Honeypot

KHAOULA BLEL (RT3)
ARWA BEN HMIDA (RT3)
JIHENE HERGLI (GL3)
WALID ABID (RT3)
MOHAMED RIDHA KAANICH (RT3)
DORRA KHLIFI

Table des matières

1. Présentation de l'atelier

- Le terme de honeypot ou, en français, " pot de miel " est un principe consistant à utiliser des systèmes pour attirer et piéger les pirates informatiques par la ruse, afin notamment de collecter des informations sur leurs méthodes.
  Dans cet atelier on va créer un réseau qui contient un ensemble de machines représentant le système honeypot et qui seront utilisées comme un piège pour les pirates.

- Les honeypots sont divisés en deux catégories : les honeypots à faible interaction et les honeypots à forte interaction.

  Dans cet atelier on va s'intéresser aux honeypots à faible interaction .

2. Présentation des outils utilisés

-Honeyd est un outil Open Source développé par Niels Provos, il permet la configuration d'un réseau virtuel "pot de miel" et détecter ainsi toutes les interactions entre les machines du réseau et celles étrangères .
Honeyd peut en effet héberger jusqu'à 65536 hôtes virtuels ! Cela permet donc à l'utilisateur de créer de vastes architectures réseau complexes et longues à explorer.
Avec ce logiciel, on peut aussi créer des réseaux complets, c'est-à-dire avec la notion supplémentaire de « routage » : création de sous réseaux engendrant sous réseaux… L'émulation de « lenteur » ou de « pertes des données » est ainsi possible.

-Ubuntu :C'est un système d'exploitation intuitif et sécurisé, idéal pour les ordinateurs de bureau, les serveurs, les netbooks et les ordinateurs portables. Ubuntu est libre, gratuit, et est composé de logiciels qui le sont également

-nmap: C'est un scanner de ports libre créé par Fyodor et distribué par Insecure.org. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant.

a. Type de honeypot:

- honeypots de recherche sont utilisé pour collecter des informations sur les différent comportement et technique des pirates .Ensuite, analyser ses information pour découvrir les nouveaux menaces et apprendre à protéger le système

- honeypots de production sont utilisé pour protéger les entreprises en détournant les attaques des pirates vers des machines virtuelles. Ils sont implémentés dans le réseau de production pour améliorer leur sécurité .

b. Niveau d'interaction de honeypot:

Les honeypots sont principalement divisés en deux catégories : les honeypots à faible interaction et les honeypots à forte interaction.

- Niveau interaction faible émule des faux services réseaux qui sont vulnérables , il les simule par l'intermédiaire de script. Avec ce type d'honeypot, le pirate n'interagit jamais avec le système d'exploitation même s'il en a l'impression. La sécurité est donc ainsi conservée.
- Niveau interaction fort ne sont pas basés sur l'émulation de services ou de systèmes d'exploitation. Au contraire, ils reposent sur un vrai système d'exploitation où de véritables services, vulnérables ou non, tournent et sont accessibles aux pirates. Ainsi, la méthode d'approche est complètement différente puisque l'on offre au pirate la possibilité de rentrer dans le système et de faire ce qu'il lui plaît une fois le système compromis.

3.Configuration des outils

a) Installation:

Sous ubuntu on exécute la commande :

- ***apt-get install honeyd :*** Cette commande permettra d'installer honeyd sous linux .

Les principaux fichiers installés sous /etc/honeypot sont :

-nmap.prints

-nmap.assoc

-pf.os

-xprobe2.conf

-honeyd.conf : le fichier qui contiendra la configuration de notre réseau virtuel.

b) Configuration :

- La configuration consiste à mettre en place un ensemble de machines piégées.

