# 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.

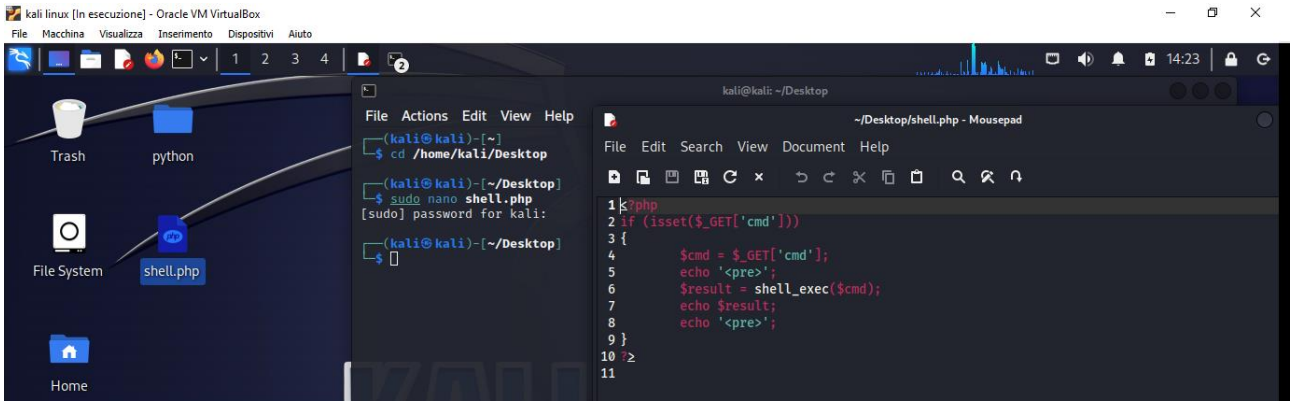
**Consegna**:

