

Урок 3. HTTP

Оглавление

1. Создать файл test.txt в корневом каталоге сервера. Получить этот файл через браузер.	2
2. Создать на сервере файл sensitive_info.txt. Добавить базовую HTTP авторизацию для этого файла.	4
3. Открыть инструменты разработчика, вкладку Сеть (Network). Зайти на сайт https://geekbrains.ru . Проанализировать куки каждого запроса за HTML и картинками. Какие запросы уходят с куками, а какие без кук? Почему в каждом из случаев происходит именно такое поведение?.....	6
4. * Для выполнения этого задания вам потребуется:.....	6
5. (*) Сгенерировать самоподписанный сертификат и разместить его на своем сервере.	7

1. Создать файл test.txt в корневом каталоге сервера. Получить этот файл через браузер.

Установить в терминале программу curl, получить тот же файл с помощью этой программы.

Установить telnet или netcat, получить тот же файл с помощью одной из этих программ.

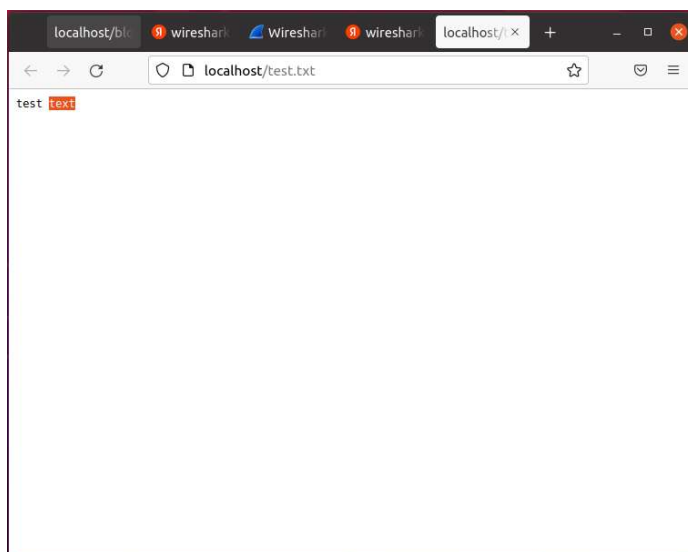
1.1. Создадим файл test.txt, выполним команды

```
cd /usr/share/nginx/html/
```

```
sudo nano test.txt
```

Введем произвольный текст «test text»

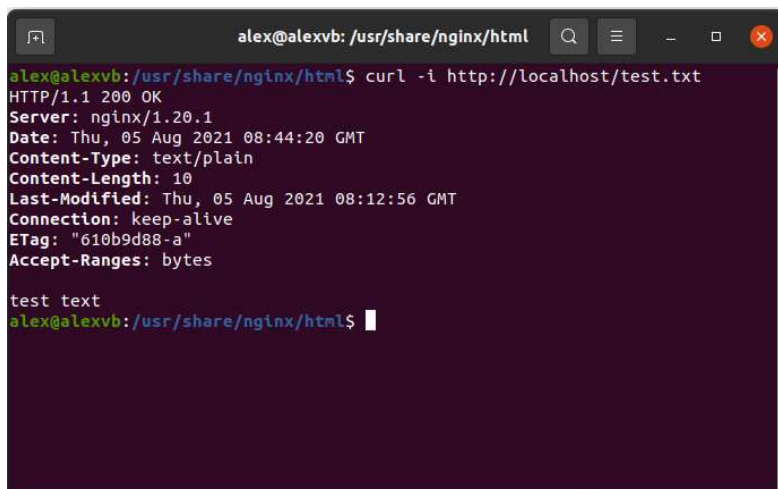
1.2. Откроем файл test.txt через браузер



1.3. Установим утилиту curl и откроем файл test.txt, выполним команду

```
sudo apt install curl
```

```
curl -i http://localhost/test.txt
```



