TP Pentest / IDS / Protection applicative

Fabrice Prigent

13 mars 2009

1 Préambule

Le but de ce TP est de voir ce qui se passe quand les pare-feux ont été installés, les ports nécessaires ont été ouverts.

Nous partirons d'une cible "crédible" pour le TP en créant un site web volontairement vulnérable. Ceci sera fait grâce à un logiciel honeypot.

Nous travaillerons ensuite sur un pentest (penetration test) rapide pour voir ce que nous obtenons en vérifiant le site en question, et nous constaterons les dégâts.

Nous analyserons ce que nous donnent les journaux à propos de ce pentest, et nous travaillerons ensuite à la mise en place de détections plus élaborée grâce à un IDS (détecteur d'intrusion).

Enfin, Nous travaillerons à la mise en place d'une protection applicative sur ce dispositif

2 Installation d'une cible

Afin de créer une cible apte à déclencher des alertes, et nous éviter la mise en place de logiciels trop lourd, nous allons utiliser des dispositifs de honeypot "applicatif". Deux logiciels libres permettent de faire cela de manière simple

- GHH : http ://ghh.sourceforge.net qui n'a pas évolué depuis le 23 février 2007
- PHPHOP : qui n'est plus maintenu depuis 2006, et n'est plus disponible. Nous utiliserons sa version 0.5

2.1 Installation du serveur WWW

```
apt-get install apache2
apt-get php5
apt-get php5-mysql

/etc/init.d/apache2 start
```

2.2 Installation du honeypot PHPHOP

```
cd /var/www
#        Si le répertoire html n'existe pas
ln -s . html
```

```
cd html
wget http://192.168.30.175/securite/phphop.tgz
tar zxvf phphop.tgz
rm phphop.tgz

#
#       Création du répertoire de logs
#
mkdir logs
chown www-data:www-data logs

#
#       On place les applications simulées à l'endroit habituel
#
ln -s phphop/applis/phpmyadmin/ phpmyadmin
ln -s phphop/applis/horde/ horde
```

L'ensemble des scripts du honeypot se trouve désormais accessible par http ://127.0.0.1/phphop et http ://127.0.0.1/phphop/applis. Une rapide exploration du répertoire donne l'ensemble des applications simulées.

2.3 Installation d'un serveur MySQL

```
apt-get install mysql-server
/etc/init.d/mysql start
mysql
CREATE DATABASE secu;
USE secu;
CREATE TABLE simple (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY, text VARCHAR(20));
INSERT INTO simple (text) VALUES ("titi");
INSERT INTO simple (text) VALUES ("tata");
INSERT INTO simple (text) VALUES ("tutu");
INSERT INTO simple (text) VALUES ("toto");
INSERT INTO simple (text) VALUES ("tete");
```

On crée alors un script test_simple.php

```
<?php
echo '<HTML>
<HEAD>
<TITLE>Test un accès à la base de données</TITLE>
</HEAD>
<BODY>
';
$ID=$_GET['ID'];
$serveur="localhost";
$compte_bd="root";
$passwd_bd="";
```

```
echo "<UL>";
$link = mysql_connect($serveur,$compte_bd,$passwd_bd);
if (!$link) {
    die('Le serveur de base de données sur $serveur est injoignable: ' . mysql_error());
}
echo '<LI>Connexion au serveur réussie';

$db_selected = mysql_select_db('secu', $link);
if (!$db_selected) {
    die ('La base secu est inutilisable: ' . mysql_error());
}
echo '<LI>Connexion à la base de donnée réussie';

$result = mysql_query("SELECT * FROM simple WHERE id=$ID");
if (!$result) {
        die('<LI>Requete invalide ' . mysql_error());
        }
else
        {
        echo '<LI>Requête valide';
        while ($row = mysql_fetch_assoc($result)) {
                echo "<LI> Résultat ID:$ID contient ";
                echo $row['text'];
                }
        }

mysql_close($link);
echo '
</UL>
</BODY>
</HTML>
';
?>
```

2.4 Protection minimale avec iptables

Bien évidemment, nous supposerons que le serveur a été protégé par les règles iptables qui ont été définies auparavant. Il faudra donc se référer au script iptables.sh.

3 Outils de pentest

Nous allons rapatrier et utiliser 3 outils différents de PenTest, chacun d'eux amenant ses fonctionnalités.