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*

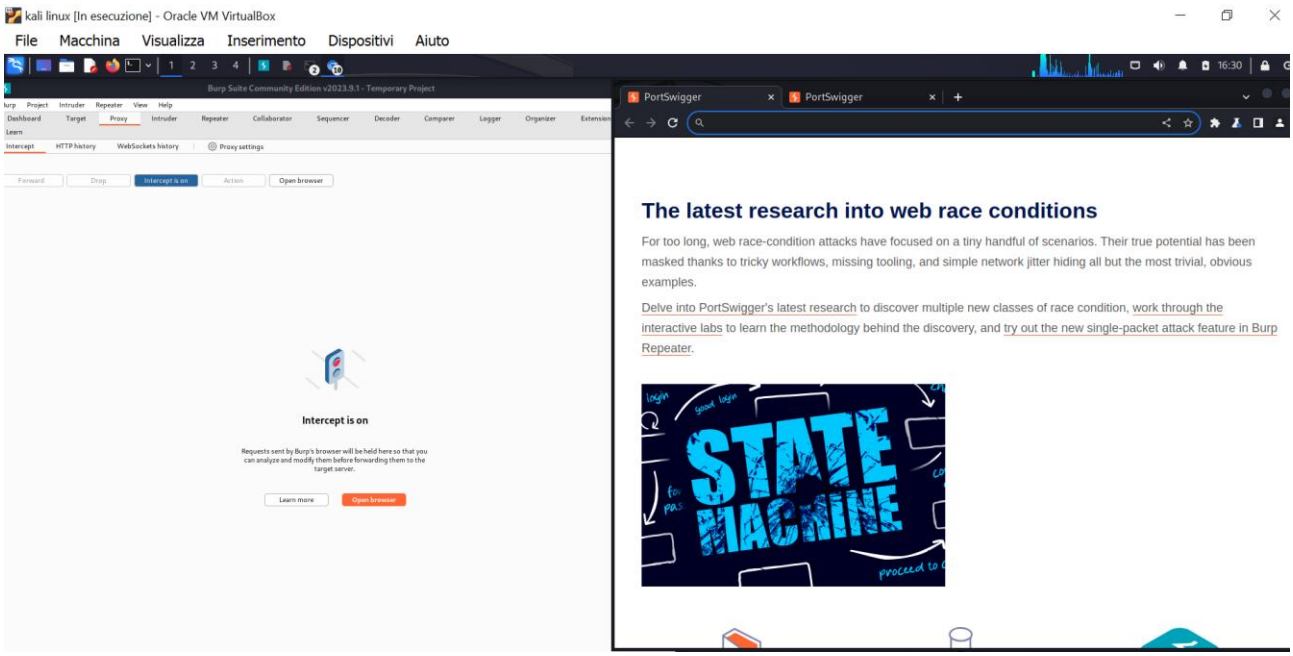


*Grafico 3*

