18.09.2023

Курс:

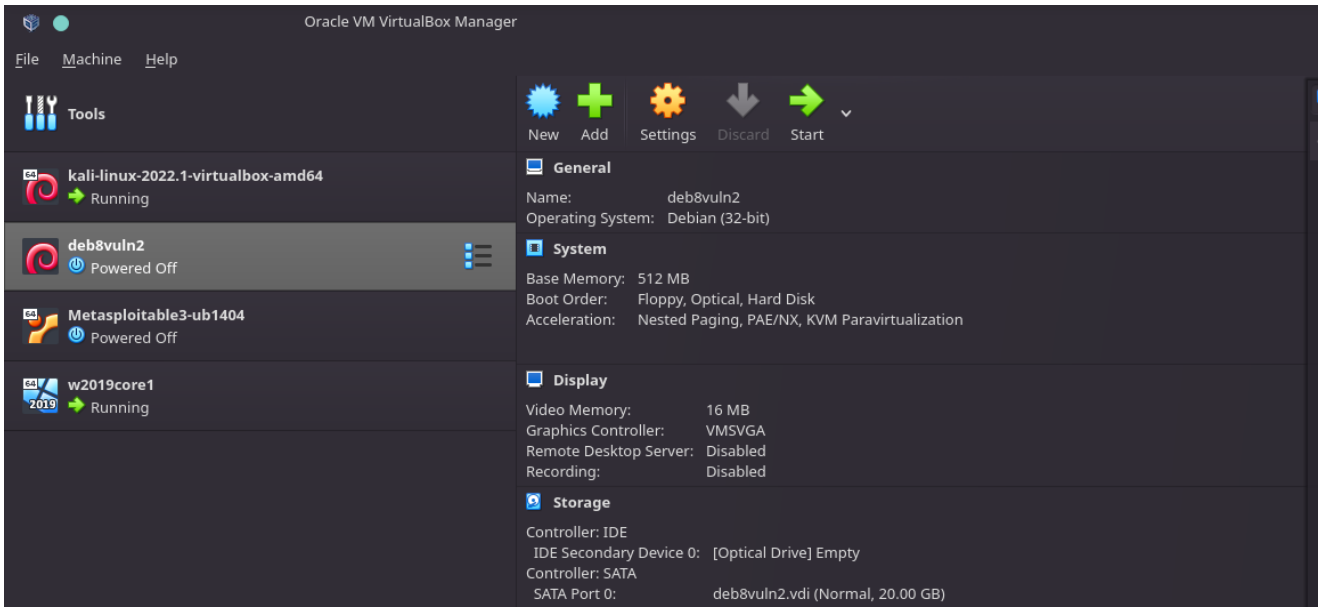
Практическая работа к уроку № Lesson_1

--

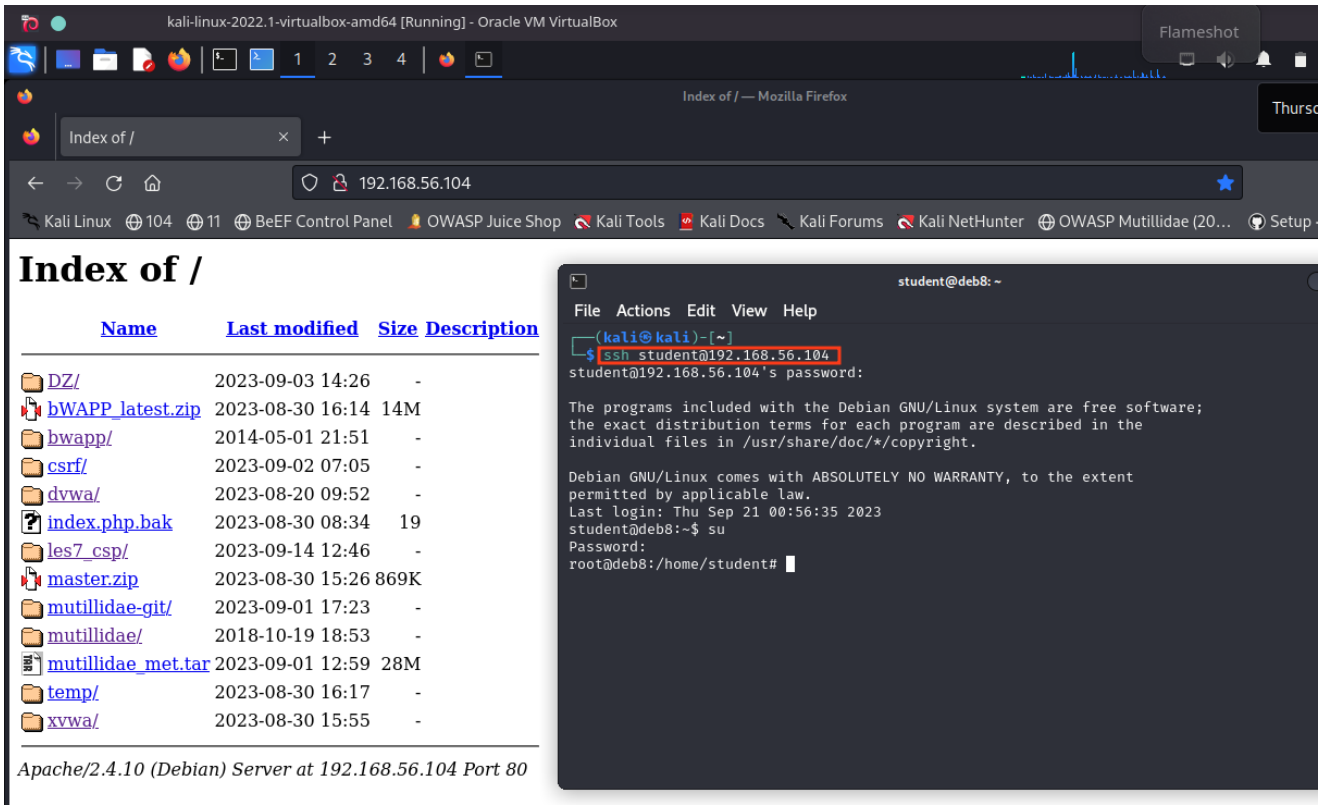
Введение в server-side-уязвимости

Задание_1:

Создайте удаленный сервер, подключитесь к нему по ssh.



- ssh user@ip




```
1 apt-get install mysql-server mysql-client
2 apt-get install apache2-mpm-prefork
3 apt-get install php5 libapache2-mod-php5
4 apt-get install php5-mysql php5-curl php5-gd php5-intl php-pear php5-imagick php5-imap
  php5-mcrypt php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
5 apt-get install git
6 apt-get install unzip
```

```
root@serv:/home/student# cd /var/www/html
root@serv:/var/www/html# wget https://sourceforge.net/projects/bwapp/files/bWAPP
/bWAPP_latest.zip
--2019-02-16 08:42:55--  https://sourceforge.net/projects/bwapp/files/bWAPP/bWAP