3.1 nmap

Nous avons déjà utilisé cet outil de test de ports ouverts et de version. Une installation basique, suivie d'un paramétrage un peu spécifique devrait nous

donner des renseignements intéressants. On pourra aussi préférer l'interface graphique zenmaps qui se trouve elle aussi sur le site http ://nmap.org

```
apt-get install nmap
```

3.1.1 nmap : utilisation

```
nmap -T4 -A -v -PE -PA21,23,80,3389         192.168.30.175
```

On trouve alors un nombre important de renseignements :

```
Starting Nmap 4.85BETA3 ( http://nmap.org ) at 2009-03-09 14:39 CET
Failed to resolve given hostname/IP: T4.  Note that you can't use '/mask' AND '1-4,7,100-'
Initiating Parallel DNS resolution of 1 host. at 14:39
Completed Parallel DNS resolution of 1 host. at 14:39, 0.00s elapsed
Initiating SYN Stealth Scan at 14:39
Scanning 192.168.30.175 [1000 ports]
Discovered open port 3306/tcp on 192.168.30.175
Discovered open port 111/tcp on 192.168.30.175
Discovered open port 22/tcp on 192.168.30.175
Discovered open port 80/tcp on 192.168.30.175
Completed SYN Stealth Scan at 14:39, 0.08s elapsed (1000 total ports)
Initiating Service scan at 14:39
Scanning 4 services on 192.168.30.175
Completed Service scan at 14:39, 6.00s elapsed (4 services on 1 host)
Initiating OS detection (try #1) against 192.168.30.175
SCRIPT ENGINE: Initiating script scanning.
Initiating SCRIPT ENGINE at 14:39
Completed SCRIPT ENGINE at 14:39, 0.02s elapsed
Host 192.168.30.175 appears to be up ... good.
Interesting ports on 192.168.30.175:
Not shown: 996 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 5.1 (protocol 2.0)
80/tcp   open  http    Apache httpd 2.2.10 ((Fedora))
|_ html-title: Serveur WWW de Fabrice Prigent
111/tcp  open  rpcbind
|  rpcinfo:
|  100000  2,3,4    111/udp  rpcbind
|  100024  1      40824/udp  status
|  100000  2,3,4    111/tcp  rpcbind
|_ 100024  1      38136/tcp  status
3306/tcp open  mysql   MySQL (unauthorized)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.15 - 2.6.26
Uptime guess: 0.272 days (since Mon Mar  9 08:08:33 2009)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=202 (Good luck!)
IP ID Sequence Generation: All zeros
```

```
Read data files from: /usr/share/nmap
OS and Service detection performed. Please report any incorrect results at http://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 8.11 seconds
           Raw packets sent: 1019 (45.598KB) | Rcvd: 2045 (87.076KB)
```

3.2 nikto

3.2.1 Installation de Nikto

Nikto est un logiciel qui va rechercher des vulnérabilités web courantes. On le trouve sur le site http ://www.cirt.net/nikto2.

```
cd ~
wget http://www.cirt.net/nikto/nikto-current.tar.gz
tar zxvf nikto-current.tar.gz
cd nikto
#
#        On met à jour le
#
./nikto.pl -update
```

3.2.2 Utilisation de Nikto

```
./nikto.pl -h www.univ-perp.fr        #        ATTENTION !! Ceci n'est pas à faire. C'est
                                  #        La syntaxe "normale"
./nikto.pl -h 192.168.30.175        #        Syntaxe ici

#        On utilise la technique d'évasion 1 (il y en a 8)
./nikto.pl -evasion 1 -h 192.168.30.175
```

Nikto va alors nous donner les urls à "observer".

```
- Nikto v2.03/2.04
---------------------------------------------------------------------------
+ Target IP:          192.168.30.125
+ Target Hostname:    192.168.30.125
+ Target Port:        80
+ Using IDS Evasion:  Random URI encoding (non-UTF8)
+ Start Time:         2009-03-10 14:59:50
---------------------------------------------------------------------------
+ Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.5 with Suhosin-Patch
- Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP method ('Allow' Header): 'TRACE' is typically only used for debugging an
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.2.10). Apache 1.3.41 a
+ PHP/5.2.4-2ubuntu5.5 appears to be outdated (current is at least 5.2.6)
+ OSVDB-0: GET /./ : Appending '/./' to a directory allows indexing
+ OSVDB-0: GET /\%2e/ : Weblogic allows source code or directory listing, upgrade to v6.0
+ OSVDB-877: TRACE / : TRACE option appears to allow XSS or credential theft. See http://w
+ OSVDB-119: GET /?PageServices : The remote server may allow directory listings through W
+ OSVDB-119: GET /?wp-cs-dump : The remote server may allow directory listings through Web
```

```
+ OSVDB-3092: GET /html/ : This might be interesting...
+ OSVDB-3092: GET /logs/ : This might be interesting...
+ OSVDB-3092: GET /phpmyadmin/ : phpMyAdmin is for managing MySQL databases, and should be
+ OSVDB-3268: GET /icons/ : Directory indexing is enabled: /icons
+ OSVDB-3233: GET /icons/README : Apache default file found.
+ 3577 items checked: 14 item(s) reported on remote host
+ End Time:        2009-03-10 15:00:26 (36 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested

Test Options: -evasion 1 -h 192.168.30.125
```

3.3 sqlmap

sqlmap est un outil de sql injection. On lui fournit une url avec un paramètre, il va se charger de tester les SQL injection que l'on peut faire dessus.