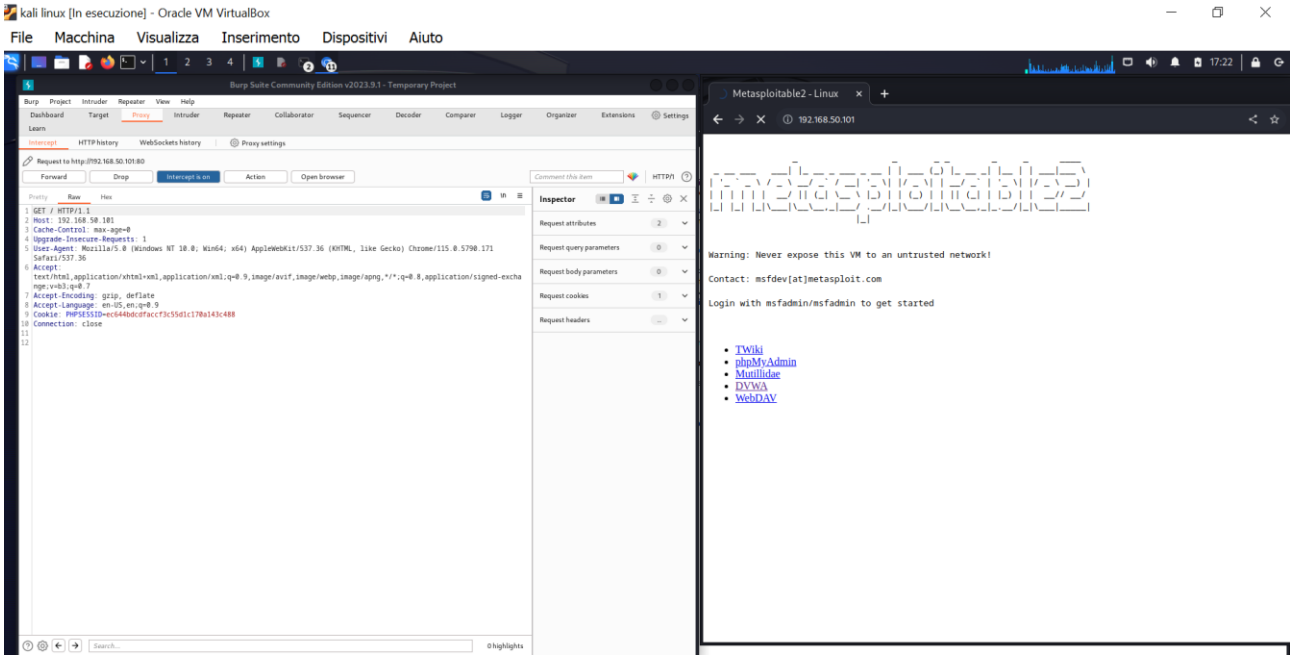
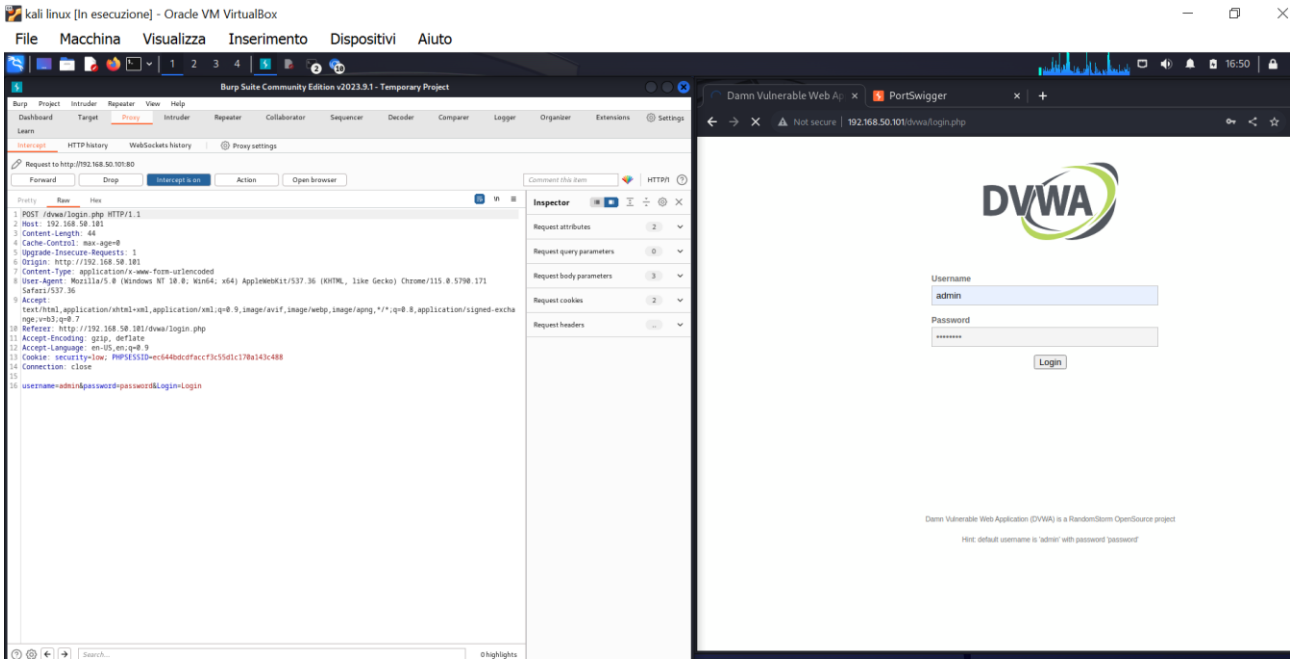
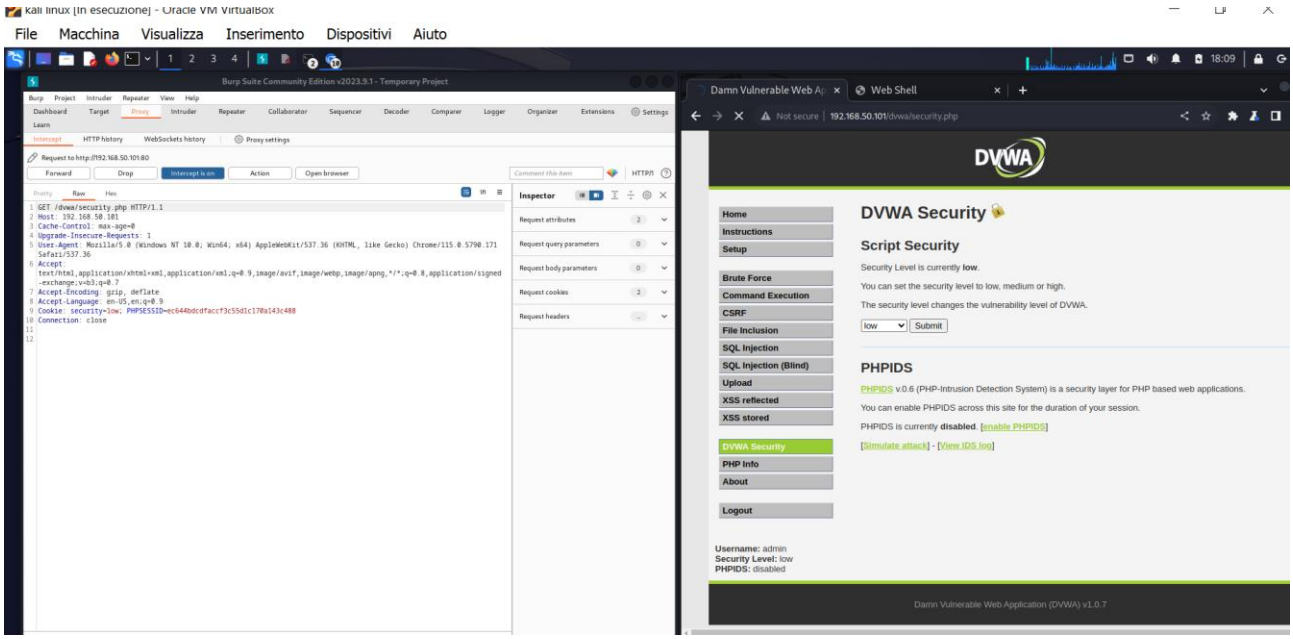