P_latest.zip
```

```
root@serv:/var/www/html# chmod 777 -R bwapp/
```

```
 GNU nano 2.2.6                File: settings.php


Malik Mesellem
Twitter: @MME_IT

bWAPP is licensed under a Creative Commons Attribution-NonCommerci

*/

// Database connection settings
$db_server = "localhost";
$db_username = "root";
$db_password = "Qwerty1";
$db_name = "bWAPP";

// SQLite database name
$db_sqlite = "db/bwapp.sqlite";

// SMTP settings
$smtp_sender = "bwapp@mailinator.com";
```

Задание_2:

Зарегистрируйте домен, делегируйте домен на какой-либо NS-сервер.

- Доделываю

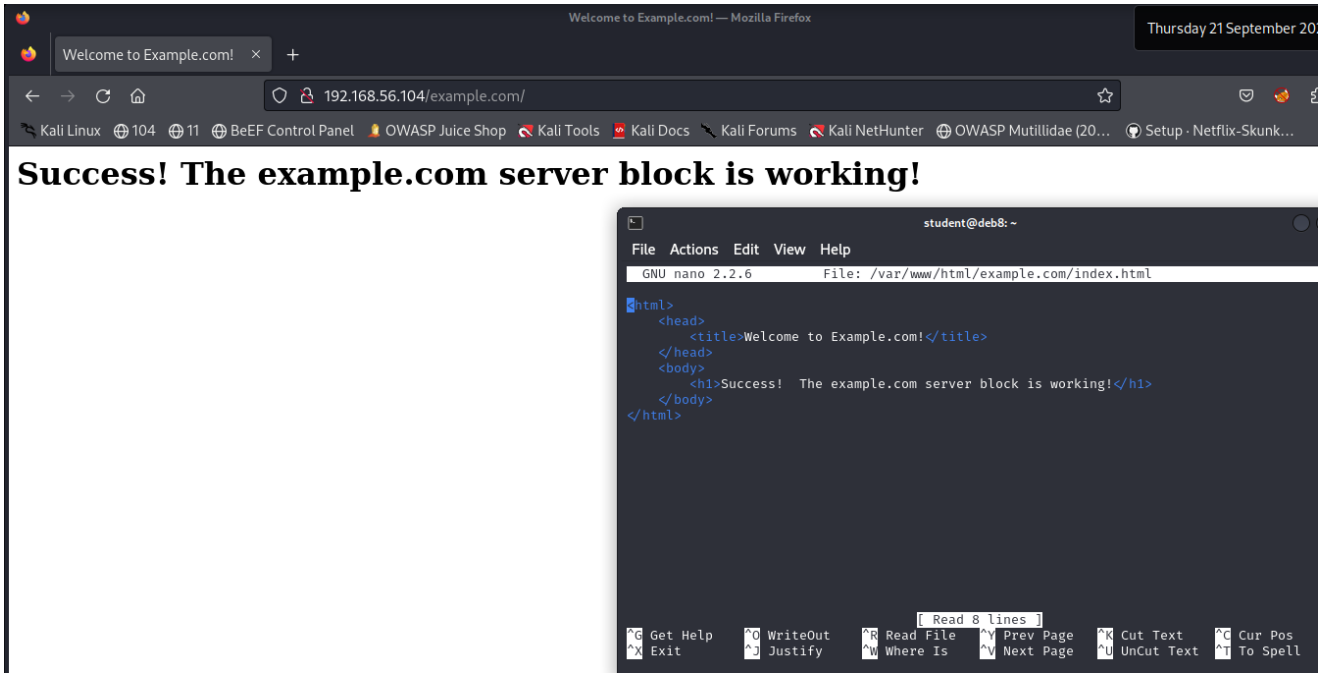
Задание_3:

Установите nginx, создайте тестовую страницу. Убедитесь, что, обратившись по своему доменному имени из интернета, вы получите тестовую страницу.

