UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ingeniería Industrial y de Sistemas

Informes de exploración de vulnerabilidades en HTB

"De las máquinas: Time, Jarvis Forest "

ELABORADO POR:

- Suárez Moncada, Luis Alfonso
- Mottoccanche Tantaruna, Joseph
- Lau Ma, Chi Jon

Índice

1. Time

1.1. Enumeración

Lo primero a realizar en cualquier máquina es un escaneo rápido con nmap, para esto usamos el comando con los parámetros:

- -p-
- –min-rate=5000
- -v
- -oN puertos

```
Completed Connect Scan at 12:27, 14.95s elapsed (65535 total ports)
Nmap scan report for 10.10.10.214
Host is up (0.12s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 15.16 seconds
```

Figura 1: escaneo con nmap

Entonces encontramos estos dos servicios, algo que podríamos hacer para verificar la versión de los servicios es incluir el -sV. La razón por la cual no usamos esto desde el inicio es porque al analizar todos los puertos en algunos casos hace que se demore considerablemente más, en especial cuando descubre muchos puertos.

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.97 seconds
```

Figura 2: escaneo con version nmap

Un parámetro adicional que podríamos usar para este análisis es el :

- -sV
- -Pn
- –script=Vuln

Analizamos ahora los directorios para ver si encontramos algo con gobuster, esto podría ayudarnos a encontrar alguna carpeta oculta antes de revisar el contenido, para esto usamos un parámetro importante que es el -t 200 que ayuda a que use más hilos.

```
❯ gobuster -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u "http:
//10.10.10.214/" -t 200

===============================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
===============================================================
[+] Mode         : dir
[+] Url/Domain   : http://10.10.10.214/
[+] Threads      : 200
[+] Wordlist     : /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Status codes : 200,204,301,302,307,403
[+] Timeout      : 10s
===============================================================
2021/11/25 12:37:41 Starting gobuster
===============================================================
/images (Status: 301)
/css (Status: 301)
/js (Status: 301)
/javascript (Status: 301)
/vendor (Status: 301)
/fonts (Status: 301)
/server-status (Status: 403)
===============================================================
2021/11/25 12:40:40 Finished
```

Figura 3: fuzzeo con gobuster

Mientras tanto analizamos también la página web que tenemos en el puerto 80, nos encontramos con un validador de json como los que solemos encontrar en internet.

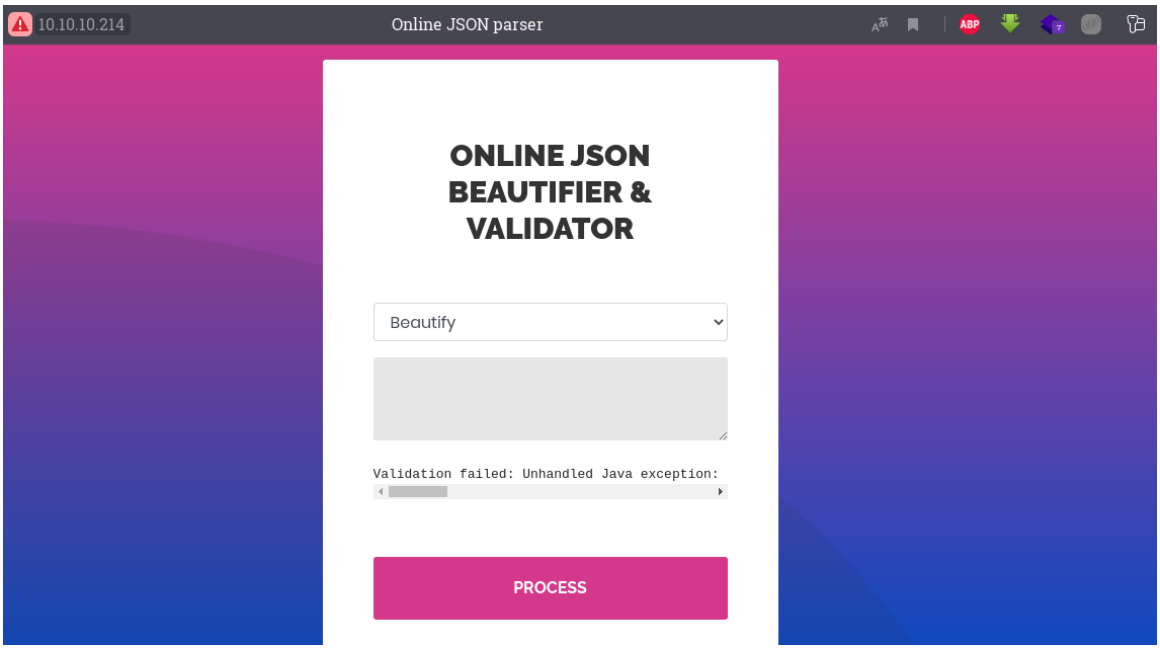

Figura 4: página de json

Entonces tenemos este validador, probamos metiendo un poco de código json, obtenemos este código de https://json.org/example.html. Seleccionamos dos códigos para hacer la prueba porque ambos botaban errores diferentes:

```
{"title": "Sample Konfabulator Widget",
"name": "main_window",
"width": 500,
"height": 500}
```

Con este código te bota el siguiente error:

```
Validation failed:    "title": "Sample Konfabulator Widget",
```

Luego probamos con este código de una línea a ver si había diferencia

```
{"value": "New", "onclick": "CreateNewDoc()"}
```

Con este código te bota el siguiente error:

```
Validation failed: Unhandled Java exception: com.fasterxml.jackson.databind.
exc.MismatchedInputException: Unexpected token (START_OBJECT),
expected START_ARRAY: need JSON Array to contain As.WRAPPER_ARRAY type
information for class java.lang.Object
```

Entonces el segundo error nos da algo más significativo, buscando en google encontramos algunas vulnerabilidades.

1.2. Explotación

1.3. Escalamiento de privilegios

1.4. Post Explotación

1.5. Hardening

2. Jarvis

2.1. Enumeración

2.2. Explotación

2.3. Escalamiento de privilegios

2.4. Post Explotación

2.5. Hardening