*Grafico 4*



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*

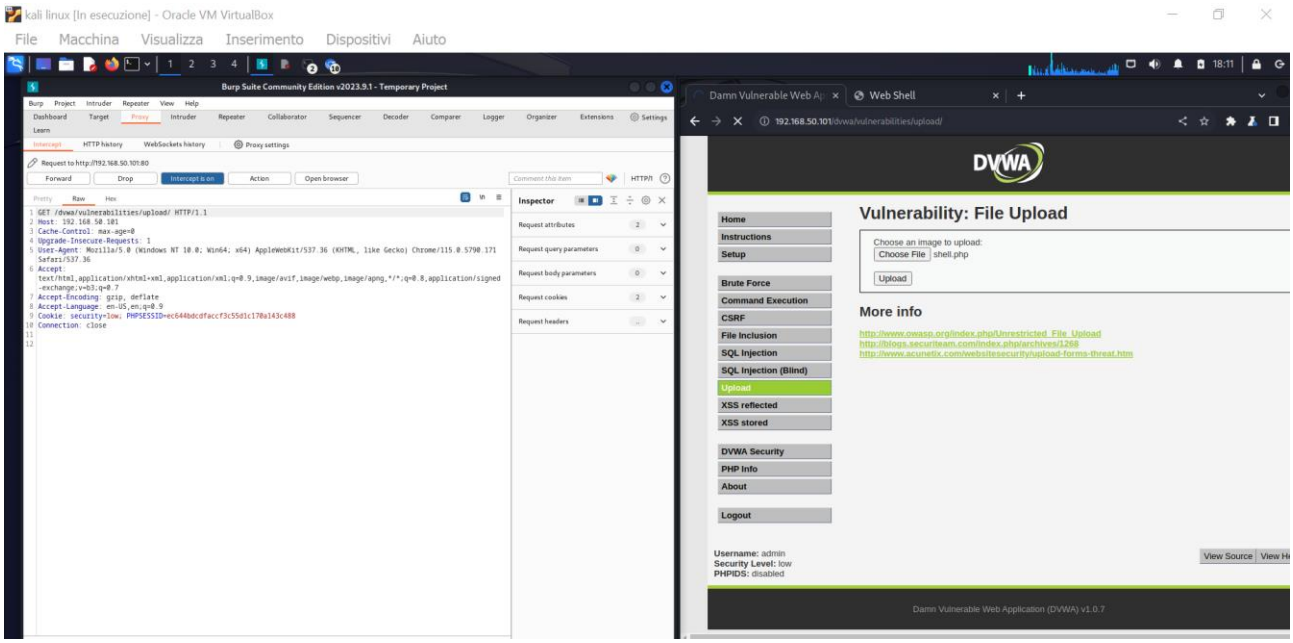