- Установил

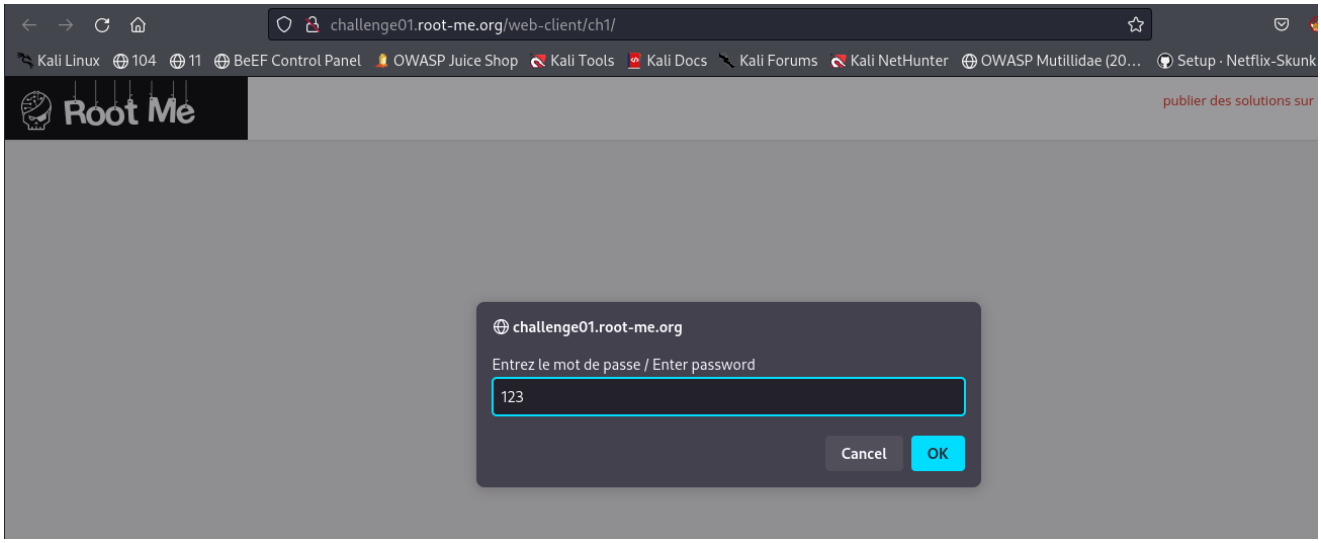
```
su
apt update
apt install nginx
(ufw app list)
(ufw allow 'Nginx HTTP')
(ufw status)
systemctl status nginx
ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\/.*$//'
curl -4 icanhazip.com
sudo systemctl enable nginx
mkdir -p /var/www/html/example.com
(chmod -R 755 /var/www/html/example.com)
nano /var/www/html/example.com/index.html
```

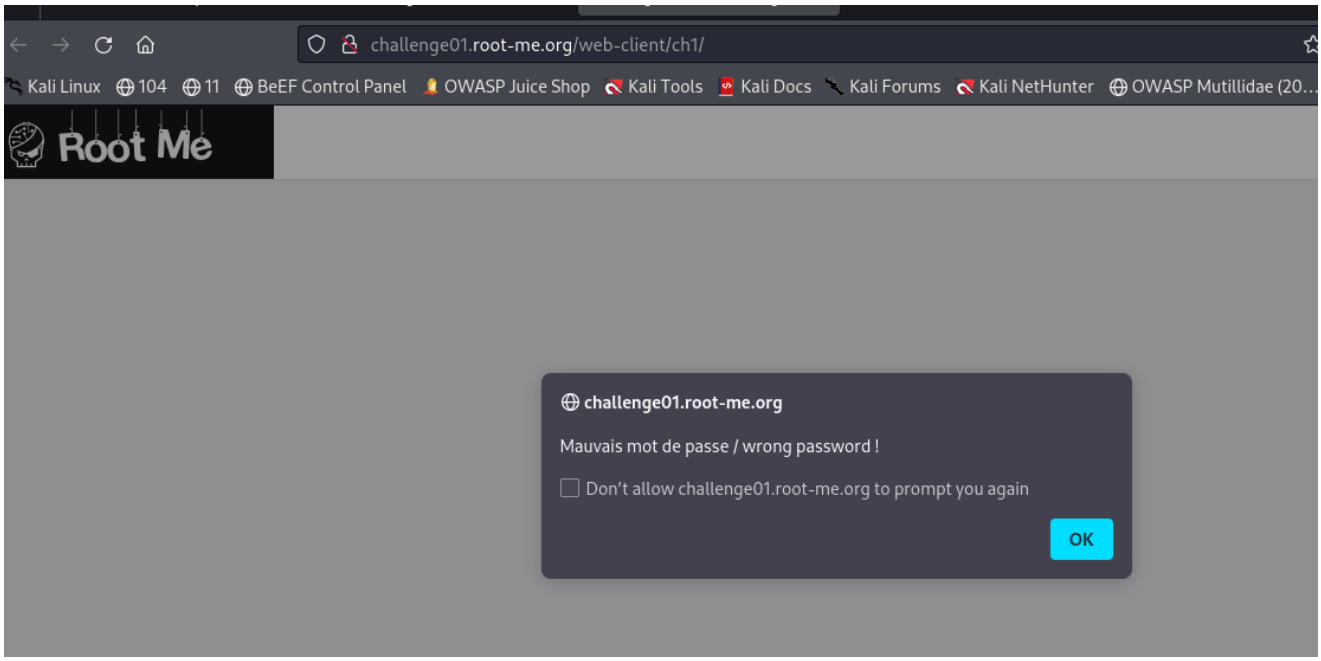
```
<html>
    <head>
        <title>Welcome to Example.com!</title>
    </head>
    <body>
        <h1>Success!  The example.com server block is working!</h1>
    </body>
</html>
```



