

Выполнил Мешечкин Д. Инфобез-2345

### Задание ДЗ№8:

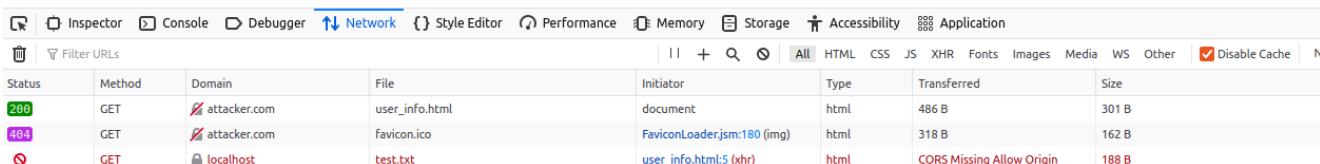
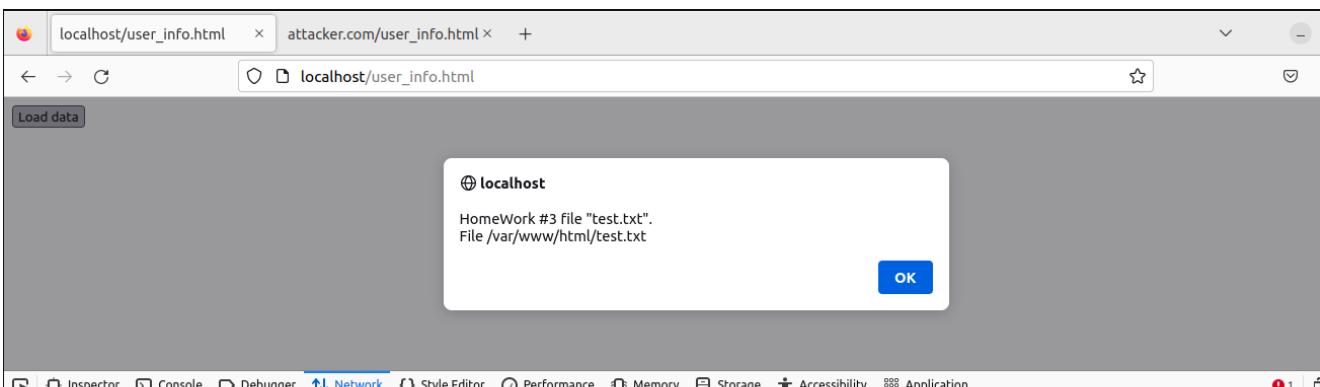
- Перед выполнением задания необходимо:
- Создать страницу user\_info.html на домене localhost
- Добавить на домене localhost заголовок CORS: Access-Control-Allow-Origin: \* На домене attacker.com создать страницу, которая:
- Выполнит XHR запрос за страницей localhost/user\_info.html, выведет содержимое страницы user\_info.html
- Настройте CORS так, чтобы вывести содержимое страницы user\_info.html мог только http://localhost или http://trustedhost.com.

Файл с методички к ДЗ№7 l-7-2.html переведу в необходимый user\_info.html

```
GNU nano 6.2                                     user_info.html
<script>
function xhrTest() {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "http://localhost/test.txt", false);
    xhr.send();

    if (xhr.status != 200) {
        alert (xhr.status + ': ' + xhr.statusText);
    } else {
        alert (xhr.responseText);
    }
}
</script>
<button onclick="xhrTest()">Load data</button>
```

Далее этот файл пробовал загрузить localhost и attacker.com.



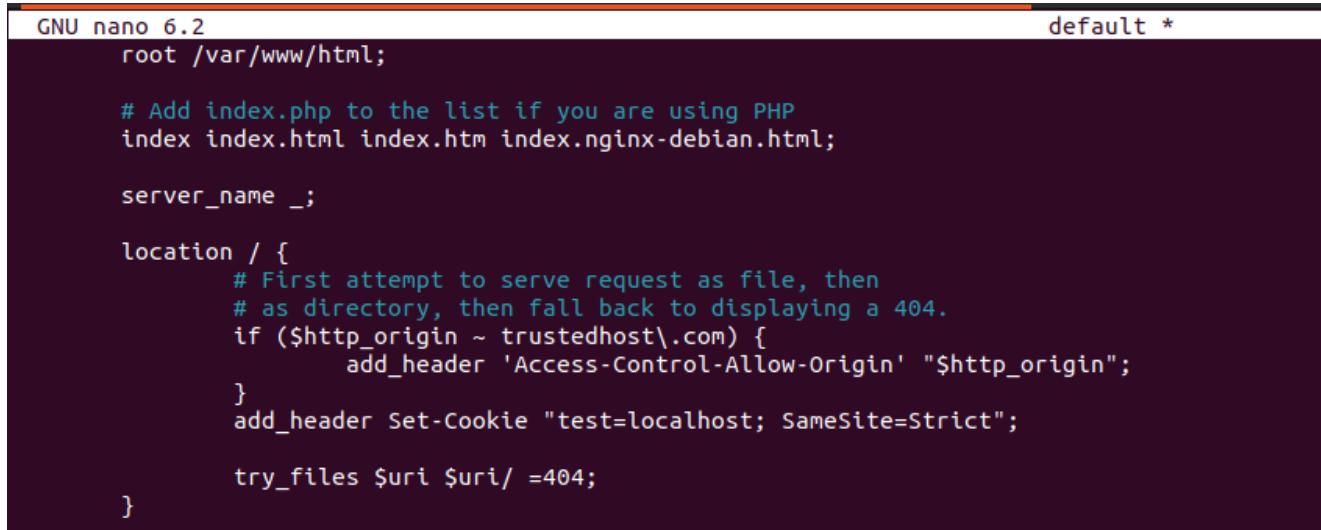
В данной ситуации XHR запрос не выполняется из-за наполнения файла user\_info.html, в котором присутствует строка  
`xhr.open("GET", "http://localhost/test.txt", false);`