*Grafico 7*

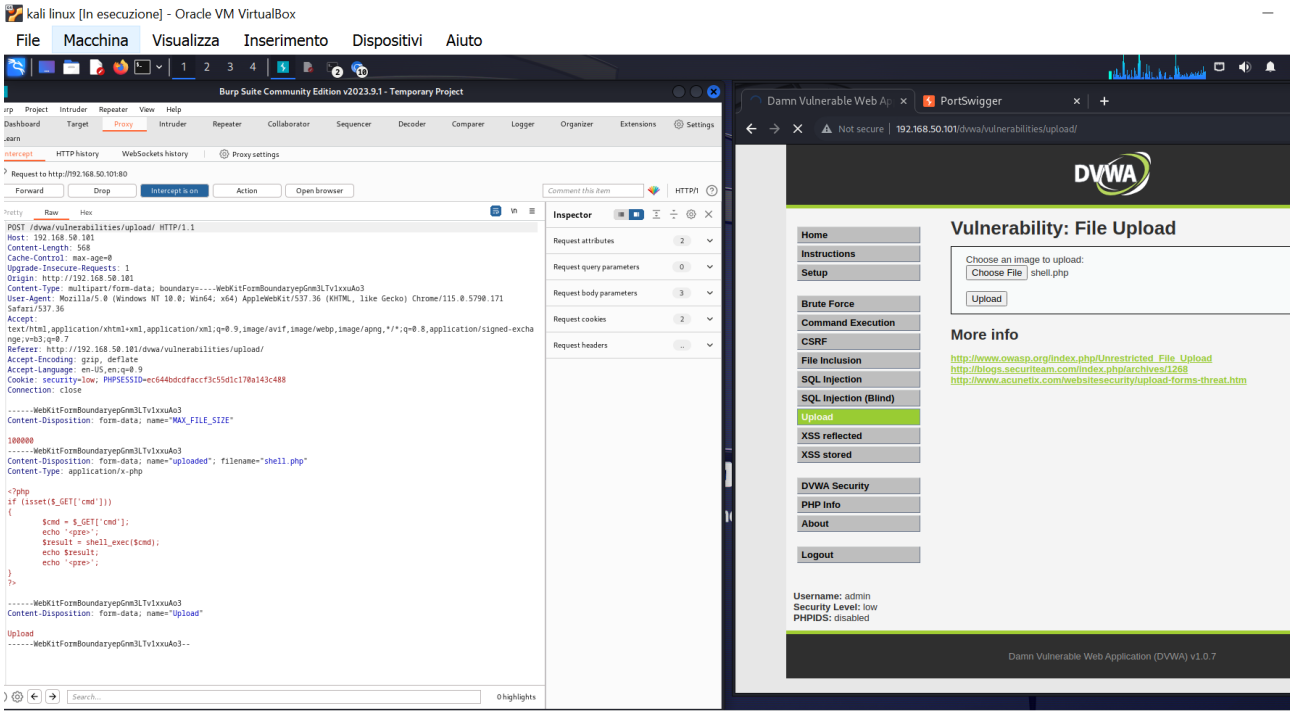
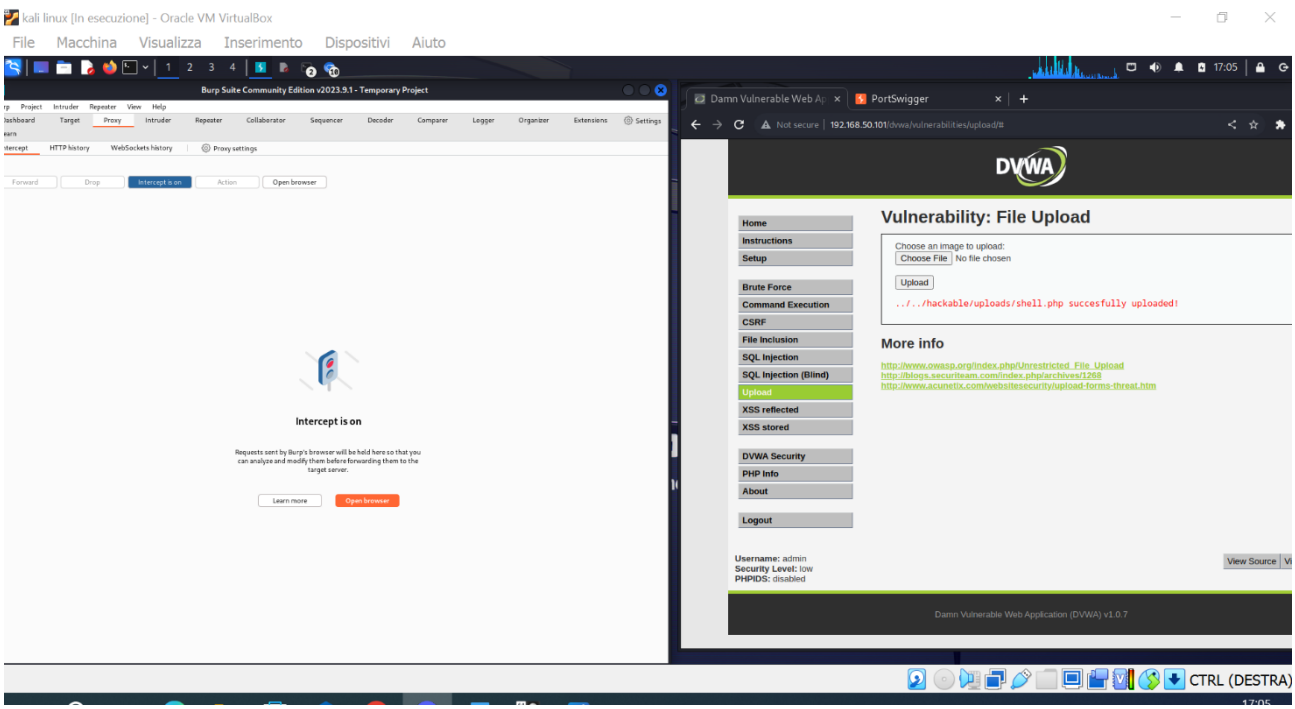



