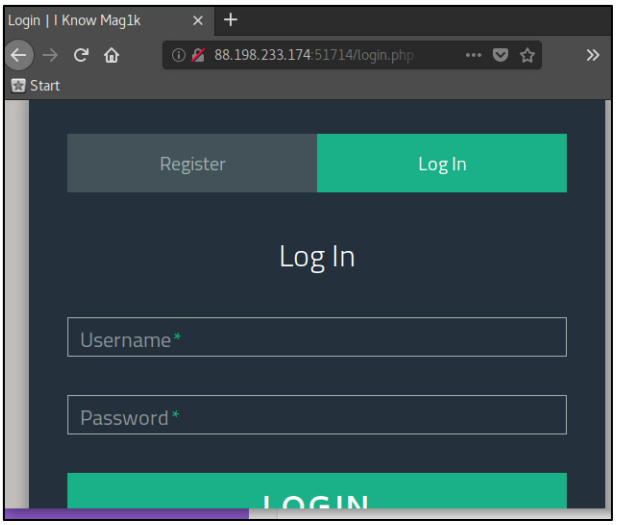
Hack the Box – I know Mag1k challenge write-up

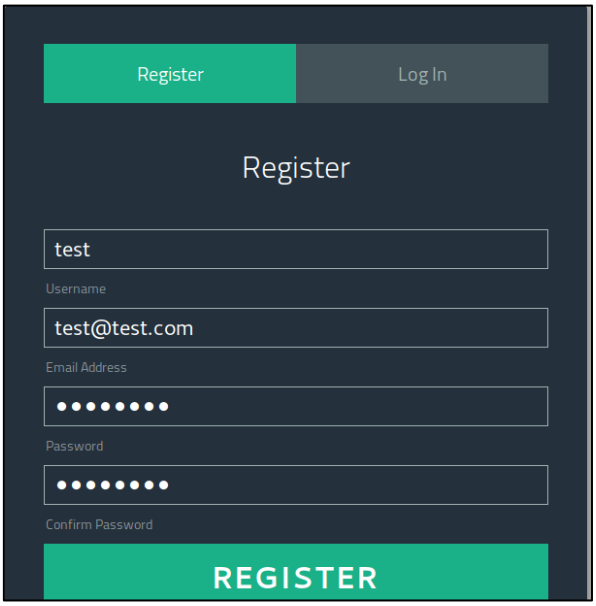
Write-up

Levantamos la instancia del reto web. En mi caso nos asignan la ip y puerto 88.198.233.174:51714

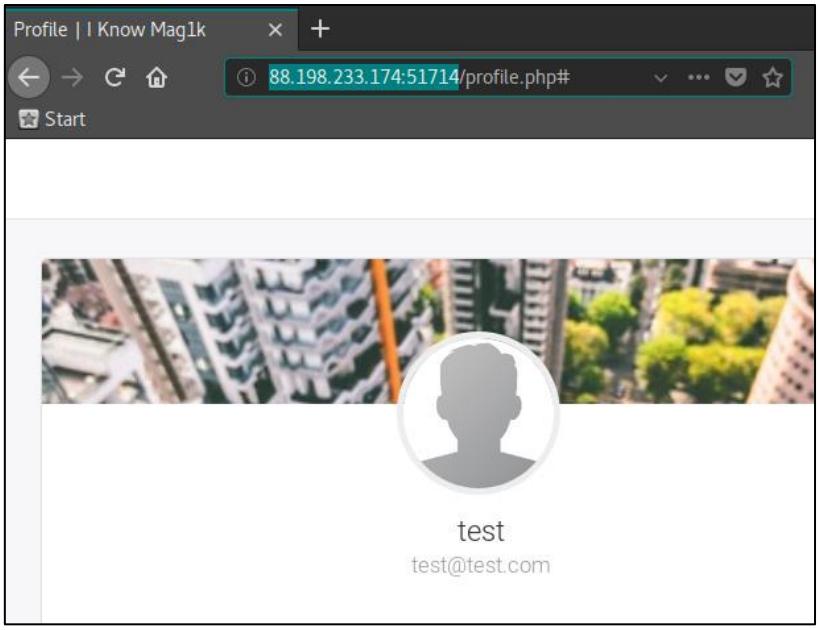
Al entrar vemos una web con esta pinta:



Tras bichear un poco y no conseguir nada, vamos directos al grano. Registramos un usuario pulsando en “Register”. En mi caso el usuario lo llamaremos “test” con la clave que sea, los demás datos no son relevantes.