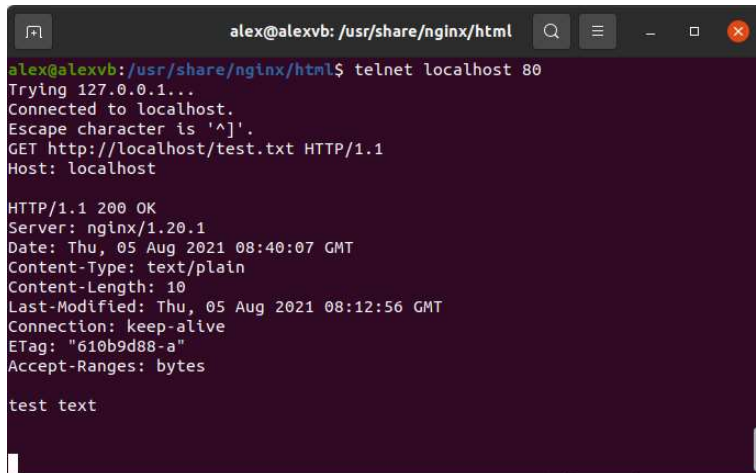
1.4. Откроем файл test.txt утилитой telnet, выполним команду

telnet localhost 80

Передадим в запросе параметры

GET http://localhost/test.txt HTTP/1.1

Host: localhost

A terminal window titled 'alex@alexvb: /usr/share/nginx/html' showing a telnet session. The user enters 'telnet localhost 80', and the terminal shows 'Trying 127.0.0.1...', 'Connected to localhost.', and 'Escape character is '^[''. Then, the user enters 'GET http://localhost/test.txt HTTP/1.1' and 'Host: localhost'. The server responds with 'HTTP/1.1 200 OK', 'Server: nginx/1.20.1', 'Date: Thu, 05 Aug 2021 08:40:07 GMT', 'Content-Type: text/plain', 'Content-Length: 10', 'Last-Modified: Thu, 05 Aug 2021 08:12:56 GMT', 'Connection: keep-alive', 'ETag: "610b9d88-a"', and 'Accept-Ranges: bytes'. Finally, the body of the response is 'test text'.

```
alex@alexvb: /usr/share/nginx/html$ telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.
GET http://localhost/test.txt HTTP/1.1
Host: localhost

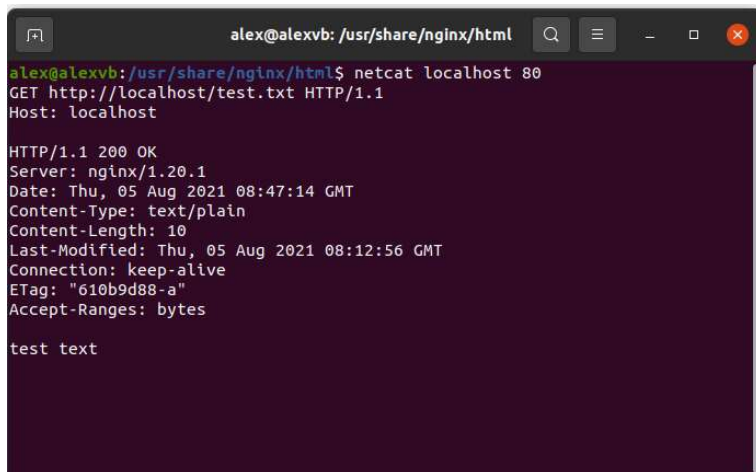
HTTP/1.1 200 OK
Server: nginx/1.20.1
Date: Thu, 05 Aug 2021 08:40:07 GMT
Content-Type: text/plain
Content-Length: 10
Last-Modified: Thu, 05 Aug 2021 08:12:56 GMT
Connection: keep-alive
ETag: "610b9d88-a"
Accept-Ranges: bytes

test text
```

1.5. Откроем файл test.txt утилитой netcat, выполним команду

netcat localhost 80

Передадим параметры как в п. 1.4.

