TP Pentest / IDS / Protection applicative

Fabrice Prigent

March 9, 2021

1 Préambule

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

Nous partirons d'une cible "crédible" pour le TP en utilisant un site web volontairement vulnérable. Ceci sera fait grâce à une distribution spécialisée.

Nous travaillerons ensuite sur un pentest (penetration test : test d'intrusion) 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 élaborées grâce à un IDS (détecteur d'intrusion).

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

2 Installation de l'infrastructure

2.1 La cible

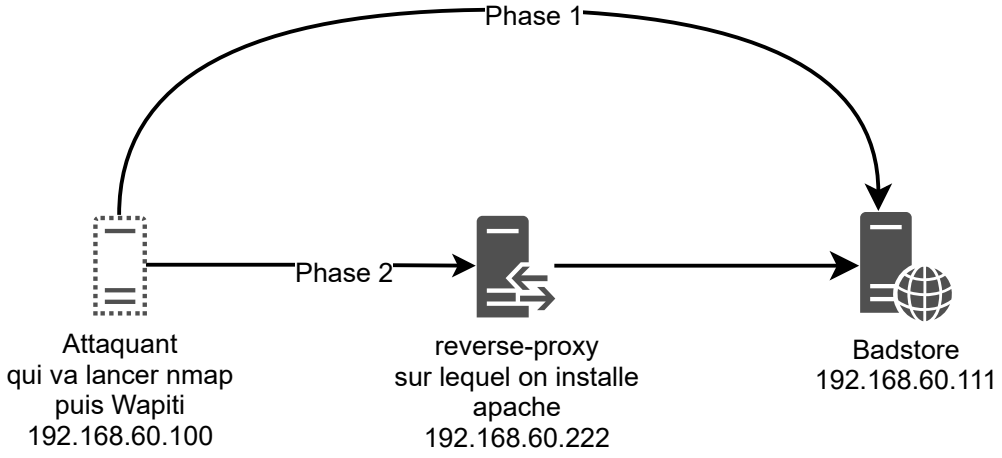
Afin de créer une cible apte à déclencher des alertes, et nous éviter la mise en place de logiciels trop lourds, nous utiliserons en fait la distribution BadStore qui a été conçue spécifiquement pour cela :

- https://dsi.ut-capitole.fr/cours/BadStore_123.iso

Après avoir récupéré l'ISO de 10 Mo, on définit une machine virtuelle (virtualbox) avec les caractéristiques suivantes

- 256 Mo de RAM
- Disque dur de 128 Mo
- OS : Linux, version Linux kernel 2.4 (ou "Other Linux") / 32bits
- Configuration réseau : en mode "bridge / pont". Pour obtenir son IP, il faut taper "ifconfig". Pour VirtualBox, prendre la PCnet-PCI II.

Il ne reste plus qu'à démarrer la machine à partir de l'ISO. ATTENTION : le clavier sera en anglais. Pour la suite de l'exercice, nous supposerons que l'adresse IP de la cible est 192.168.60.111.



2.2 La protection

La protection sera une machine linux. Elle devra être démarrée dans le mode "bridge" ou "pont". Pour la suite de l'exercice nous supposerons que la machine de protection aura pour adresse IP 192.168.60.222.

2.3 L'attaquant

C'est la machine support (Windows ou Linux) sur laquelle nous allons installer chacun des outils de pentest.

3 Outils de pentest

Nous allons rapatrier et utiliser 2 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 zenmap qui se trouve elle aussi sur le site https://nmap.org.

Une version pour Windows existe et peut être rapatriée et utilisée directement

```
apt-get install nmap
```

3.1.1 nmap : utilisation

```
nmap -T4 -A -v -PE -PA21,23,80,3389 192.168.60.111
```

On trouve alors un nombre important de renseignements :

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-03-04 15:57 CET
NSE: Loaded 153 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:57
Completed NSE at 15:57, 0.00s elapsed
Initiating NSE at 15:57
Completed NSE at 15:57, 0.00s elapsed
Initiating NSE at 15:57
Completed NSE at 15:57, 0.00s elapsed
Initiating Ping Scan at 15:57
Scanning badstore1 (193.49.52.14) [5 ports]
Completed Ping Scan at 15:57, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:57
Completed Parallel DNS resolution of 1 host. at 15:57, 0.01s elapsed
Initiating SYN Stealth Scan at 15:57
Scanning badstore1 (193.49.52.14) [1000 ports]
Discovered open port 443/tcp on 193.49.52.14
Discovered open port 3306/tcp on 193.49.52.14
Discovered open port 80/tcp on 193.49.52.14
Completed SYN Stealth Scan at 15:57, 0.03s elapsed (1000 total ports)
Initiating Service scan at 15:57
Scanning 3 services on badstore1 (193.49.52.14)
Completed Service scan at 15:58, 12.03s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against badstore1 (193.49.52.14)
Initiating Traceroute at 15:58
Completed Traceroute at 15:58, 0.01s elapsed
Initiating Parallel DNS resolution of 1 host. at 15:58
Completed Parallel DNS resolution of 1 host. at 15:58, 0.00s elapsed
NSE: Script scanning 193.49.52.14.
Initiating NSE at 15:58
Completed NSE at 15:58, 0.49s elapsed
Initiating NSE at 15:58
Completed NSE at 15:58, 0.04s elapsed
Initiating NSE at 15:58
Completed NSE at 15:58, 0.00s elapsed
```