Hacemos login y entramos. Vemos esto. De momento nada interesante:



Pero usando Burp por ejemplo (hay otras maneras of course!) vemos una cookie de sesión que se llama iknowmag1c:

```
GET /profile.php HTTP/1.1
Host: 88.198.233.174:51714
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=a3oujmnnc132vd2radmtq896n7; iknowmag1k=woiqHfCsYAjNXOaZICj3p%2F81xwhryW0Gt2cHB7EN9KyKOE53cAslBw%3D%3D
DNT: 1
```

Parece base64 pero al decodificarla no dice nada… en principio. La transformamos ya que está en “URL encoding”. Tiene símbolos de igual “=” que están representados por “%3D”. Vamos a ponerla normal… Si usas Burp, la seleccionas y pulsas Ctrl+Shift+u y directamente nos la descodifica, y hasta nos pinta un botón para copiarla al portapapeles… bendito Burp! Nos queda en mi caso:

woiqHfCsYAjNXOaZICj3p/81xwhryW0Gt2cHB7EN9KyKOE53cAslBw==

Bien, aquí el siguiente paso es intentar abusar de la cookie. ¿Cómo? Pues realizando un “Padding Oracle Attack” sobre ella. Esto se suele hacer cuando se sospecha que la cookie es “casera” y que no se ha generado con una librería bien diseñada para estos fines. Si “olisqueamos” que la cookie puede estar hecha un poco “de andar por casa”, igual no se genera con la suficiente entropía y se puede intentar adivinar parte de ella. Algún link interesante sobre esto:

http://www.hackingarticles.in/hack-padding-oracle-lab/

¿Cómo se ataca esto? Vamos a verlo…

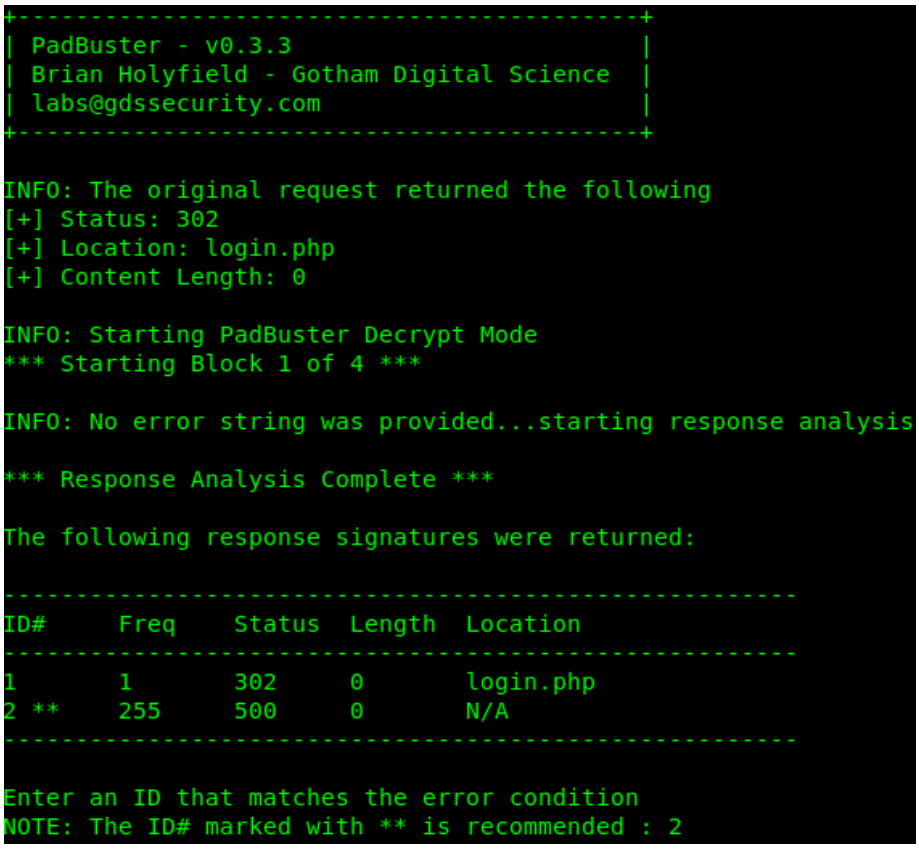
Descargamos esta herramienta hecha en perl (Padbuster):