On modifie ainsi le contenu du fichier honeyd.conf:

```
create default
set default default udp action block
set default default icmp action block

create router
set router personality "Cisco 1601R router running IOS 12.1(5)"
set router default tcp action reset
add router tcp port 22 "/usr/share/honeyd/scripts/test.sh"
add router tcp port 23 "/usr/share/honeyd/scripts/router-
telnet.pl"
```

```
bind 192.168.1.1 router

create windows
set windows personality "Microsoft Windows XP Professional SP1"
set windows default tcp action reset
set windows default udp action reset
add windows tcp port 135 open
add windows tcp port 139 open
add windows tcp port 445 open
add windows tcp port 22 open
add windows tcp port 23 "perl
/usr/share/honeyd/scripts/telnet/faketelnet.pl"
add windows tcp port 21 "sh /usr/share/honeyd/scripts/ftp.sh"
add linux tcp port 80 “sh /usr/share/honeyd/web.sh”
set windows ethernet "00:00:24:ab:8c:12"
bind 192.168.1.2 windows


#linux 2.4.x computer
create linux
set linux personality "Linux Kernel 2.4.20"
set linux default tcp action reset
set linux default udp action reset
add linux tcp port 110 "sh /usr/share/honeyd/scripts/pop3.sh"
add linux tcp port 25 "perl /usr/share/honeyd/scripts/smtp.pl"
add linux tcp port 21 "sh /usr/share/honeyd//scripts/ftp.sh"
add linux tcp port 80 “sh /usr/share/honeyd/web.sh”
set linux uptime 5223212
set linux ethernet "00:20:ED:78:C5:A3"
bind 192.168.1.3 linux
```

- Explication du fichier de configuration :

-Nous avons crée deux machines virtuelles : -Windows

-Linux

-Chaque système est créé avec la commande create.

-Ensuite, le système est en outre précisé et configuré avec les commandes add et set. Avec la commande set personality une personnalité est affecté à un système créé.(exemple : set linux personality "Linux 2.2.12 - 2.2.19").

-Une adresse Mac a été attribuée à chacune des deux machines "set linux Ethernet « 00 :20 :Ed :78 :C5 :A3 » ",en plus d’une adresse IP statique à chaque poste virtuel

"bind 192.168.1.2 Windows "

- Afin de faire une correspondance entre adresse Mac et adresse IP des machines utilisées on utilise la commande farpd :
farpd 192.168.1.2-192.168.1.3

-Il est aussi possible de choisir l'action par défaut prise en charge par les protocoles réseaux comme block, reset ou open. Si la valeur par défaut est mise à ouvert, tout les ports pour le

protocole souhaité sont dans un état d'écoute. La valeur reset définit tous les ports d'être fermé tandis que block est utilisé pour passer tous les paquets pour le protocole désigné.

Ajouter des services, des scripts désignés à un certain port se fait en utilisant la commande add. Au lieu de lier un script à un port, il est également possible de transmettre le trafic vers

un autre IP en utilisant proxy.

=> En résultat, quand Honeyd reçoit un paquet dirigé à l'une de ces adresses, il utilisera le profil associé et répondra conformément à sa configuration.
Chaque profil possède les paramètres de configuration permettant de déterminer comment le système se comportera. Sur le listing ci-dessus, vous pouvez observer que les lignes 11,27 déterminent le type de système d'exploitation des ordinateurs virtuels par la définition du comportement de leur pile TCP/IP (conformément à la base de caractéristiques du programme Nmap).
En effet, si l'intrus scanne le réseau 192.168.1.0 à l'aide de ce programme, il trouvera en plus des machines du réseau, 2 autres machines : le système honeypot, Linux Kernel 2.4.20,Microsoft Windows XP Professional SP1.

-Les scripts d'émulation de services :

Pour émuler un service fonctionnant sur une machine virtuelle, Honeyd permet l'utilisation de scripts. Ceux-ci peuvent être écrit en langage Perl ou même directement en SHELL. Des exemples de scripts sont fournis avec l'installation d'Honeyd. Les différents scripts sont situés dans le répertoire :

/usr/share/honeyd

Pour obtenir d'autres scripts , on peut visiter la rubrique « contributions »sur le site Web de honeyd : http://www.honeyd.org/contrib.php

4.Un scénario de test (la partie la plus importante)