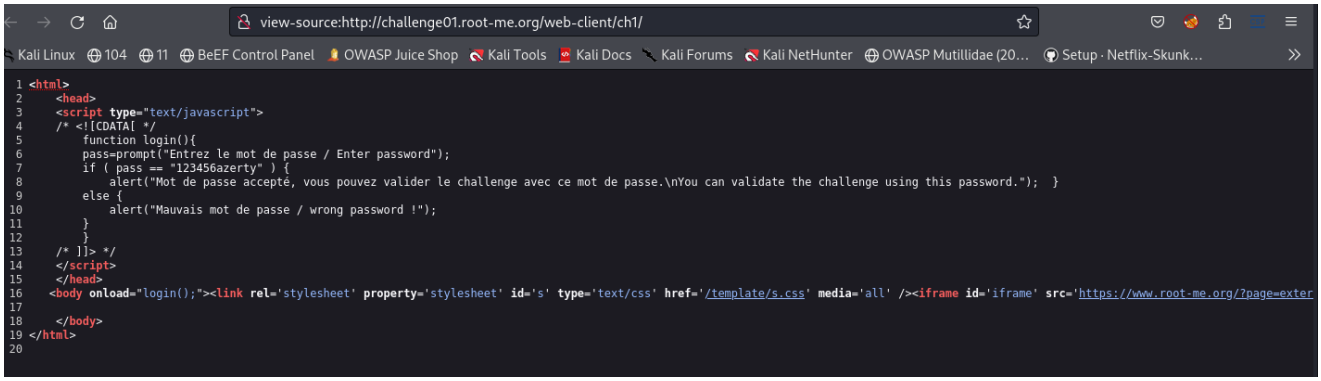
Задание_4:

Выполните задание https://www.root-me.org/en/Challenges/Web-Client/Javascript-Source из раздела client side на root-me.org и сдайте флаг, чтобы познакомиться с интерфейсом сайта.