A terminal window titled 'alex@alexvb: /usr/share/nginx/html' showing a netcat session. The user enters 'netcat localhost 80'. Then, the user enters 'GET http://localhost/test.txt HTTP/1.1' and 'Host: localhost'. The server responds with 'HTTP/1.1 200 OK', 'Server: nginx/1.20.1', 'Date: Thu, 05 Aug 2021 08:47:14 GMT', 'Content-Type: text/plain', 'Content-Length: 10', 'Last-Modified: Thu, 05 Aug 2021 08:12:56 GMT', 'Connection: keep-alive', 'ETag: "610b9d88-a"', and 'Accept-Ranges: bytes'. Finally, the body of the response is 'test text'.

```
alex@alexvb: /usr/share/nginx/html$ netcat localhost 80
GET http://localhost/test.txt HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
Server: nginx/1.20.1
Date: Thu, 05 Aug 2021 08:47:14 GMT
Content-Type: text/plain
Content-Length: 10
Last-Modified: Thu, 05 Aug 2021 08:12:56 GMT
Connection: keep-alive
ETag: "610b9d88-a"
Accept-Ranges: bytes

test text
```

2. Создать на сервере файл sensitive_info.txt. Добавить базовую HTTP авторизацию для этого файла.

Получить этот файл через браузер.

Получить тот же файл с помощью curl и telnet или netcat.

2.1. Создадим файл sensitive_info.txt, выполним команды

```
cd /usr/share/nginx/html/
```

```
sudo mkdir topsicret
```

```
sudo nano sensitive_info.txt
```

Введем произвольный текст «sensitive info text»

2.2. Установим htpasswd для генерации шифрованных паролей, выполним команду

```
sudo apt-get install apache2-utils
```

2.3. Создадим пользователя user1 и зададим ему пароль

```
sudo htpasswd -c /etc/nginx/.htpasswd user1
```

2.4. Настроим nginx, выполним команды

```
sudo nano /etc/nginx/conf.d/basic.conf
```

добавим в конфигурацию

```
location /topsicret {  
    auth_basic      "closed site";  
    auth_basic_user_file .htpasswd;  
}
```

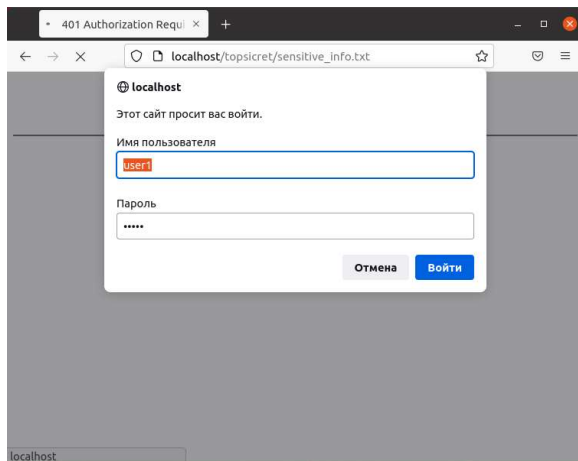
перезапустим nginx

```
systemctl stop nginx
```

```
systemctl start nginx
```

2.5. Проверим работу авторизации по паролю, откроем браузер и введём

http://localhost/topsecret/sensitive_info.txt



2.6. Получим файл с помощью curl и telnet или netcat.

```
alex@alexvb: /usr/share/nginx/html
alex@alexvb:/usr/share/nginx/html$ curl -i http://localhost/topsecret/sensitive_info.txt
HTTP/1.1 401 Unauthorized
Server: nginx/1.20.1
Date: Thu, 05 Aug 2021 10:28:04 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
WWW-Authenticate: Basic realm="closed site"

<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx/1.20.1</center>
</body>
</html>
alex@alexvb:/usr/share/nginx/html$
```

```
alex@alexvb: /usr/share/nginx/html
alex@alexvb:/usr/share/nginx/html$ telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET http://localhost/topsecret/sensitive_info.txt HTTP/1.1
Host: localhost

HTTP/1.1 401 Unauthorized
Server: nginx/1.20.1
Date: Thu, 05 Aug 2021 10:29:18 GMT
Content-Type: text/html
Content-Length: 179
Connection: keep-alive
WWW-Authenticate: Basic realm="closed site"

<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx/1.20.1</center>
</body>
</html>
```

```
alex@alexvb: /usr/share/nginx/html
alex@alexvb:/usr/share/nginx/html$ netcat localhost 80 < http://localhost/topsecret/sensitive_info.txt
bash: http://localhost/topsecret/sensitive_info.txt: Нет такого файла или каталога
alex@alexvb:/usr/share/nginx/html$
```

3. Открыть инструменты разработчика, вкладку Сеть (Network).