```
Nmap scan report for badstore1 (193.49.52.14)
Host is up (0.00047s latency).
rDNS record for 193.49.52.14: badstore1.ut-capitole.fr
Not shown: 997 closed ports
PORT STATE SERVICE VERSION
80/tcp open http Apache httpd 1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c)
|_http-favicon: Unknown favicon MD5: A9CBB6E162F76BE464E6BC308B3266B9
| http-methods:
| Supported Methods: GET HEAD OPTIONS TRACE
|_ Potentially risky methods: TRACE
| http-robots.txt: 5 disallowed entries
|_/cgi-bin /scanbot /backup /supplier /upload
|_http-server-header: Apache/1.3.28 (Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c
|_http-title: Welcome to BadStore.net v1.2.3s
443/tcp open ssl/http Apache httpd 1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c)
|_http-favicon: Unknown favicon MD5: A9CBB6E162F76BE464E6BC308B3266B9
| http-methods:
| Supported Methods: GET HEAD OPTIONS TRACE
|_ Potentially risky methods: TRACE
| http-robots.txt: 5 disallowed entries
|_/cgi-bin /scanbot /backup /supplier /upload
|_http-server-header: Apache/1.3.28 (Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c
|_http-title: Welcome to BadStore.net v1.2.3s
| ssl-cert: Subject: commonName=www.badstore.net/organizationName=BadStore.net/stateOrProvinceName=
    Illinois/countryName=US
| Subject Alternative Name: email:root@badstore.net
| Issuer: commonName=Snake Oil CA/organizationName=Snake Oil, Ltd/stateOrProvinceName=Snake Desert/
    countryName=XY
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: md5WithRSAEncryption
| Not valid before: 2006-05-10T12:52:53
| Not valid after: 2009-02-02T12:52:53
| MD5: 4d68 3443 fab8 8f51 2057 19e9 ed18 78c5
|_SHA-1: c0bf 6ef8 98c5 b661 f703 0bb4 f4bc 5bbd e6a0 fbc1
|_ssl-date: 2021-03-04T14:58:08+00:00; 0s from scanner time.
| sslv2:
| SSLv2 supported
| ciphers:
| SSL2_IDEA_128_CBC_WITH_MD5
| SSL2_DES_192_EDE3_CBC_WITH_MD5
| SSL2_RC4_64_WITH_MD5
| SSL2_RC2_128_CBC_WITH_MD5
| SSL2_RC4_128_EXPORT40_WITH_MD5
| SSL2_DES_64_CBC_WITH_MD5
| SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_ SSL2_RC4_128_WITH_MD5
3306/tcp open mysql MySQL 4.1.7-standard
| mysql-info:
| Protocol: 10
| Version: 4.1.7-standard
| Thread ID: 15
| Capabilities flags: 33324
| Some Capabilities: Support41Auth, Speaks41ProtocolNew, LongColumnFlag, ConnectWithDatabase,
    SupportsCompression
| Status: Autocommit
|_ Salt: pf/UA%3p[X*b4w!711B7
Device type: print server|specialized
Running: HP embedded, Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4.21
OS details: HP 4200 PSA (Print Server Appliance) model J4117A, Linux 2.4.21 (embedded)
Uptime guess: 0.053 days (since Thu Mar 4 14:41:13 2021)
Network Distance: 2 hops
```

```
TCP Sequence Prediction: Difficulty=198 (Good luck!)
IP ID Sequence Generation: All zeros

TRACEROUTE (using port 995/tcp)
HOP RTT ADDRESS
1 0.20 ms ars-gw55.ut-capitole.fr (10.55.255.254)
2 0.44 ms badstore1.ut-capitole.fr (193.49.52.14)

NSE: Script Post-scanning.
Initiating NSE at 15:58
Completed NSE at 15:58, 0.00s elapsed
Initiating NSE at 15:58
Completed NSE at 15:58, 0.00s elapsed
Initiating NSE at 15:58
Completed NSE at 15:58, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.09 seconds
          Raw packets sent: 1034 (46.226KB) | Rcvd: 1022 (41.602KB)
```

3.2 Wapiti

Wapiti est un scanner de vulnérabilités disponible pour debian

```
apt update
apt install wapiti
```

On peut ensuite le lancer sur la cible, en faisant attention d'effacer les précédents résultats.