https://github.com/GDSSecurity/PadBuster

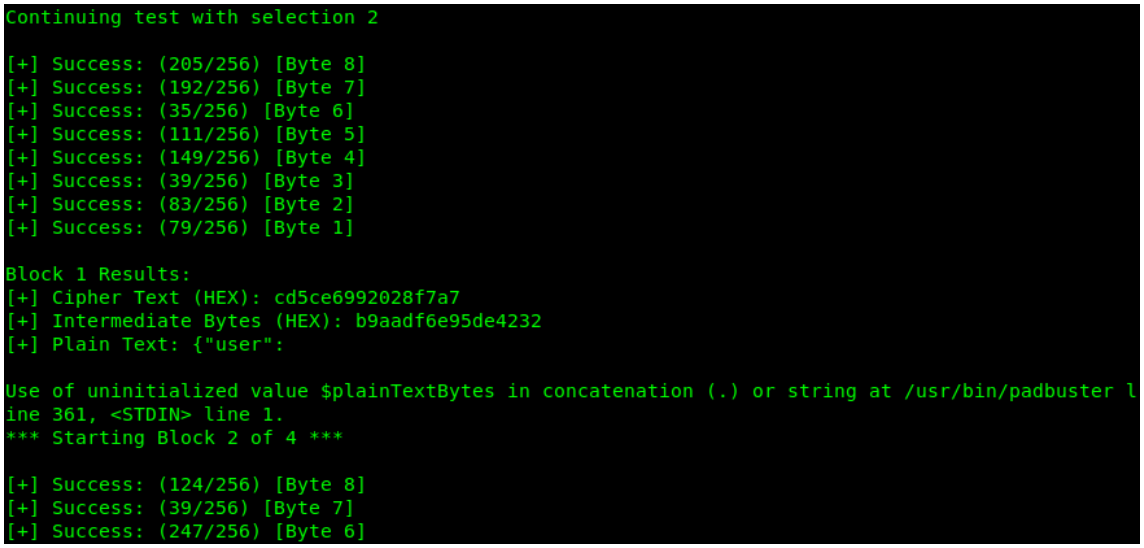
Y lanzamos el comando (tardará un rato!):

padbuster http://88.198.233.174:51714/profile.php woiqHfCsYAjNXOaZICj3p/81xwhryW0Gt2cHB7EN9KyKOE53cAslBw== 8 --cookies iknowmag1k=woiqHfCsYAjNXOaZICj3p/81xwhryW0Gt2cHB7EN9KyKOE53cAslBw==

Esto empezará a hacer peticiones a la página del perfil del usuario generando cookies y más cookies y más cookies…. Así hasta intentar ver como se genera la dichosa cookie. Nos preguntará por el camino que como identifica un error… Si es así, pulsaremos el 2 que hace referencia al status code 500 que es un error y seguimos:



Esto continúa haciendo su magia…



Al final nos saca esto que es muy interesante:

```
-------------------------------------------------------
** Finished ***

[+] Decrypted value (ASCII): {"user":"test","role":"user"}

[+] Decrypted value (HEX): 7B2275736572223A2274657374222C22726F6C65223A2275736572227D030303

[+] Decrypted value (Base64): eyJ1c2VyIjoidGVzdCIsInJvbGUiOiJ1c2VyIn0DAwM=

-------------------------------------------------------
```

Vemos que ha conseguido decodificar la cookie y vemos lo que lleva por debajo.

Bien, pues el siguiente viendo que hay un parámetro “role” que nos gustaría que fuera “admin” en lugar de “user” como es ahora, intentar generar una cookie que tuviera ese texto deseado… así que lanzamos el comando (escapando las comillas de dentro, dejamos las de fuera):

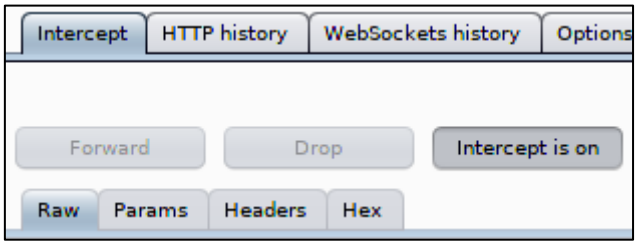
padbuster http://88.198.233.174:51714/profile.php woiqHfCsYAjNXOaZICj3p/81xwhryW0Gt2cHB7EN9KyKOE53cAslBw== 8 --cookies iknowmag1k=woiqHfCsYAjNXOaZICj3p/81xwhryW0Gt2cHB7EN9KyKOE53cAslBw== --plaintext="{\"user\":\"test\",\"role\":\"admin\"}"