*Grafico 8*



- 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*

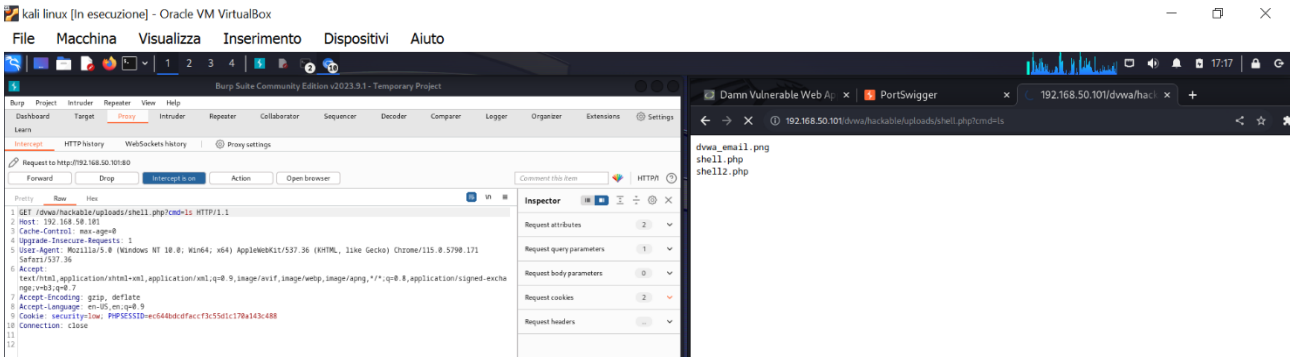
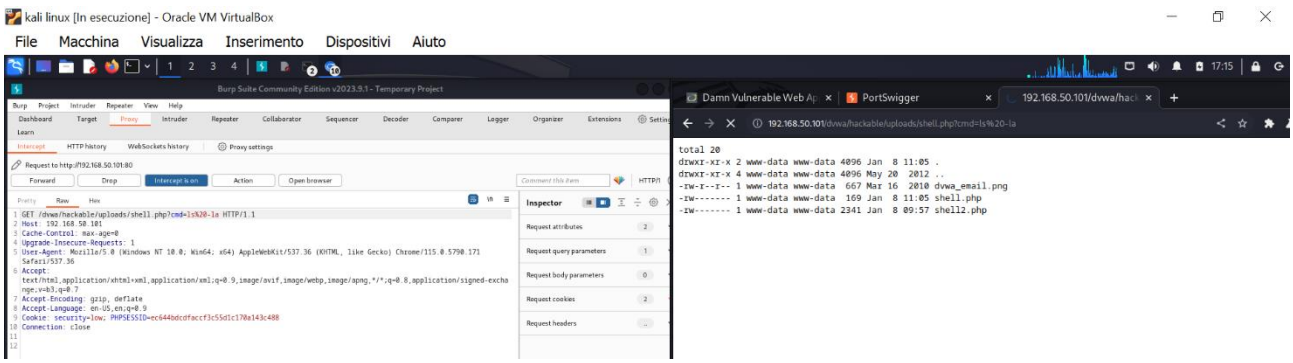