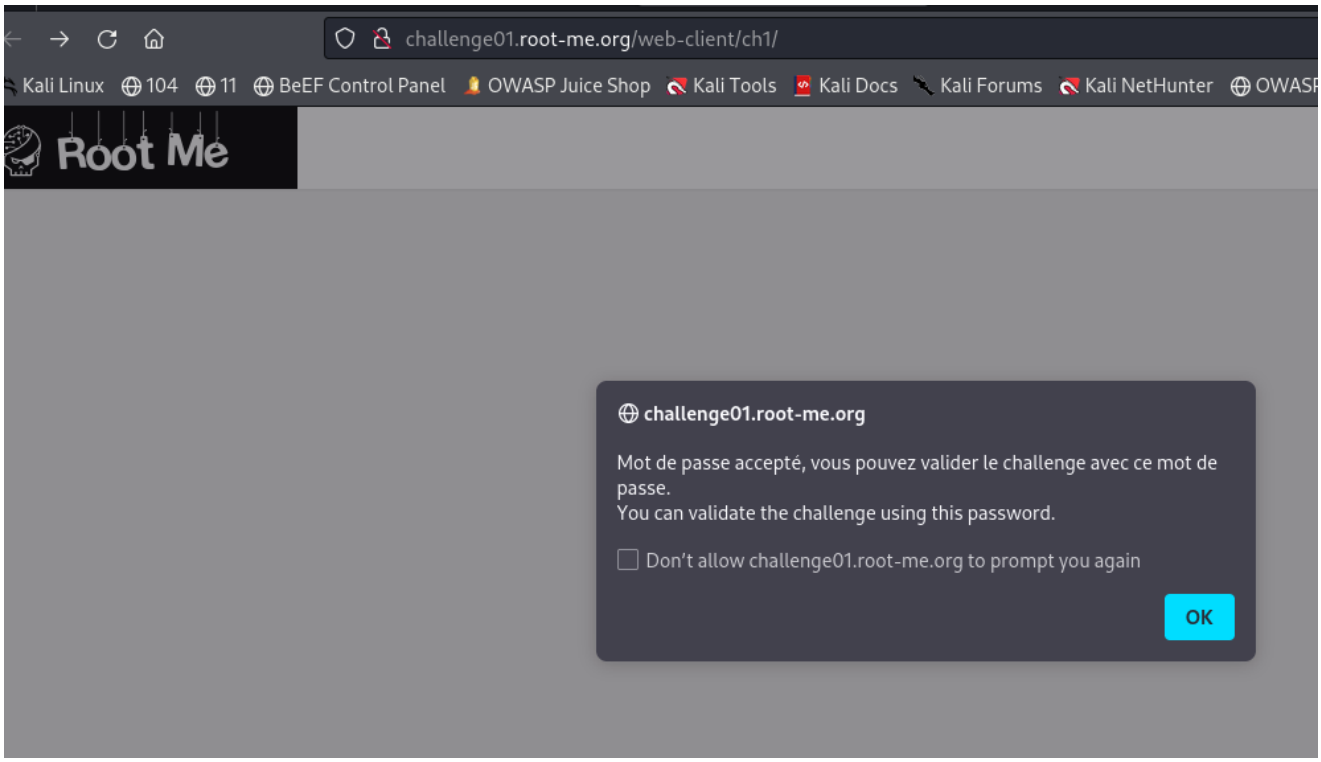




Смотрим код страницы и видим в *if*

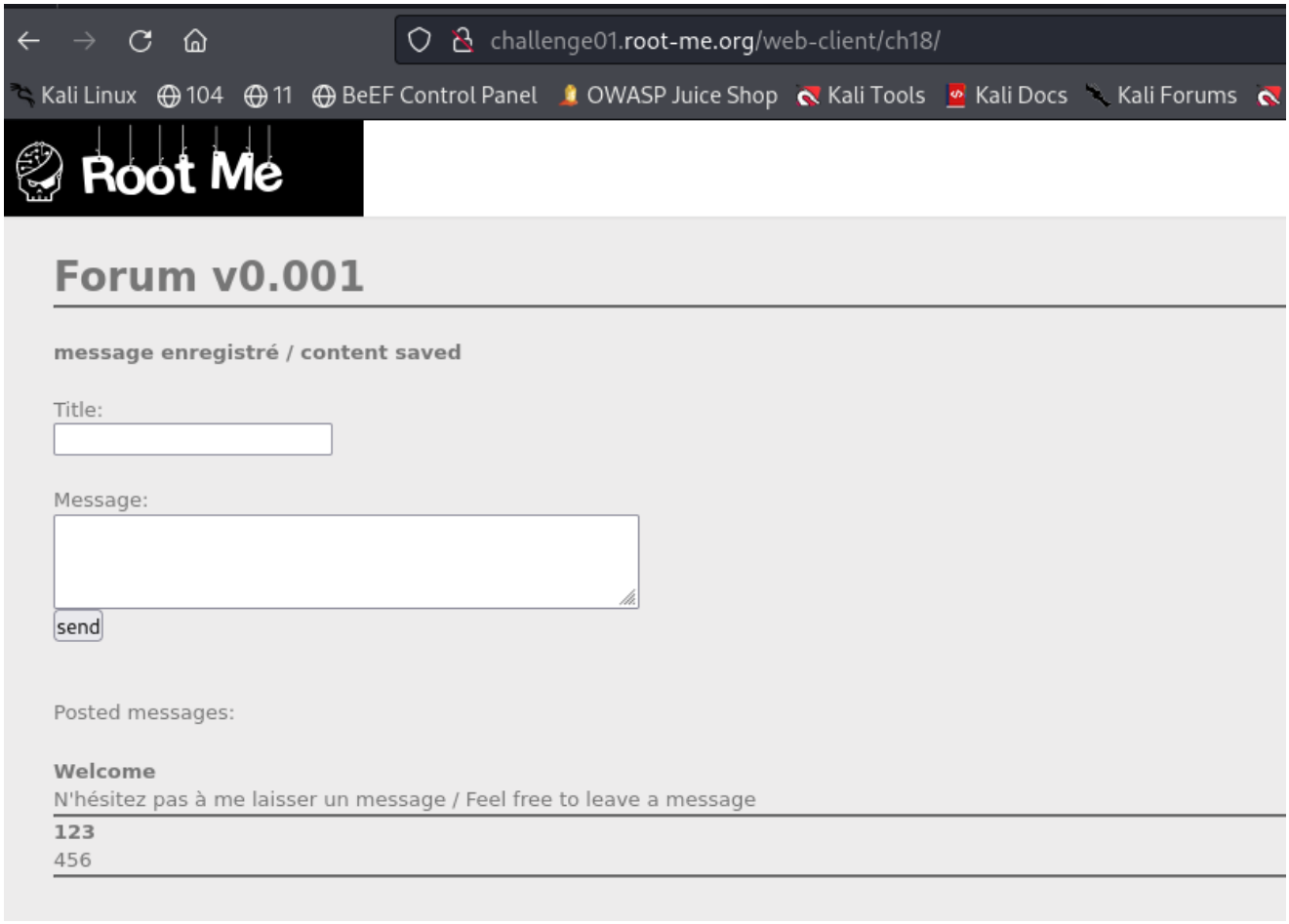


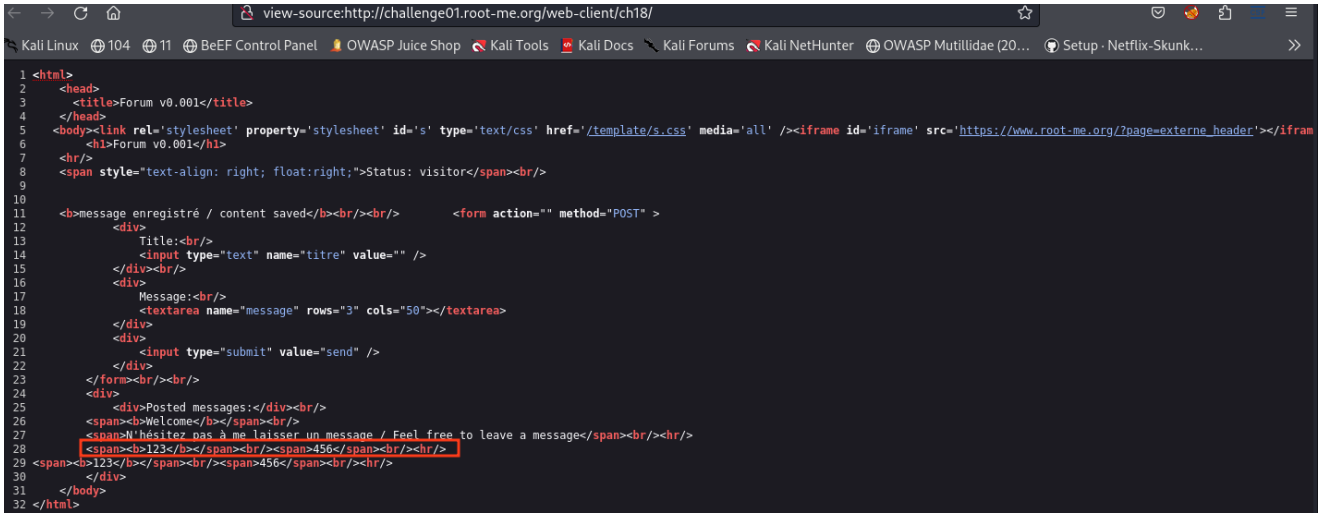
ПАРОЛЬ: 123456azerty



Задание_5:

(*) Выполните задания https://www.root-me.org/en/Challenges/Web-Client/XSS-Stored-1 и https://www.root-me.org/en/Challenges/Web-Client/XSS-Stored-2.


challenge01.root-me.org/web-client/ch18/
Kali Linux
104
11
BeEF Control Panel
OWASP Juice Shop
Kali Tools
Kali Docs
Kali Forums
Root Me
Forum v0.001
message enregistré / content saved
Title:
Message:
send
Posted messages:
Welcome
N'hésitez pas à me laisser un message / Feel free to leave a message
123
456