```
wapiti -u http://192.168.60.111/
```

3.3 Conclusions

Si les étudiants sont motivés et intéressés, on pourra leur conseiller de faire les attaques "à la main" sur l'outil. Ils trouveront certainement dans le manuel BadStore (disponible sur le site web BadStore) des pistes intéressantes.

4 La Protection

Deux types de protection sont possibles :

- protection locale : le serveur assure sa propre protection
- protection distante : un reverse-proxy assure la protection (cas général du point précédent car le reverse-proxy sera protégé avec la même méhode que la protection locale).

4.1 Protection locale

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.

On pourra aussi se reporter à l'installation du module mod_security.

4.2 Installation d'un reverse-proxy

Le reverse-proxy pourra être installé en vérifiant les préceptes indiqués plus haut. On notera cependant qu'il faudra l'autoriser à se connecter en HTTP à la machine qu'il protège.

Nous allons installer le module mod_proxy_html ainsi que mod_substitute sur le serveur qui va faire la protection. On effectuera le relais sur un serveur extérieur (dans notre cas, le serveur BadStore http://192.168.60.111).

```
a2enmod proxy
a2enmod proxy_http
a2enmod substitute
a2enmod rewrite
a2enmod ssl
```

Ne pas s'inquiéter si l'un des modules est annoncé comme

```
module proxy already enabled
```

4.3 Configuration du reverse-proxy

On ajoute à la configuration par défaut du apache, dans un fichier /etc/apache2/conf-available/reverse-proxy.conf les éléments suivants

```
ProxyRequests Off  # Pour empecher d'etre proxy "normal"
<Proxy *>
   Order deny,allow
   Allow from all
</Proxy>
ProxyPass  / http://192.168.60.111/
ProxyPassReverse / http://192.168.60.111/

Substitute "s#https?://192.168.60.111([A-Za-z0-9/\-\.]*)#http://192.168.60.222$1#iq"
```

Il ne reste plus qu'à regarder ce que cela donne en activant la configuration, en la vérifiant puis en relançant le daemon httpd

```
a2enconf reverse-proxy
apachectl configtest
service apache2 restart
```

4.3.1 Test du reverse proxy

On peut alors vérifier que le reverse proxy marche en se connectant à http://192.168.60.222. Cela devrait nous donner la même page que lorsque l'on se connectait sur http://192.168.60.111.

4.4 Mod_Security

4.4.1 Installation de mod_security

Nous allons installer le module mod_security sur le serveur qui doit être protégé (ici en fait le reverse-proxy).

```
apt-get update
apt-get install libapache2-mod-security2
```

4.4.2 Configuration de mod_security

Pour configurer le mod_security

```
cat /etc/modsecurity/modsecurity.conf-recommended|sed -e '
s/DetectionOnly/On/;
' > /etc/modsecurity/modsecurity.conf
service apache2 restart
```

4.4.3 Vérification

Après une installation réussie, il ne reste plus qu'à vérifier si les attaques précédentes réussissent toujours.

4.4.4 Alternative

On pourra s'intéresser aussi au travail de Jeff Starr avec une protection qui se contente d'être mise en tant que configuration apache, mais aussi en configuration Nginx.

5 Détection classique

5.1 Comment

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

5.2 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 FW_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.

5.3 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 normal 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
```

Mod_security est souvent trop exigeant. Il faudra regarder du côté des désactivations de règles pour éviter des blocages injustifiés sur certaines machines (dans l'exemple : globalement, par url, par argument).

```
SecRuleRemoveById 960010
SecRule REQUEST_URI "^/url/a_ne_pas_analyser" "phase:1,id:10001,nolog,ctl:ruleRemoveById=960010"
SecRule ARGS "logoutRequest" "phase:1,id:10002,nolog,ctl:ruleRemoveById=973332"
```

6 Détection par IDS

Suricata est un logiciel de détection d'intrusion qui se base sur des signatures pour repérer des attaques. Il est libre et disponible sur http://www.suricata-ids.org.

6.1 Installation de Suricata

```
apt install suricata suricata-update
```

6.2 Vérification du paramétrage de Suricata

On vérifiera le paramétrage avec la commande suivante:

```
suricata -T
```

6.3 Paramétrage de Suricata

On change les interfaces avec celles fournies par un "ip a" de la machine attaquée (le reverse-proxy). Il faut donc chercher dans le fichier /etc/suricata/suricata.yaml, les zones qui définissent certaines variables et les changer, ainsi que les zones IP, le chemin des règles.

```
    HOME_NET: "[192.168.60.222]"
    EXTERNAL_NET: "any"
```

ensuite

```
pfring:
  - interface: enp0s3
```

puis

```
af-packet:
  - interface: enp0s3
```

et enfin (en prenant soin de détruire les autres default-rule-path et rule-files

```
default-rule-path: /var/lib/suricata/rules
rule-files:
  - suricata.rules
```

6.4 Utilisation de Suricata

```
suricata-update

service suricata restart
```

Pas d'inquiétude sur les erreurs, tant que l'on obtient que le suricata est actif. On vérifie ensuite les attaques

```
tail -f /var/log/suricata/fast.log
```

6.4.1 Pour aller plus loin

Il est intéressant d'installer https://github.com/OISF/suricata-update