la ip 192.168.200.2, viatjant les dades encriptades pel VPN.

IDS/HoneyPot/Logs remots

Des d'un client a la xarxa pública amb IP 192.168.1.55, faig un scaneig de ports a la IP 192.168.1.50:

```
# nmap -v -sS  192.168.1.50

Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2005-01-24 16:22 CET
Interesting ports on 192.168.1.50:
(The 1655 ports scanned but not shown below are in state: closed)
PORT   STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
25/tcp open  smtp
80/tcp open  http
Nmap run completed -- 1 IP address (1 host up) scanned in 1.989 seconds
```

Mirem els arxius de logs del servidor privat 192.168.3.1:

auth.log:

```
...
Jan 24 16:22:55 192.168.100.2 snort: Portscan detected from 192.168.1.55 Talker(fixed: 30 sliding: 30)
Scanner(fixed: 0 sliding: 0)
...
```

daemon.log:

```
...
Jan 24 16:23:28 192.168.100.2 arpd[14338]: arp reply 192.168.1.50 is-at 00:0e:35:91:13:fa
Jan 24 16:23:38 192.168.100.2 honeyd[14310]: Sending ICMP Echo Reply: 192.168.1.50 -> 192.168.1.55
Jan 24 16:23:39 192.168.100.2 honeyd[14310]: Killing unknown connection: tcp (192.168.1.55:37819 -
192.168.1.50:80)
Jan 24 16:23:39 192.168.100.2 honeyd[14310]: Killing attempted connection: tcp (192.168.1.55:37799 -
192.168.1.50:264)
Jan 24 16:23:39 192.168.100.2 honeyd[14310]: Killing attempted connection: tcp (192.168.1.55:37799 -
192.168.1.50:1452)
Jan 24 16:23:39 192.168.100.2 honeyd[14310]: Killing attempted connection: tcp (192.168.1.55:37799 –
192.168.1.50:887)
...
```

L'enviament de logs es correcte, el honeypot funciona perfectament i l'IDS detecta l'scan exitosament.

Alumne: Sergio Blanco Cuaresma