

Major computer science

Lab work Hue light bulbs

Nicolas LAGAILLARDIE
Paul BREUGNOT

8 novembre 2018

Table des matières

1	Objectives	2
2	Analyse du trafic internet	2
3	Sécurité de l'application	2
4	Deny of service et buffer overflow	3
5	Exécution de commande à distance	3
6	Cross site scripting	5

1 Objectives

Thanks to this labwork, we will be able to control the light bulbs ©

2 Analyse du trafic internet

Voici les différentes commandes utilisées pour établir 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 permettent de récupérer l'adresse ip de la machine courante, de récupérer le port TCP, et d'établir des des connexions *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 contrôle.

3 Sécurité de l'application

Après avoir démarré le serveur depuis notre machine, nous accédons à l'adresse **localhost:8080** sur la machine distante. Cela signifie que le port 8080 est maintenant occupé et indisponible pour tout autre type de connection. Voic une capture d'écran de ce que nous obtenons :

Un exemple de formulaire

Quel est votre nom ?

Quel est votre combinaison favorite ?

☐ am ☐ stram ☒ gram ☒ pic ☐ colegram

Quelle est votre couleur préférée ?

Votre nom est *Nicolas*

Votre combinaison est : *gram, pic*

Votre couleur favorite est le *rouge*

FIG. 1 – Screenshot de l'application

4 Deny of service et buffer overflow

Une première attaque consiste à envoyer un nom bien trop lourd pour pouvoir être traité correctement par le serveur. Cela résulte tout simplement en un crash du serveur, que l'on doit alors redémarrer.

Une solution pour prévenir cette attaque est de limiter la taille maximale de l'input et refuser l'input si la taille est trop importante.

Une autre attaque consiste à simuler un très grand nombre de connections simultanées au serveur, un DDOS. Nous n'avons pas pu le faire sur nos propres machines puisque c'est elle-même qui s'attaque (en terme d'hardware). Mais avec au moins deux machines distinctes, c'est possible. Dans ce cas, le serveur prend énormément de temps pour répondre à une requête voir refuse toute requête et crash. Il n'existe pas de véritable solution à ce problème, et une alternative consiste à refuser des requêtes provenant de ces requêtes intempestives.

5 Exécution de commande à 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 civilite\{e}, du
    ↪ prenom et du nom de la personne -->

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

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

```
$ a2dismod cgi
```

6 Cross site scripting

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

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