Guida della vulnerabilità
File Inclusion o Directory Traversal

Introduzione

Questa è la guida della vulnerabilità di tipo **File Inclusion o Directory Traversal**, seguendo questa guida riuscirai a sfruttare la vulnerabilità all'interno di HackerLab.

Questo tipo di vulnerabilità permette ad un malintenzionato di poter sfruttare delle falle nelle validazioni dei nomi di file forniti dall'utente, sfruttando caratteri come per esempio "/" per poter navigare all'interno della macchina che ospita l'applicativo vulnerabile.

Requisiti

- Browser (Nella guida viene utilizzato Chrome)
    - https://support.google.com/chrome/answer/95346

Guida

Per eseguire questa vulnerabilità si andrà a sfruttare un parametro di tipo GET in una richiesta specifica. Il percorso in questione è il seguente: /image/?file_name=
Ho ricavato questo percorso grazie ad un articolo già presente in HackerLab che fa uso di immagini. L'articolo in questione è accessibile al percorso /post/2.
Una volta nella pagina dell'articolo ho ricavato il percorso selezionando l'immagine, cliccando il tasto destro e aprendola in una nuova pagina.

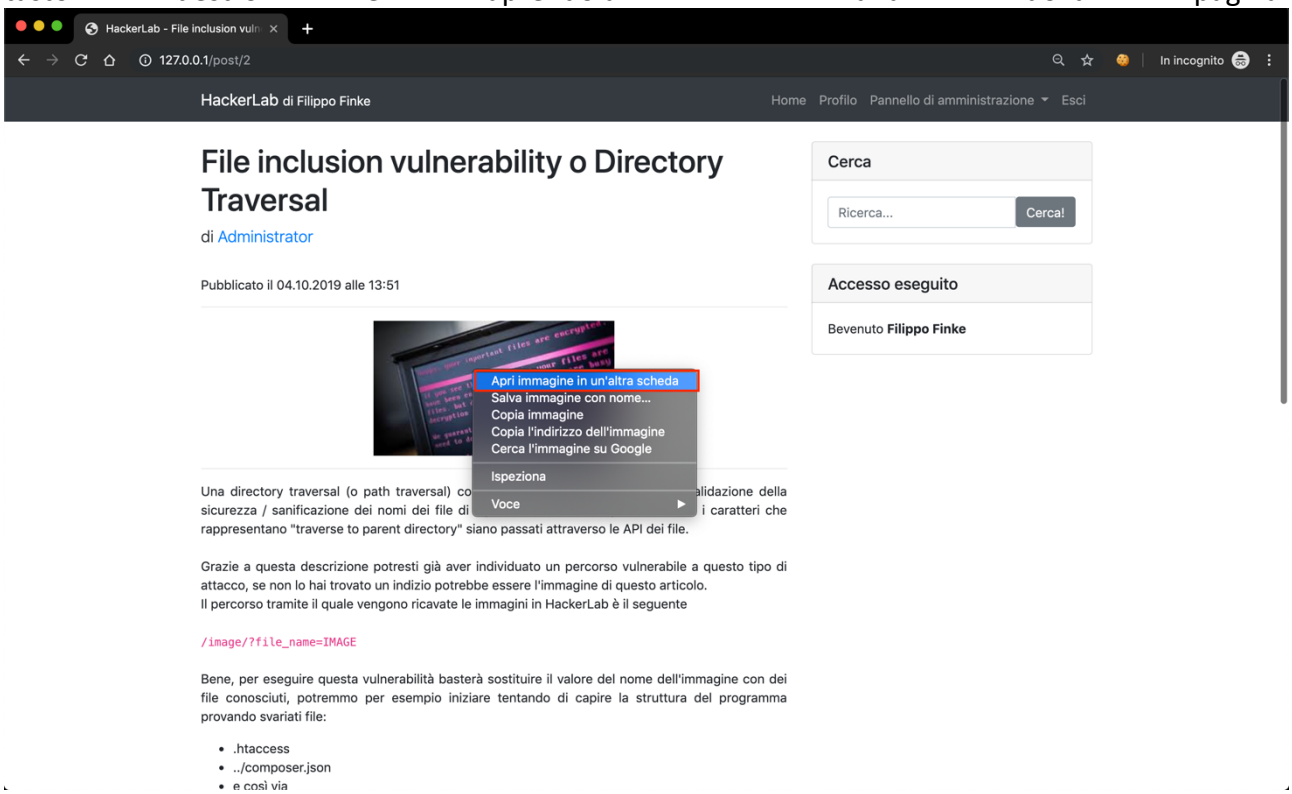

Figura 1 Aprire immagine in una nuova scheda

Verrà quindi aperta l'immagine in una nuova scheda separata.

