

Remote buster/mapper

- José Catalán Fuentes 201551010 - 5

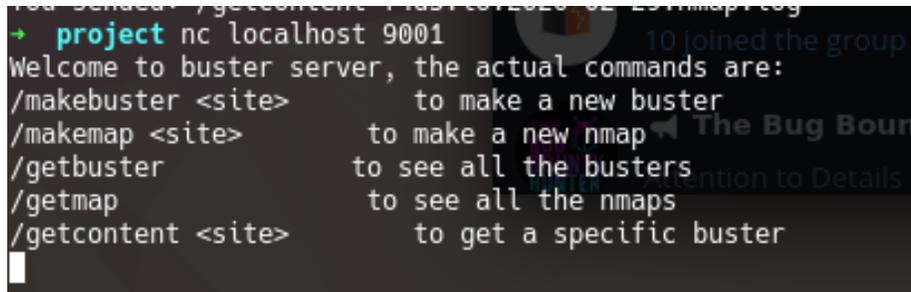
Introducción

El proyecto se trata de hacer un centro de comandos que permita, administrar escaneos de puertos y mapeos de sitios remotamente, para esto se utilizó lo aprendido en el curso en cuanto a `java` y `websockets`, lo cual permitió llegar a una solución la cual permite, correr y leer outputs de `nmap` y `dirsearch` remotamente.

Bosquejo de la solución

De manera por defecto, el programa correrá en el puerto 9001, esto para no necesitar permisos de `root` para lograr correr el programa. De la misma manera, como aún no se ha decidido bajo qué directorio cerrará el programa, tiene que haber de antemano un directorio en `/tmp/project`, el cual servirá para albergar los outputs del programa. En un futuro esto perfectamente podría cambiar a ser `:$HOME/.remote_buster`.

El programa cuenta con 5 comandos específicos:



```
→ project nc localhost 9001
Welcome to buster server, the actual commands are:
/makebuster <site>      to make a new buster
/makemap <site>        to make a new nmap
/getbuster              to see all the busters
/getmap                to see all the nmaps
/getcontent <site>     to get a specific buster
```

- `/makebuster`
- Comando que permite iniciar un nuevo buster, del modo que se guardará el output respectivo de `dirsearch` con el formato `dominio.fecha.buster.log`
- `/makemap`
- Comando que permite iniciar un nuevo escaneo de puerto a un sitio arbitrario, el output de `nmap`, quedará guardado del modo formato: `dominio.fecha.nmap.log`
- `/getbuster`
- comando que permite conseguir todos los busters ya hechos.

- /getmap
- Comando que permite conseguir todos los mapeos ya hechos
- getcontent
- Comando que permite conseguir el output de algún **buster** o mapeo ya terminado

Para conseguir mejor rendimiento del programa, se utilizó paralelismo haciendo que cada nueva conexión entrante sea atendido con un nuevo **hilo**, utilizando **threads**.

Ejemplo de uso

Para correr el programa

```
make  
java Main
```

Con esto, dejaremos el servidor corriendo en el puerto 9001.

```
→ proyecto_final make  
javac Main.java  
→ proyecto_final java Main  
█
```

Al conectarnos al servidor, nos dará el banner simple, por lo que correremos un nuevo scan nmap.

```
→ project nc localhost 9001  
Welcome to buster server, the actual commands are:  
/makebuster <site>      to make a new buster  
/makemap <site>         to make a new nmap  
/getbuster              to see all the busters  
/getmap                 to see all the nmaps  
/getcontent <site>     to get a specific buster  
/makemap f4d3.io  
Doing nmap for: f4d3.io
```

Luego podremos consultar por el archivo si este ya terminó.

```

+ project nc localhost 9001
Welcome to buster server, the actual commands are:
/makebuster <site>      to make a new buster
/makemap <site>        to make a new nmap
/getbuster              to see all the busters
/getmap                to see all the nmaps
/getcontent <site>    to get a specific buster
/getmap
f4d3.io.2020-02-29.nmap.log

```

Como ya terminó, y nos ha devuelto los .log de nmap almacenados hasta el momento, los podremos ver

```

+ project nc localhost 9001
Welcome to buster server, the actual commands are:
/makebuster <site>      to make a new buster
/makemap <site>        to make a new nmap
/getbuster              to see all the busters
/getmap                to see all the nmaps
/getcontent <site>    to get a specific buster
/getcontent f4d3.io.2020-02-29.nmap.log
# Nmap 7.80 scan initiated Sat Feb 29 20:00:29 2020 as: nmap -sC -sV -o /tmp/project/f4d3.io.2020-02-29.nmap.log f4d3.io
Nmap scan report for f4d3.io (3.13.113.106)
Host is up (0.21s latency).
rDNS record for 3.13.113.106: ec2-3-13-113-106.us-east-2.compute.amazonaws.com
Not shown: 995 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 2048 68:ab:0f:dd:e7:31:95:49:6e:5b:90:e4:11:76:37:83 (RSA)
|_ 256  e1:0e:17:cb:78:e5:5d:7c:bb:9a:d6:5e:15:18:6f:64 (ECDSA)
|_ 256  22:3d:77:30:e5:26:b3:18:e7:cb:23:b6:ff:43:6a:12 (ED25519)
80/tcp    open  http     nginx 1.14.0 (Ubuntu)
|_ http-server-header: nginx/1.14.0 (Ubuntu)
|_ http-title: Did not follow redirect to https://f4d3.io/
443/tcp   open  ssl/http nginx 1.14.0 (Ubuntu)
|_ http-server-header: nginx/1.14.0 (Ubuntu)
|_ http-title: f4d3.io
|_ http-trane-info: Problem with XML parsing of /evox/about
|_ ssl-cert: Subject: commonName=f4d3.io
|_ Subject Alternative Name: DNS:f4d3.io, DNS:www.f4d3.io
|_ Not valid before: 2020-02-13T02:20:31
|_ Not valid after: 2020-05-13T02:20:31
|_ ssl-date: TLS randomness does not represent time
|_ tls-alpn:
|_ http/1.1
|_ tls-nextprotoneg:
|_ http/1.1
3333/tcp  closed dec-notes
8080/tcp  closed http-proxy
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sat Feb 29 20:01:20 2020 -- 1 IP address (1 host up) scanned in 51.42 seconds
+ project

```

Para limpiar todo:

`make clean`

Conclusión

Con esto, podemos administrar de manera fácil y simple la creación de nuevos busters/mapeos, como también, iniciar nuevos on the fly, sin la necesidad de estar dentro del servidor utilizado para realizarlos.

TODO things

En un futuro, lo principal a lograr sería poder administrar los actuales **procesos corriendo**, pudiendo pararlos si están consumiendo mucho recurso, con el objetivo de optimizar de buena manera el uso del servidor.