view-source:http://challenge01.root-me.org/web-client/ch18/
Kali Linux
104
11
BeEF Control Panel
OWASP Juice Shop
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
OWASP Mutillidae (20...
Setup · Netflix-Skunk...
<html>
<head>
<title>Forum v0.001</title>
</head>
<body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css' media='all' /><iframe id='iframe' src='https://www.root-me.org/?page=externe_header'></ifram
<h1>Forum v0.001</h1>
<hr/>
Status: visitor
message enregistré / content saved

 <form action="" method="POST" >

Title:
<input type="text" name="titre" value="" />

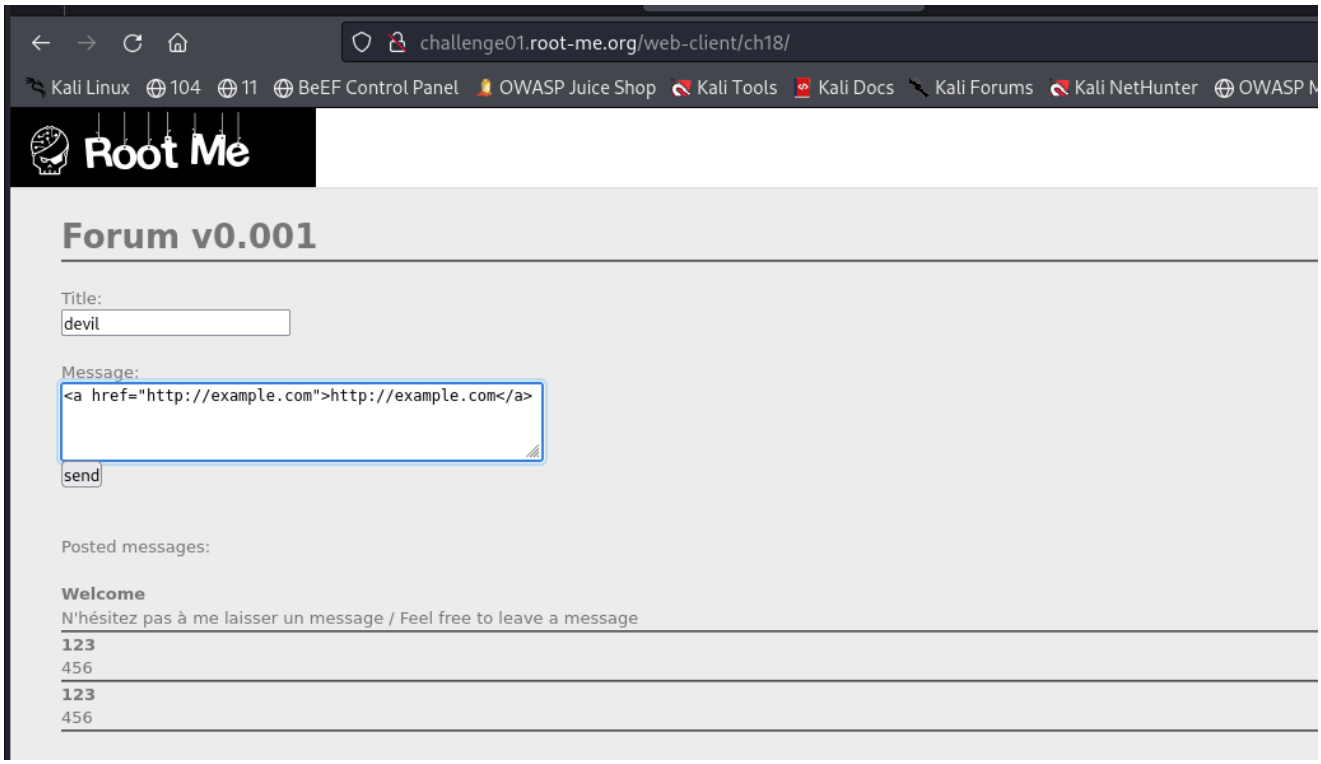
Message:
<textarea name="message" rows="3" cols="50"></textarea>