Зайти на сайт <https://geekbrains.ru>. Проанализировать куки каждого запроса за HTML и картинками. Какие запросы уходят с куками, а какие без кук? Почему в каждом из случаев происходит именно такое поведение?

Html запросы содержат куки. Для повышения быстродействия изображения хранятся на различных серверах (top-fwz1.mail.ru, cloudfront.net) куки отсутствуют, либо в них не содержатся данные связанные с безопасностью.

4. * Для выполнения этого задания вам потребуется:

Проведите исследование механизма проставления кук, для этого попробуйте установить следующие куки:

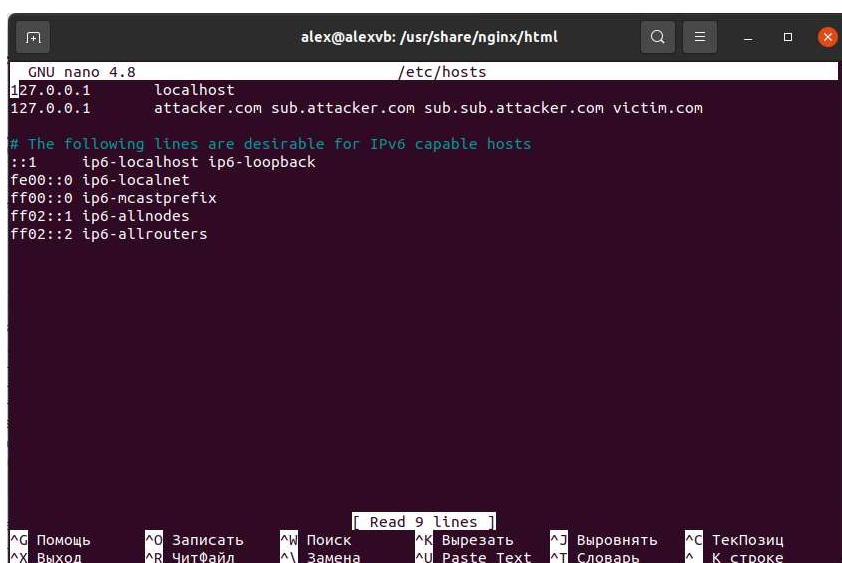
1. С домена attacker.com на домен sub.attacker.com
2. С домена attacker.com на домен victim.com
3. С домена sub.attacker.com на домен attacker.com
4. С домена sub.sub.attacker.com на домен attacker.com

По каждому пункту ответьте на вопросы:

1. Куда установились куки?
2. Если не установились, то почему?

Обобщите полученные знания и напишите вывод в формате: "Домен может проставлять куки для себя, для ... и ..., но не может проставлять куки для ..., ... и ...".

4.1. Редактируем /etc/hosts



```
alex@alexvb: /usr/share/nginx/html
GNU nano 4.8 /etc/hosts
127.0.0.1 localhost
127.0.0.1 attacker.com sub.attacker.com sub.sub.attacker.com victim.com

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

Read 9 lines
^G Помощь ^O Записать ^W Поиск ^K Вырезать ^J Выводить ^C ТекПозиц
^X Выход ^R ЧитФайл ^M Замена ^U Paste Text ^T Словарь ^_ К строке
```

4.2. Добавляем метаданные согласно заданию, выполним команды

```
sudo nano /etc/nginx/sites-available/cookie-research.conf
```

Копируем данные кук для доменов

4.3. С домена attacker.com на домен victim.com

не установятся куки, так как другой домен

4.4. С домена sub.attacker.com на домен attacker.com

установятся куки, так как поддомен может выставлять куки на родительский домен

4.5. С домена sub.sub.attacker.com на домен attacker.com

установятся куки, так как поддомен может выставлять куки на родительский домен

5. (*) Сгенерировать самоподписанный сертификат и разместить его на своем сервере.