не позволяющая выполнять запрос с других доменов.

Одновременно с этим в файл /etc/hosts добавлю строку, присвоив trustedhost.com IP-адрес localhost(127.0.0.1).

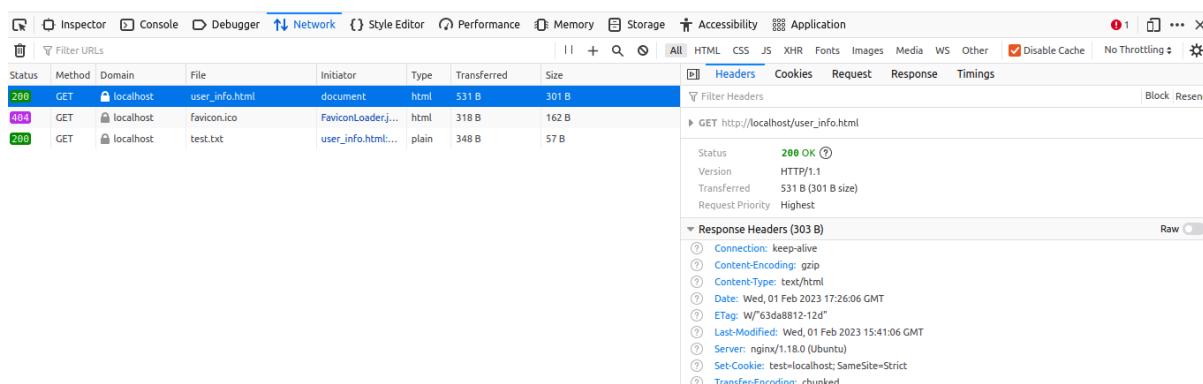
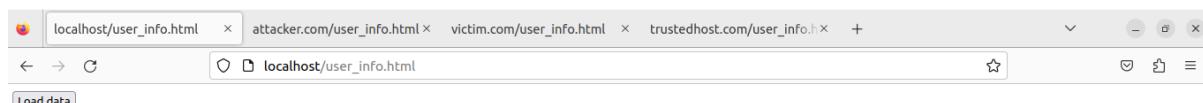
Для выполнения задания необходимо дополнить /etc/nginx/sites-available/default заголовками:

```
if ($http_origin ~ trustedhost\.com) {  
    add_header 'Access-Control-Allow-Origin' "$http_origin";  
}  
add_header Set-Cookie "test=localhost; SameSite=Strict";
```



```
GNU nano 6.2                               default *  
root /var/www/html;  
  
# Add index.php to the list if you are using PHP  
index index.html index.htm index.nginx-debian.html;  
  
server_name _;  
  
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    if ($http_origin ~ trustedhost\.com) {  
        add_header 'Access-Control-Allow-Origin' "$http_origin";  
    }  
    add_header Set-Cookie "test=localhost; SameSite=Strict";  
  
    try_files $uri $uri/ =404;  
}
```

В данной ситуации заголовок origin будет сравниваться с доверенным, в данном случае с доменом trustedhost.com, присваивая ему уже заголовок origin, тем самым позволяя получать данные с заголовком Access-Control-Allow-Origin. Сохраняя доступ с localhost, и закрывая доступ XHR запросам с victim.com/attacker.com. И устанавливая заголовок Set-Cookie с параметрами test=localhost; SameSite=Strict.



Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost	user_info.html	document	html	531 B	301 B
404	GET	localhost	favicon.ico	FaviconLoader.j... FaviconLoader.j...	html	318 B	162 B
200	GET	localhost	test.txt	user_info.html...	plain	348 B	57 B

Headers

Status	200 OK
Version	HTTP/1.1
Transferred	531 B (301 B size)
Request Priority	Highest

Response Headers (303 B)

Raw
Connection: keep-alive Content-Encoding: gzip Content-Type: text/html Date: Wed, 01 Feb 2023 17:26:06 GMT ETag: W/"6368812-12d" Last-Modified: Wed, 01 Feb 2023 15:41:06 GMT Server: nginx/1.18.0 (Ubuntu) Set-Cookie: test=localhost; SameSite=Strict Transfer-Encoding: chunked

The screenshots show a browser interface with Network and Application tabs open, displaying network traffic and cookie information across four tabs:

- Tab 1: attacker.com/user\_info.html**
- Tab 2: victim.com/user\_info.html**
- Tab 3: trustedhost.com/user\_info.html**
- Tab 4: localhost/test.txt**

**Network Tab Details:**

- localhost/test.txt:** Status 200 OK, Response Headers include Content-Type: text/plain, Date: Wed, 01 Feb 2023 18:35:21 GMT, ETag: "63c2c2f4-39", Last-Modified: Sat, 14 Jan 2023 14:57:56 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.
- victim.com/user\_info.html:** Status 200 OK, Response Headers include Content-Type: text/html, Date: Wed, 01 Feb 2023 17:26:27 GMT, ETag: "63c2c2f4-39", Last-Modified: Sat, 14 Jan 2023 14:57:56 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.
- trustedhost.com/user\_info.html:** Status 200 OK, Response Headers include Content-Type: text/html, Date: Wed, 01 Feb 2023 19:21:52 GMT, ETag: "63d48812-12d", Last-Modified: Wed, 01 Feb 2023 15:41:06 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.
- attacker.com/user\_info.html:** Status 200 OK, Response Headers include Content-Type: text/html, Date: Wed, 01 Feb 2023 18:35:21 GMT, ETag: "63c2c2f4-39", Last-Modified: Sat, 14 Jan 2023 14:57:56 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.

**Application Tab Details:**

- localhost/test.txt:** Status 200 OK, Response Headers include Content-Type: text/plain, Date: Wed, 01 Feb 2023 18:35:21 GMT, ETag: "63c2c2f4-39", Last-Modified: Sat, 14 Jan 2023 14:57:56 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.
- victim.com/user\_info.html:** Status 200 OK, Response Headers include Content-Type: text/html, Date: Wed, 01 Feb 2023 17:26:27 GMT, ETag: "63c2c2f4-39", Last-Modified: Sat, 14 Jan 2023 14:57:56 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.
- trustedhost.com/user\_info.html:** Status 200 OK, Response Headers include Content-Type: text/html, Date: Wed, 01 Feb 2023 19:21:52 GMT, ETag: "63d48812-12d", Last-Modified: Wed, 01 Feb 2023 15:41:06 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.
- attacker.com/user\_info.html:** Status 200 OK, Response Headers include Content-Type: text/html, Date: Wed, 01 Feb 2023 18:35:21 GMT, ETag: "63c2c2f4-39", Last-Modified: Sat, 14 Jan 2023 14:57:56 GMT, Server: nginx/1.18.0 (Ubuntu), Set-Cookie: test=localhost; SameSite=Strict.

На последнем скриншоте открыт trustedhost.com/user\_info.html. Присвоен заголовок Set-Cookie, на пробном этапе создания отчёта ДЗ, был заголовок "Access-Control-Allow-Origin", далее ряд "весёлых злоключений" с отключением света, перезагрузкой системы, отвала диска, ошибок загрузки суперблока. Результат - "нет прогруза в адекватной мере достигнутых результатов". Дополнительных попыток перезагрузки не предпринимал для сохранения достигнутых результатов.

Задание №6-8:

- Вы - злоумышленник, поэтому в Firefox вы заходите только через приватное окно. Вы хотите украсть супер секретные данные со страницы http://victim.com/hw-8-2.php. На ней установлена защита по сессии. Но вы знаете пользователя у которого эта сессия есть и что секрет отдается postMessage после открытия страницы...
  - Заманите пользователя на страницу http://attacker.com/hw-8-2-attacker.html и получите секретные данные.
  - Допишите страницу http://victim.com/hw-8-2.php, так чтобы она была безопасной. Страница hw-8-2.php
- ```
<?php if($_COOKIE['sessionid'] ==  
'0a7016d5f7346a6f14b273a66e0770fb7d6608769f233156570e878a1397a175')  
{ echo "  
    <body>Hello, sir! Sending data to window.opener!  
        <script> window.opener.postMessage('TOP secret data', '*'); </script>  
    </body>";  
} else { echo "Access denied"; } ?>
```