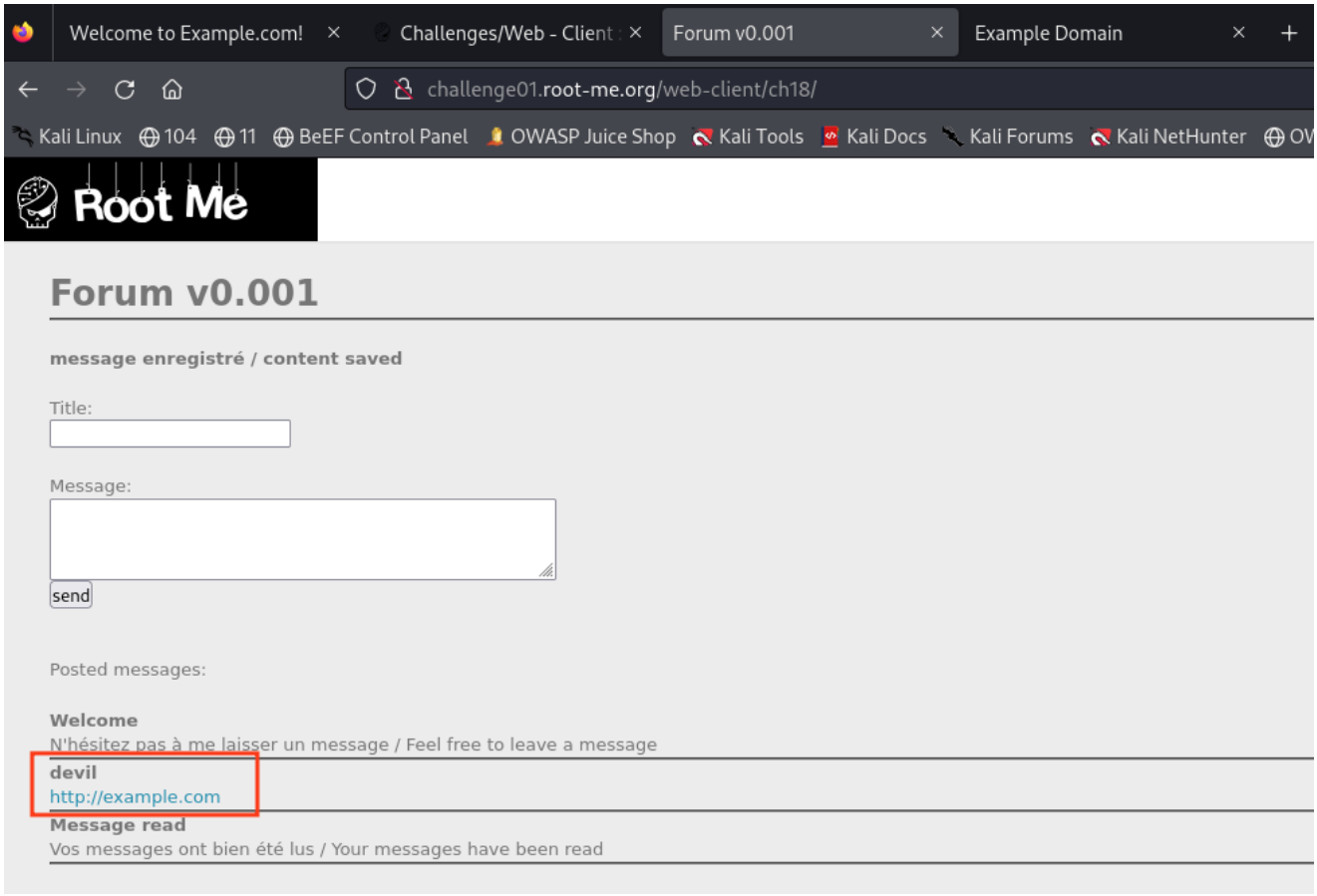
<input type="submit" value="send" />

</form>

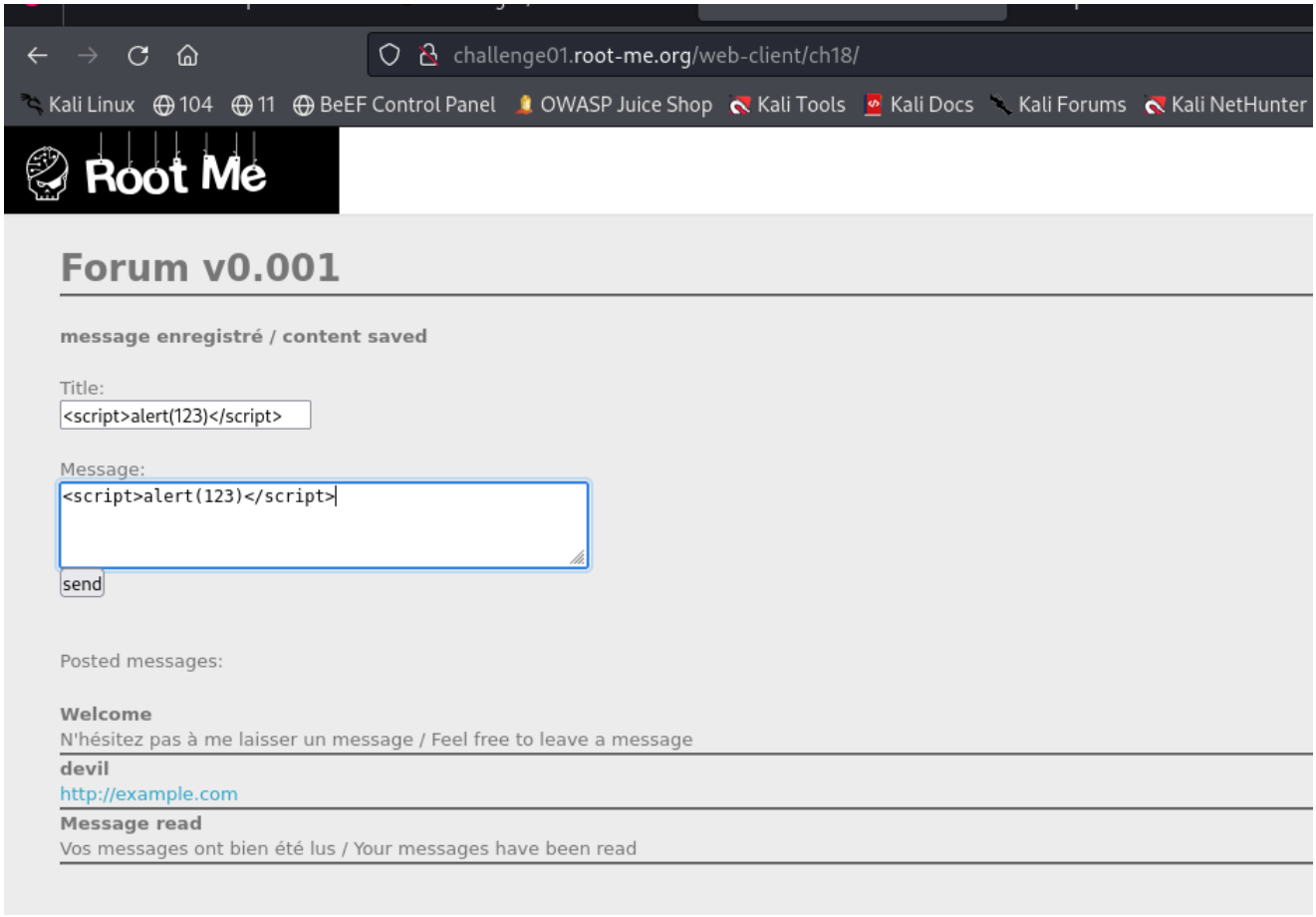
Posted messages:
Welcome
N'hésitez pas à me laisser un message / Feel free to leave a message
<hr/>
123
456
<hr/>
123
456
<hr/>

</body>
</html>





Можно сделать отправку на компрометирующий сайт.
Скрипты теги не пропускаются.



Выводы:

Мы рассмотрели достаточно тривиальные атаки, которые тем не менее все еще встречаются. Причинами тому могут быть:

- Наличие ошибок в коде виджетов, которые администраторы сайта не в состоянии убрать самостоятельно.
- Наличие ошибок в модулях некоторых фреймворков и версиях ПО (например, уязвимости в PHP).
- Отсутствие проверок данных, которые вводит пользователь.
  При построении защиты от рассматриваемых атак важно понимать, что защита должна носить комплексный характер.

Ссылки / дополнительные материалы

https://habr.com/post/186616 – полезная статья с демонстрацией Clickjacking.
http://n1cesecurity.blogspot.com/2015/10/html-injection-bwapp.html – описание HTML-инъекций на примере bWAPP.
https://stackoverflow.com/questions/10687099/how-to-test-if-a-url-string-is-absolute-or-relative/10687158 – как проверить при помощи JS, абсолютная ссылка или нет.

1. https://learn.javascript.ru/clickjacking.
2. https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet.
3. https://www.owasp.org/index.php/Testing_for_Clickjacking_(OTG-CLIENT-009).
4. https://habr.com/post/186616/.
5. https://www.acunetix.com/vulnerabilities/web/html-injection.

6. https://www.owasp.org/index.php/Testing_for_HTML_Injection_(OTG-CLIENT-003).
7. http://www.hackingarticles.in/beginner-guide-html-injection/.
8. https://www.owasp.org/index.php/Unvalidated_Redirects_and_Forwards_Cheat_Sheet.
9. https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/XSS%20injection.

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

- https://gb.ru

Выполнил: AndreiM