3.3.1 Installation de sqlmap

```
wget http://downloads.sourceforge.net/sqlmap/sqlmap_0.6.4-1_all.deb
dpkg -i sqlmap_0.6.4-1_all.deb
sqlmap --update
```

3.3.2 Utilisation de sqlmap

```
sqlmap -u "http://127.0.0.1/securite/test_simple.php?ID=1" --tables -v 0
```

4 Détection classique

4.1 Comment

Les journaux représentent le principal moyen de constater une agression. Chaque outil impacté par la communication peut signaler un événement.

4.2 Les pare-feux

Dans notre TP, c'est iptables qui va nous signaler les refus des agressions. Ces refus ne sont pas significatifs individuellement, c'est plutôt leur abondance qui va indiquer quelque chose. On notera l'intérêt de mettre une chaine significative dans le paramètre log-prefix, afin de distinguer les messages noyaux simple des iptables.

```
tail -f /var/log/syslog|grep FIREWALL_DENIED
```

Les journaux ne concerneront bien évidemment que les blocages sur les services interdits. Les adresses IP relevées ne pourront être considérées comme des preuves : que ce soit l'UDP (par essence) ou le TCP (le blocage est sur le SYN), tous deux sont insuffisants.

4.3 Les applications

Les pare-feux ayant fait leur office, ce sont les applications qui vont nous permettre de relever les infractions

```
tail -f /var/log/apache2/error.log
tail -f /var/log/apache2/access.log
```

Le fichier error.log sera particulièrement important, car les tentatives de connexion à des services inexistants ou avec des paramètres aberrants (pour peu que l'application les considère comme tels), seront immédiatement visibles.

L'autre avantage des logs applicatifs est le fait que les adresses IP relevées sont quasiment sûres (à 99%) : la connexion TCP ayant terminé le triple handshake.

Mais on se trouve face à quelques difficultés

- Le volume des journaux
- La distinction entre erreur et agression échouée dans le cas de l'error.log
- La distinction entre accès normale et agression réussie dans le cas de l'access.log

Heureusement, si l'on se doit d'abandonner le principe d'une détection à 100%, on peut tout à fait repérer des agresseurs en analysant les plus "consommateurs". Cette routine va nous donner les 10 machines ayant généré le plus d'erreurs.

```
cat /var/log/apache2/error.log|awk '{print $2}'|sort|uniq -c|sort -n|tail -10
```

5 Détection par IDS

Snort est un logiciel de détection d'intrusion qui se base sur des signatures pour repérer des attaques. Il est libre et largement utilisé dans des produits commerciaux. Il est disponible sur http ://www.snort.org.

5.1 Installation de Snort

5.1.1 Pour Linux debian

```
apt-get snort
```

5.1.2 Pour Windows XP

On se réfèrera au document http ://www.snort.org/docs/setup_guides/Snort Windows XP Guide.txt

5.2 Utilisation de Snort

```
/etc/init.d/snortd start
```

6 La Protection

Deux types de protection sont possibles :

- protection locale le serveur assure sa propre protection
- protection distante : un reverse-proxy assure la protection

6.1 Installation

On se reportera au document ”protection php pour les tout petits” pour les détails concernant l'installation d'une protection relativement complète d'un serveur LAMP. On pourra aussi s'y référer pour trouver les idées et principes applicables à tout serveur web.

6.2 Installation : TP

6.2.1 Installation de mod_security

Ce TP correspond à la protection locale. Nous allons installer le module mod_security sur le serveur qui doit être protégé. Celui-ci n'étant pas disponible directement, il faut le compiler.

```
echo 'deb http://etc.inittab.org/~agi/debian/libapache-mod-security2/etch/ .' \
        >>/etc/apt/sources.list
gpg --keyserver pgpkeys.mit.edu --recv-keys C514AF8E4BA401C3
gpg --export -a C514AF8E4BA401C3 |apt-key add -
apt-get update
apt-get install libapache2-mod-security2
```

6.2.2 Installation d'un reverse-proxy

Ce TP correspond à la protection remote. Nous allons profiter du TP précédent et installer le module mod_proxy et mod_proxy_html sur le serveur qui va faire la protection. On effectuera le relai sur un serveur extérieur (par exemple http ://cri.univ-tlse1.fr), afin de le protéger.

```
apt-get install libapache2-mod-proxy
apt-get install libapache2-mod-proxy-html
sudo a2enmod proxy
```

6.2.3 Configuration du reverse-proxy

On ajoute à la configuration par défaut du apache les éléments suivants

```
ProxyRequests Off               # Pour empêcher d'être proxy "normal"
<Proxy *>
        Order        deny,allow
        Allow from all
</Proxy>
ProxyPass                /        http://cri.univ-tlse1.fr/
ProxyPassReverse        /        http://cri.univ-tlse1.fr/

# Le module de réécriture est lancé
SetOutputFilter         proxy-html

# On utilise le mode étendu pour toucher les scripts
ProxyHTMLExtended      on
ProxyHTMLURLMap         http://cri.univ-tlse1.fr/([^\)]*) http://127.0.0.1$1 Ri
```