*Grafico 10*



## 3. 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*

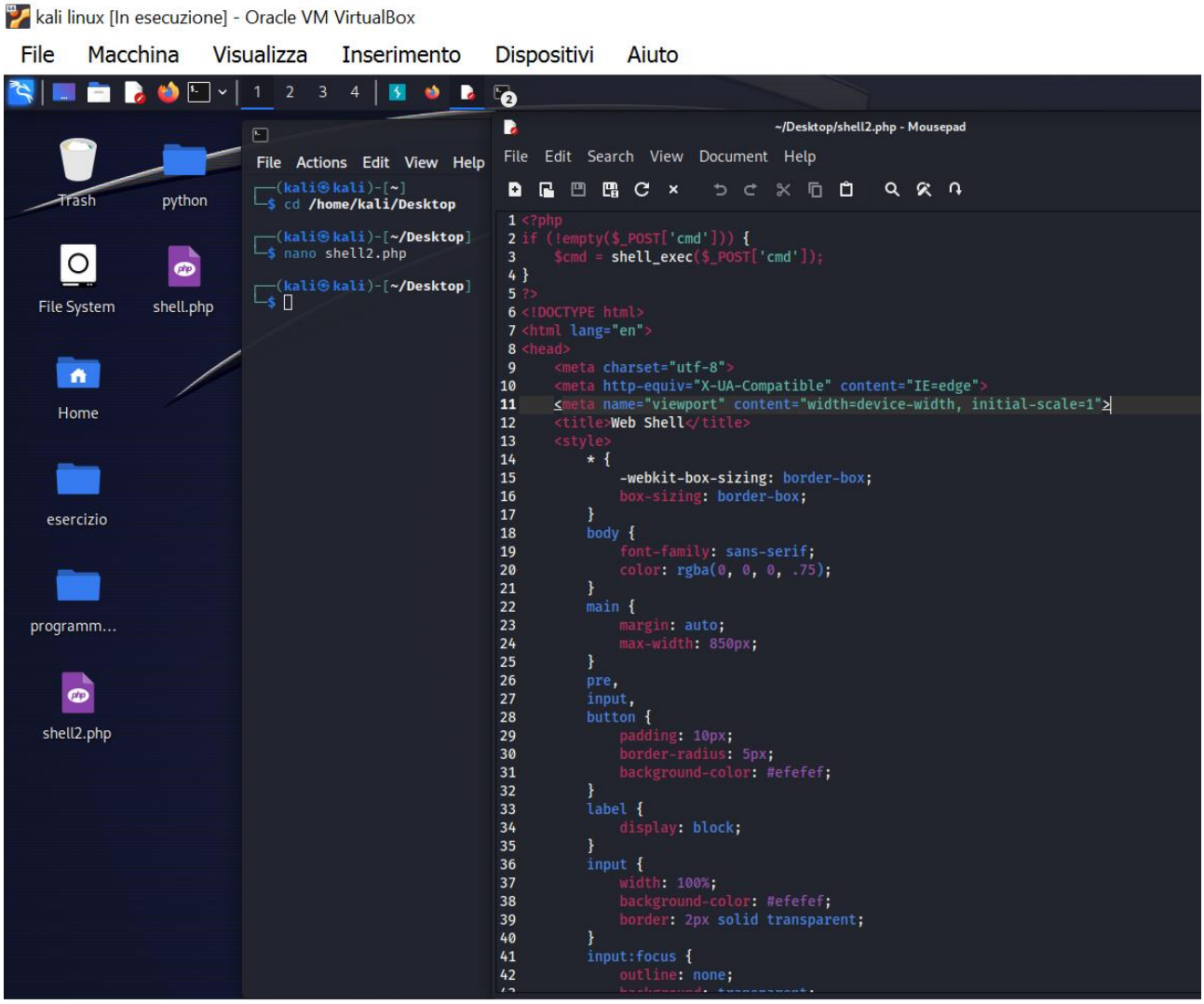


*Grafico 12*

