Majeure info
TP - Scurit Lab

Nicolas Lagaillardie

5 novembre 2018

Table des matières

1 Objectif

Le but du TP est de tester la scurit d'un serveur et d'exploiter ses diffrentes failles de scurit. Nous utilisons pour cela une machine virutlle qui hberge OpenSuse 11.2, et nous attaquerons le serveur depuis cette machine.

2 Analyse du traffic internet

Voici les diffrentes commandes utilises pour tablir des connexions entre notre PC et la machine virtuelle

```
$ ifconfig

$ netstat -an -t

$ telnet [adresse ip]

$ ftp [adresse ip]

$ ssh [user]@[adresse ip]
```

Dans l'ordre, ces commandes permmettent de rcuprer l'adresse ip de la machine courante, de rcuprer le port TCP, et d'tablir des une connections *telnet / FTP / SSH* avec l'autre machine.

Telnet permet d'envoyer et recevoir des lignes de texte avec la machine distante.
FTP permet d'envoyer et recevoir des fichiers.
SSH permet d'ouvrir un terminal sur la machine distante, et donc d'en prendre le contrle.

3 Scurit de l'application

Aprs avoir dmarr le serveur depuis notre machine, nous accdons l'adresse **localhost:8080** sur la machine distante. Cela signifie que le port 8080 est maintenant occup et indisponible pour tout autre type de connection. Voic une capture d'cran de ce que nous obtenons

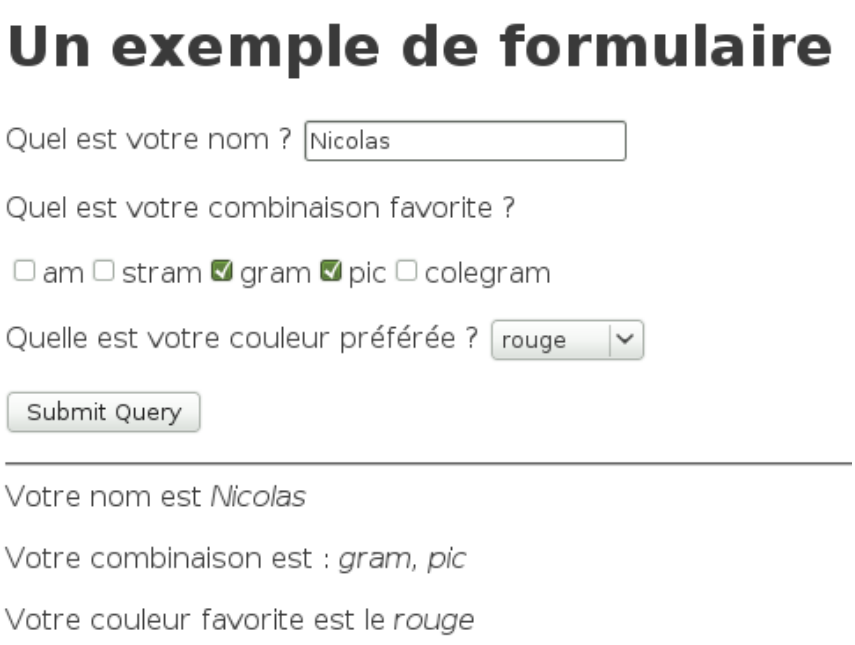

FIG. 1 – *Screenshot de l'application*

4 Deny of service et buffer overflow

Une premire attaque consiste envoyer un nom bien trop lourd pour pouvoir tre trait correctement par le serveur. Cela rsulte tout simplement en un crash du serveur, que l'on doit alors redmarrer.
Une solution pour prvenir cette attaque est de limiter la taille de l'input.
Une autre attaque consiste simuler un trs grand nombre de connections simultanes au serveur, un DDOS. Je n'ai pas pu le faire sur ma propre machine puisque c'est elle-mme qui s'attaque (en terme d'hardware). Mais avec au moins deux machines distinctes, c'est possible. Dans ce cas, le serveur prend normment de temps pour rpondre une requte voir refuse toute requte et crash. Il n'existe pas de vritable solution ce problme, et une alternative consiste refuser des requtes provenant de ces requtes intempestives.

5 Excution de commande distance

EN m'inspirant du code index.pl, voici un code un script bash qui demande quelques informations sur l'utilisateur et qui renvoie un **Bonjour** [**utilisateur**] ainsi que d'autres informations sur la machine de l'utilisateur.

```
#!/bin/bash
echo "Content-type: text/html"
echo ""
echo "<html><head><title>Bash as CGI"
echo "</title></head><body>"

echo "<h1>General system information for host $(hostname -s)</h1>"
echo ""

echo "<h1>Memory Info</h1>"
echo "<pre> $(free -m) </pre>"

echo "<h1>Disk Info:</h1>"
echo "<pre> $(df -h) </pre>"

echo "<h1>Logged in user</h1>"
echo "<pre> $(w) </pre>"

echo "<center>Information generated on $(date)</center>"
echo "</body></html>"

echo Bonjour $FORM_civilite $FORM_prenom $FORM_nom <!-- affichage de la civilit, du
    ↪ prenom et du nom de la personne -->

cat << EOF
</h2>
</center>
<hr>
</body>
</html>
EOF
```

Pour corriger ce “dfaut”, il faut dsactiver les scripts CGI sur le serveur, ce qui peut se faire l’aide de cette commande.

```
$ a2dismod cgi
```

6 Cross site scripting

Pour exploiter des failles concernant les messages d'erreur renvoys par le serveur, nous avons inject du code Javascript directement dans l'input. Le code suivant permet par exemple de connatre directement la valeur des cookies de la session actuelle.

```
<script>
       alert(document.cookie)
</script>
```