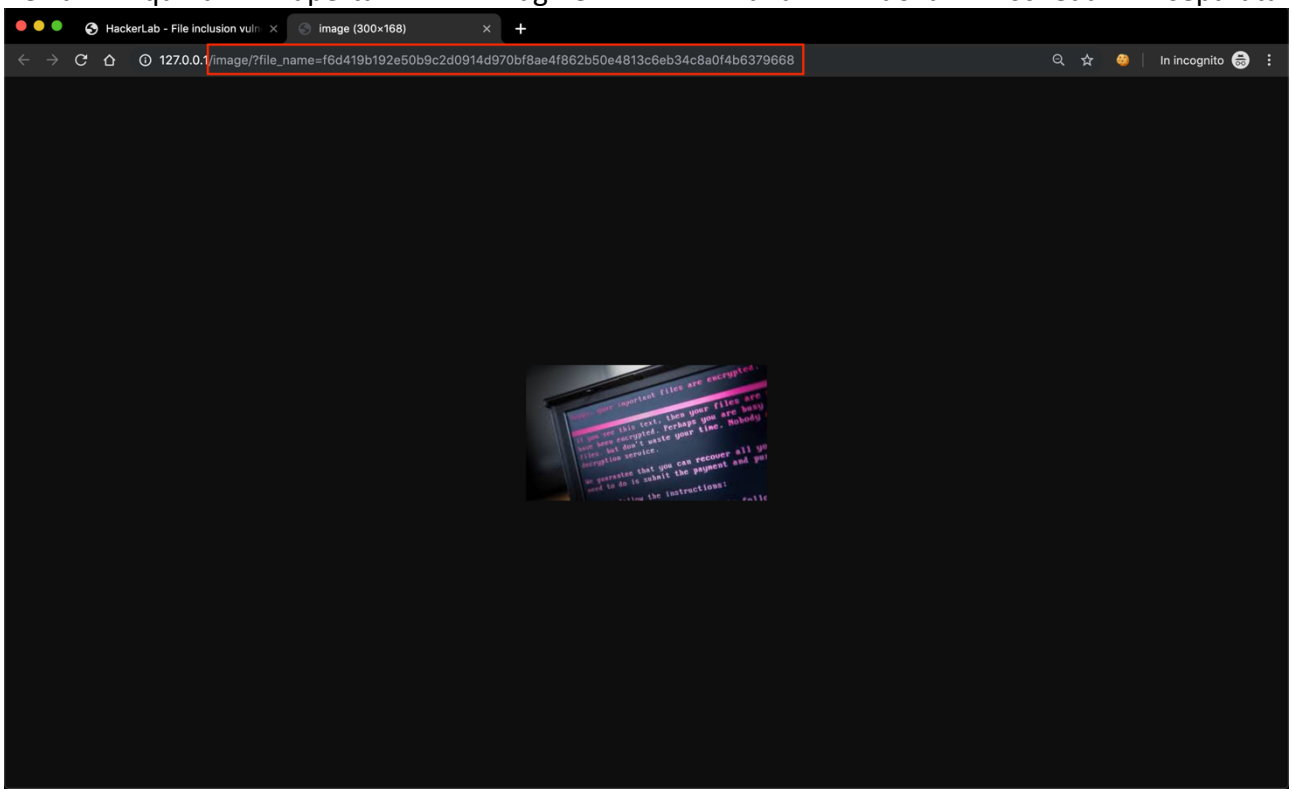

Figura 2 Percorso per caricare le immagini

Possiamo quindi notare il percorso /image/ che richiede un parametro file_name che possiamo presumere indichi il nome del file da caricare.
La vulnerabilità consiste proprio nello sfruttare questa richiesta per caricare dei file specifici al posto di immagini.
In questo caso ho provato a caricare un file molto comune ed utilizzato in progetti sviluppati in php, ovvero composer.json.
Ho quindi eseguito la richiesta a /image/?file_name=../composer.json in questo caso ho presunto che la cartella di salvataggio delle immagini sia un livello superiore alla cartella principale del progetto. Il risultato è stato il seguente:

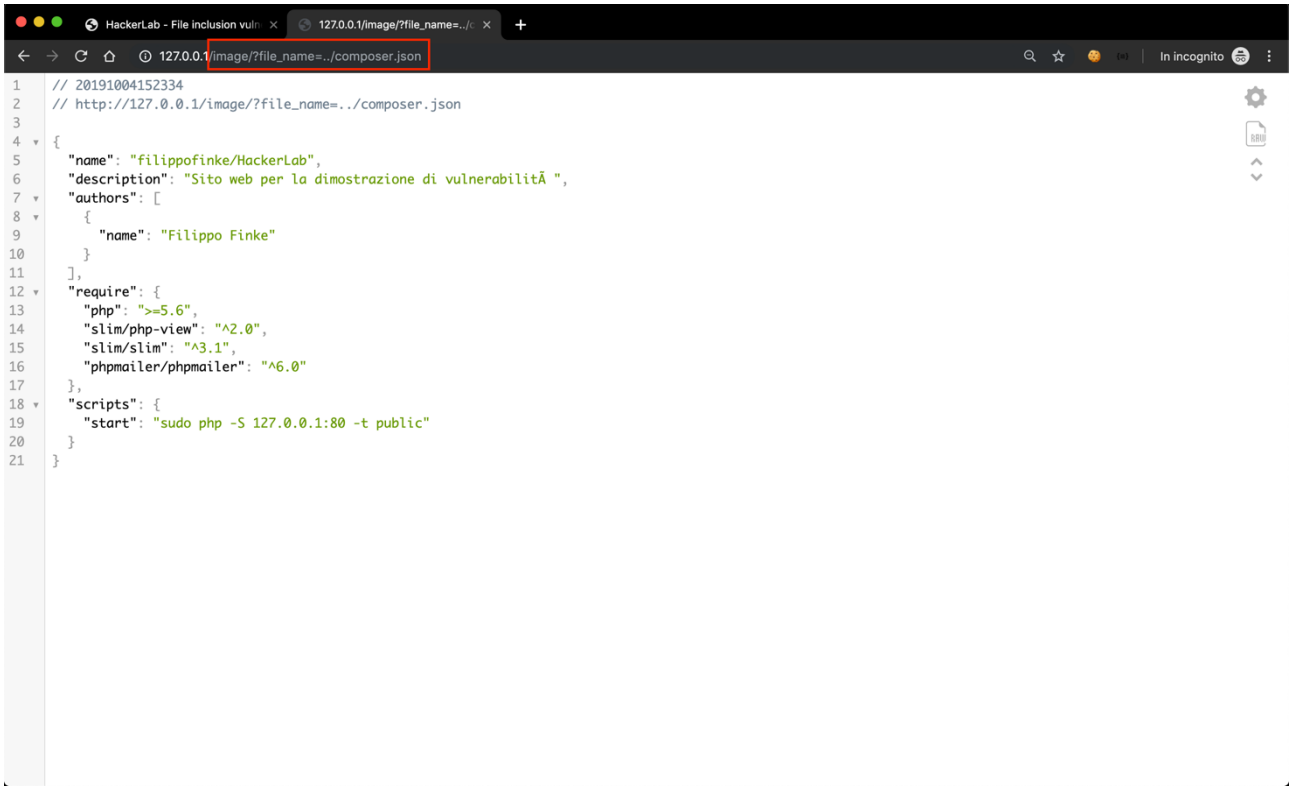

Figura 3 Caricare un file attraverso il parametro file_name

Come possiamo notare il file composer.json è stato caricato e mostrato all'utente. Attraverso questa vulnerabilità si può accedere alla maggior parte del sistema operativo che mette in funzione il servizio web.