Pour tester notre configuration ,on utilise la commande suivante :

```
root@ubuntu:~# honeyd -d -f /etc/honeypot/honeyd1.conf -l /var/log/honeypot/hone
yd.log -p /etc/honeypot/nmap.prints
```

Les options :
-d :permet de lancer honeyd en mode interactif
-f :permet de préciser le fichier de configuration
-p :permet de préciser le fichier contenant les empreintes des OS
-l :permet de préciser le fichier log

```
root@ubuntu:-# honeyd -d -f /etc/honeypot/honeyd1.conf -l /var/log/honeypot/honc
yd.log -p /etc/honeypot/nmap.prints
Honeyd V1.5c Copyright (c) 2002-2007 Niels Provos
honeyd[5365]: started with -d -f /etc/honeypot/honeyd1.conf -l /var/log/honeypot
/honeyd.log -p /etc/honeypot/nmap.prints
honeyd[5365]: listening promiscuously on eth0: (arp or ip proto 47 or (udp and s
rc port 67 and dst port 68) or (ip )) and not ether src 00:0c:29:3e:0d:c4
honeyd[5365]: Demoting process privileges to uid 65534, gid 65534
```

Pour vérifier la bonne configuration de nos deux machines ,à partir d'une autre machine on lance les commandes suivantes :

```
root@ubuntu:-# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_req=1 ttl=128 time=25.5 ms
64 bytes from 192.168.1.2: icmp_req=2 ttl=128 time=0.607 ms
64 bytes from 192.168.1.2: icmp_req=3 ttl=128 time=0.672 ms
64 bytes from 192.168.1.2: icmp_req=4 ttl=128 time=0.839 ms
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.607/6.921/25.566/10.765 ms
root@ubuntu:~# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_req=1 ttl=255 time=1.49 ms
64 bytes from 192.168.1.3: icmp_req=2 ttl=255 time=0.544 ms
64 bytes from 192.168.1.3: icmp_req=3 ttl=255 time=0.650 ms
64 bytes from 192.168.1.3: icmp_req=4 ttl=255 time=0.689 ms
^C
--- 192.168.1.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.544/0.845/1.499/0.382 ms
```

Nous constatons que nos machines honeyd sont bien en marche et répondent à notre ping :

```
honeyd[3958]: arp reply 192.168.1.2 is-at 00:00:24:3c:d0:18
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.2 -> 192.168.1.5
honeyd[3958]: arp_send: who-has 192.168.1.5 tell 192.168.1.2
honeyd[3958]: arp_recv_cb: 192.168.1.5 at 00:0c:29:6b:1d:39
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.2 -> 192.168.1.5
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.2 -> 192.168.1.5
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.2 -> 192.168.1.5
honeyd[3958]: arp reply 192.168.1.3 is-at 00:20:ed:dd:db:1f
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.3 -> 192.168.1.5
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.3 -> 192.168.1.5
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.3 -> 192.168.1.5
honeyd[3958]: Sending ICMP Echo Reply: 192.168.1.3 -> 192.168.1.5
```

En plus ,tous ces messages ont été enregistré dans le fichier log :

```
root@ubuntu:/var/log/honeypot# cat /var/log/honeypot/honeyd.log
2013-11-24-11:32:05.5832 honeyd log started ------
2013-11-24-11:32:09.5932 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:10.5895 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:11.5922 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:12.5940 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:13.5964 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:14.5994 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:15.6080 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:16.6029 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:17.6063 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:18.6091 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:19.6114 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:20.6120 icmp(1) - 192.168.1.30 192.168.1.2: 8(0): 84
2013-11-24-11:32:20.8589 udp(17) - 192.168.153.1 51732 239.255.255.250 1900: 161
2013-11-24-11:32:23.8590 udp(17) - 192.168.153.1 51732 239.255.255.250 1900: 161
2013-11-24-11:32:26.8605 udp(17) - 192.168.153.1 51732 239.255.255.250 1900: 161
2013-11-24-11:32:29.9057 udp(17) - 192.168.153.1 51732 239.255.255.250 1900: 161
2013-11-24-11:32:32.9066 udp(17) - 192.168.153.1 51732 239.255.255.250 1900: 161
2013-11-24-11:32:35.9074 udp(17) - 192.168.153.1 51732 239.255.255.250 1900: 161
```