Y le dejamos que haga su magia, igual que antes… y si pregunta de nuevo lo del error respondemos con el 2 señalando los status 500 como la otra vez. Tardará un rato largo… y al terminar nos queda esto:

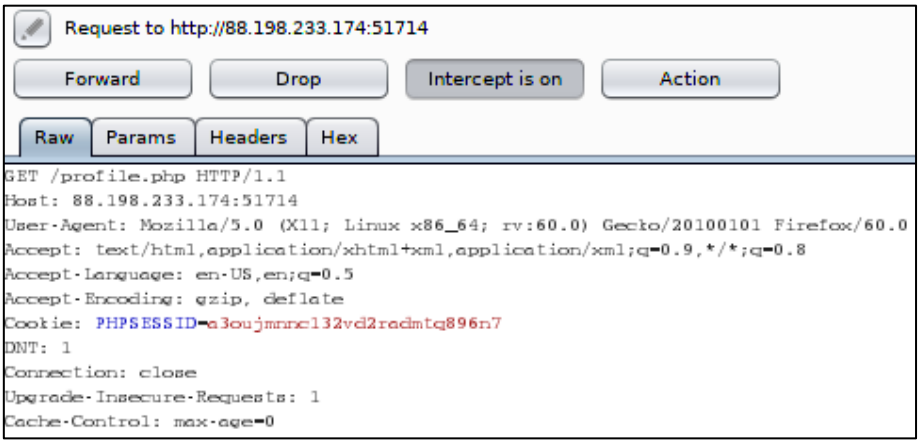
```
-------------------------------------------------------
** Finished ***

[+] Encrypted value is: %2B3ROJgSq3IrxRYmHbnTNi1ArH3JXLFFyNQMrqEHXZZwAAAAAAAAAAA%3D%3D
```

Esa sería la cookie que supuestamente tendría el “role” en “admin”… ahora solo queda inyectarla. Lo podemos hacer de mil maneras… la mía, interceptamos las peticiones en Burp para que se pare cuando refresquemos el navegador pulsando el botón “Intercept is on”:

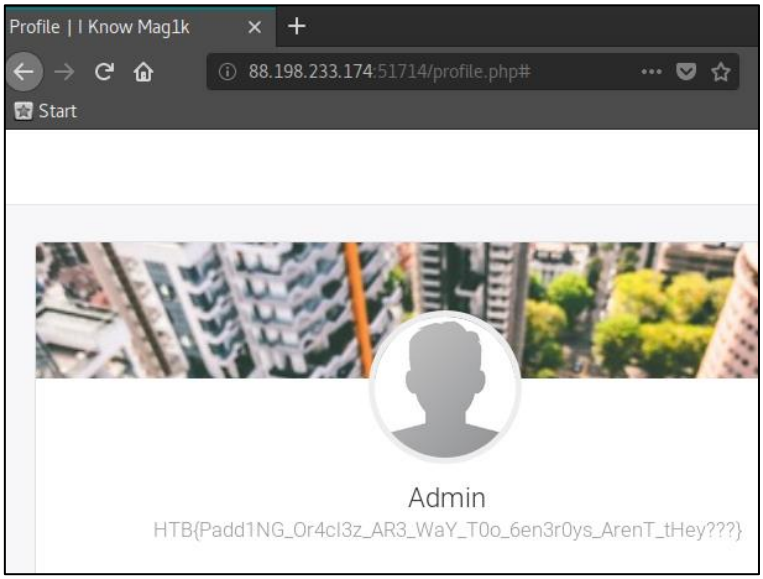


Refrescamos la página del perfil de nuestro usuario en el navegador y al pararse en burp, editamos la petición que inicialmente tiene esta pinta:



Como veis no tiene la cookie “iknowmag1c”, pero se la ponemos nosotros, no pasa nada… la inyectamos y utilizamos la que nos ha generado padbuster (la ponemos URL encoded). La petición quedaría así:

Soltamos la petición y dejamos que continúe pulsando el botón “Intercept is on”. Y en el navegador veremos esto:



Tacháaaaan… ha funcionado. Hemos inyectado la cookie de admin y nos ha dado el ansiado flag:

HTB{Padd1NG_Or4cl3z_AR3_WaY_T0o_6en3r0ys_ArenT_tHey???}

Espero que os haya gustado ☺

Aquí están mis datos de contacto:

oscar.alfonso.diaz@gmail.com

v1s1t0r.1s.h3r3@gmail.com