Altri esempi più complessi che possono essere raggiunti attraverso altre vulnerabilità presenti nel sito:
/image/?file_name=../public/index.php

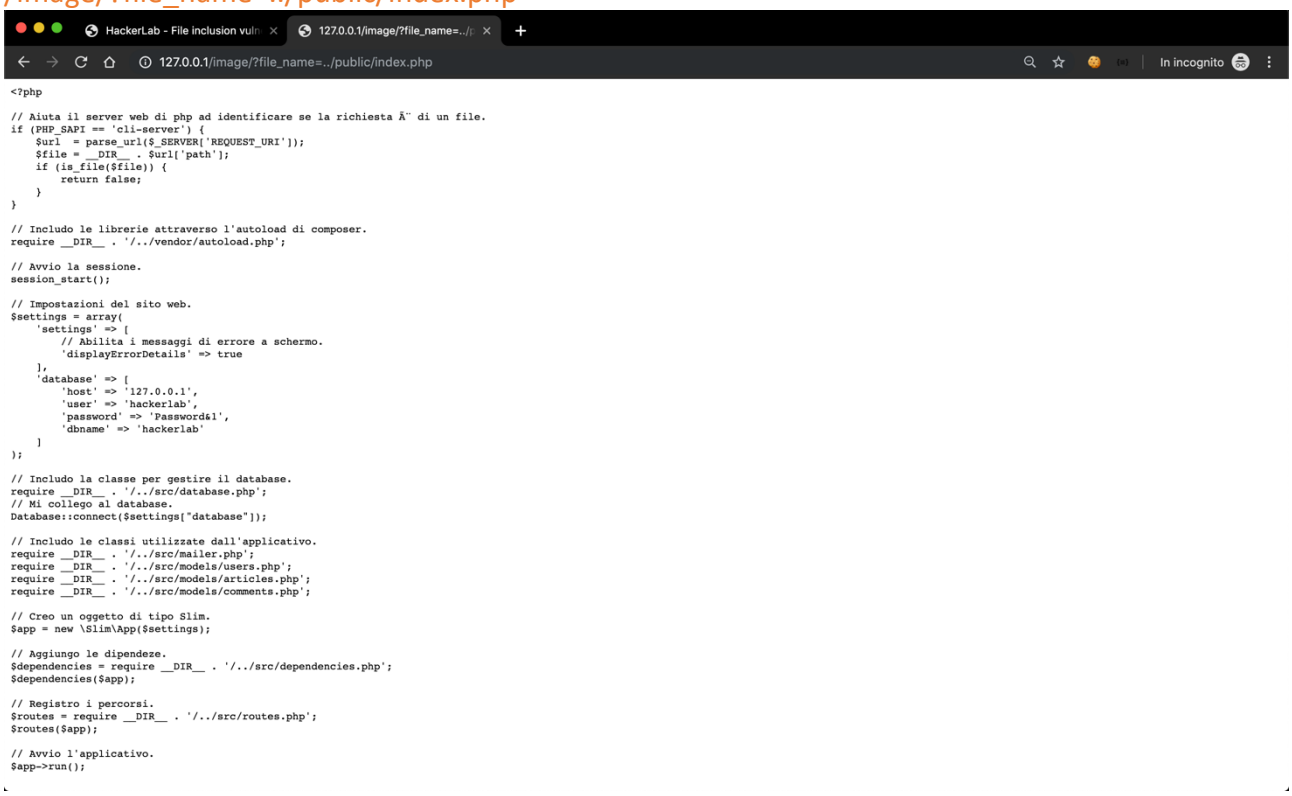

Figura 4 Esempio complesso

In questo caso viene mostrato il codice sorgente della pagina principale che avvia l'applicativo.

Un altro esempio più complesso può essere navigare attraverso la cartella di gestione di quello che riguarda GitHub.
/image/?file_name=../../.git
Può essere navigata completamente se ne si conosce il formato (https://www.siteground.com/tutorials/git/directory-structure/)