5.1. Создаем корневой сертификат, выполним команду

```
cd /etc/nginx/
```

```
sudo certs
```

```
sudo openssl genrsa -out sert_local.key 2048
```

5.2. Генерируем публичный ключ, выполним команду

```
sudo openssl req -new -key sert_local.key -out sert_local.csr
```

вводим данные: страну город, наименование компании и другие

подписываем сертификат:

```
sudo openssl x509 -req -days 365 -in sert_local.csr -signkey sert_local.key -out sert_local.csr
```

5.3. Выполняем настройку nginx

```
cd /etc/nginx/conf.d/
```

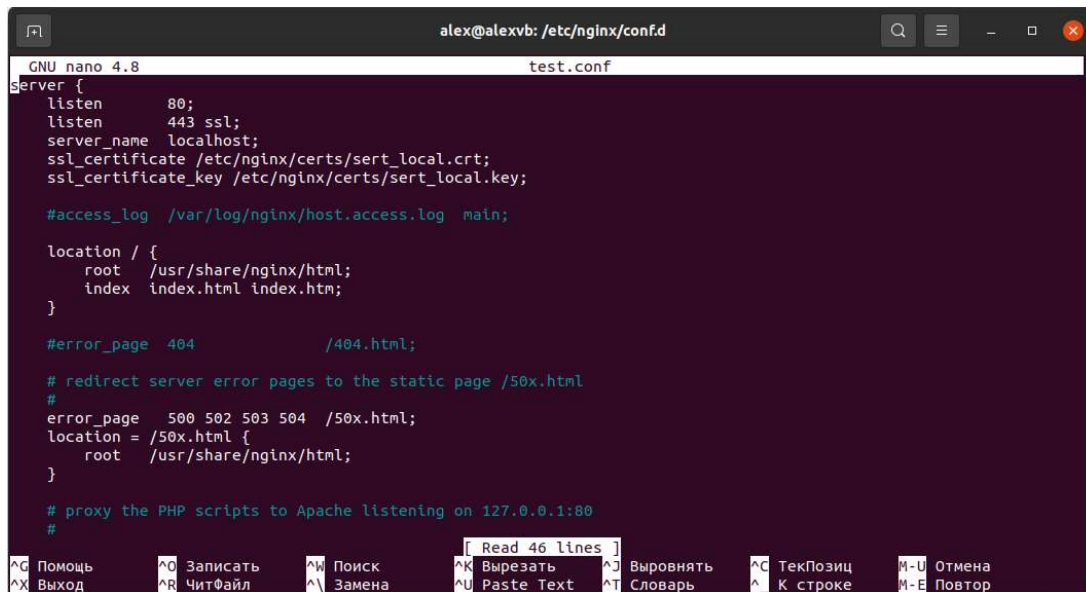
```
sudo nano test.conf
```

Добавим в файл test.conf строку

listen 443 ssl;

ssl_certificate /etc/nginx/certs/ sert_local.crt;

ssl_certificate_key /etc/nginx/certs/ sert_local.key;



```
alex@alexvb: /etc/nginx/conf.d
GNU nano 4.8 test.conf
server {
    listen      80;
    listen      443 ssl;
    server_name localhost;
    ssl_certificate /etc/nginx/certs/sert_local.crt;
    ssl_certificate_key /etc/nginx/certs/sert_local.key;

    #access_log /var/log/nginx/host.access.log main;

    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
    }

    #error_page 404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    [ Read 46 lines ]
^G Помощь  ^O Записать  ^M Поиск    ^K Вырезать  ^J Вывернуть  ^C ТекПозиц  ^M-U Отмена
^X Выход   ^R ЧитФайл   ^_ Замена  ^U Paste Text ^T Словарь   ^_ К строке  ^M-E Повтор
```

5.4. Применим новую конфигурацию nginx, убедимся что сервис поднялся без ошибок

sudo systemctl reload nginx

sudo systemctl status nginx

5.5. Проверим работу по https и проверим сертификат, откроем браузер и введём

<https://localhost>

Сертификат для alexlocalh xWelcome to nginx! x +

Firefoxabout:certificate?cert=MIIDnTCCAoUCFFRdX9JSMsj0YfcCIPjzVvjNVgfiMA0GCSqGSib3DQEBCwUAMIG ☆

Сертификат

alexlocalhost

Субъект

Страна	RU
Область/Регион	Moscow
Населённый пункт	Moscow
Организация	private
Подразделение	private
Общее имя	alexlocalhost
Адрес электронной почты	alex@gmail.com

Издатель

Страна	RU
Область/Регион	Moscow
Населённый пункт	Moscow
Организация	private
Подразделение	private
Общее имя	alexlocalhost
Адрес электронной почты	alex@gmail.com