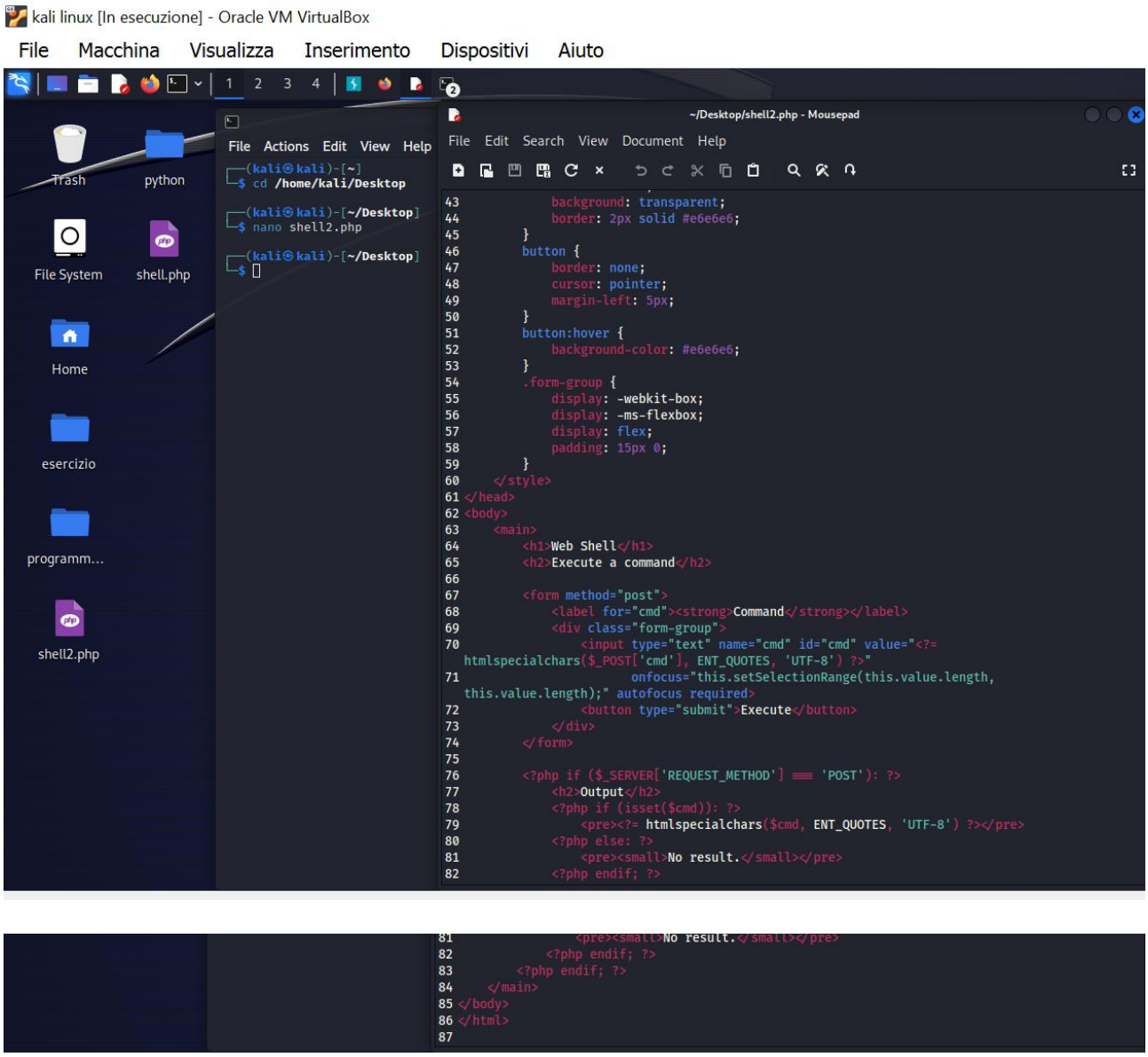



*Grafico 13*

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*

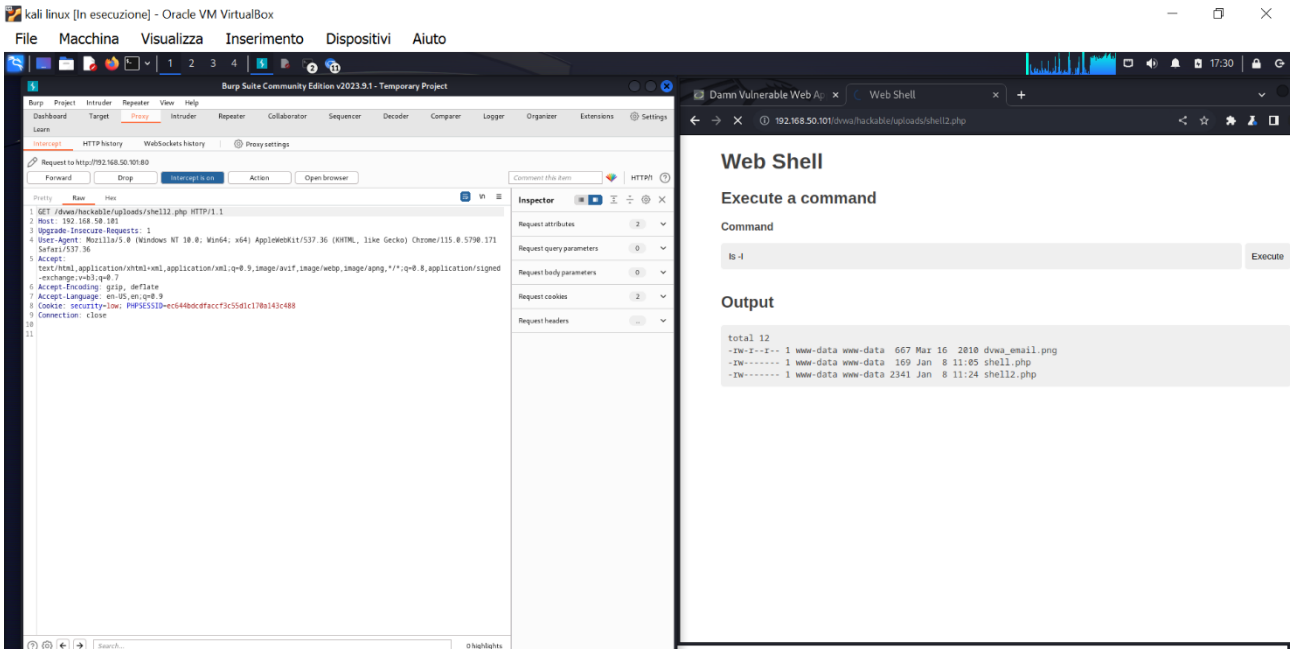



*Grafico 15*