On peut vérifier que les ports de chaque machines honeyd ouverts sont celle mentionnés dans le fichier de configuration :

nmap 192.168.1.3 :

```
root@ubuntu:~# nmap 192.168.1.3

Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-25 12:20 PST
Nmap scan report for 192.168.1.3
Host is up (0.0082s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
25/tcp  open  smtp
110/tcp open  pop3
MAC Address: 00:20:ED:80:98:46 (Giga-byte Technology CO.)

Nmap done: 1 IP address (1 host up) scanned in 13.43 seconds
```

nmap 192.168.1.2 :

```
root@ubuntu:~# nmap 192.168.1.2

Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-25 12:21 PST
Nmap scan report for 192.168.1.2
Host is up (0.0018s latency).
Not shown: 994 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
23/tcp  open  telnet
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:09:24:09:1A:C9 (Connect AS)

Nmap done: 1 IP address (1 host up) scanned in 14.47 seconds
```

On test quelque ports ouverts :

Telnet :

```
root@ubuntu:~# telnet 192.168.1.2
Trying 192.168.1.2...
Connected to 192.168.1.2.
```

On constate que Honeyd affiche sur la console ces messages :

```
honeyd[5103]: Connection request: tcp (192.168.1.30:58382 - 192.168.1.2:23)
honeyd[5103]: Killing attempted connection: tcp (192.168.1.2:23 - 192.168.1.30:5
8382)
honeyd[5103]: Connection established: tcp (192.168.1.30:58382 - 192.168.1.2:23)
<-> perl /usr/share/honeyd/scripts/telnet/faketelnet.pl
honeyd[5103]: Killing attempted connection: tcp (192.168.1.2:23 - 192.168.1.30:5
8382)
honeyd[5103]: Connection dropped by reset: tcp (192.168.1.30:58382 - 192.168.1.2
:23)
honeyd[5103]: arp reply 192.168.1.2 is-at 00:00:24:e3:50:c1
honeyd[5103]: arp_recv_cb: 192.168.1.30 at 00:0c:29:6b:1d:39
```

Ftp :

```
root@ubuntu:~# ftp 192.168.1.2
Connected to 192.168.1.2.
421 Service not available, remote server has closed connection
ftp>
```

```
honeyd[5276]: arp reply 192.168.1.2 is-at 00:00:24:23:29:c0
honeyd[5276]: Connection request: tcp (192.168.1.30:38994 - 192.168.1.2:21)
honeyd[5276]: arp_send: who-has 192.168.1.30 tell 192.168.1.2
honeyd[5276]: arp_recv_cb: 192.168.1.30 at 00:0c:29:6b:1d:39
honeyd[5276]: arp_recv_cb: 192.168.1.30 at 00:0c:29:6b:1d:39
honeyd[5276]: Connection established: tcp (192.168.1.30:38994 - 192.168.1.2:21)
<-> sh /usr/share/honeyd/scripts/ftp.sh
honeyd[5276]: arp_recv_cb: 192.168.1.30 at 00:0c:29:6b:1d:39
honeyd[5276]: Connection dropped by reset: tcp (192.168.1.30:38994 - 192.168.1.2
:21)
honeyd[5276]: arp reply 192.168.1.2 is-at 00:00:24:23:29:c0
```

5.Conclusion

Pour synthétiser, Les honeypots ne sont pas une solution que l'on place pour résoudre un problème mais un outil à exploiter.
De plus, Le coût en terme ressources matérielles est négligeable. En outre, Le faux système peut être mis en place à côté des serveurs de production. Le honeypot doit être attractif afin de susciter l'intérêt, sinon, il sera ignoré et donc inutile.
Les honeypots ont donc un intérêt certain. Mais, avant de se lancer dans cette aventure, il faut bien verrouiller son réseau. Les pots de miel sont loin d'être la première étape dans une architecture de réseau sécurisée.