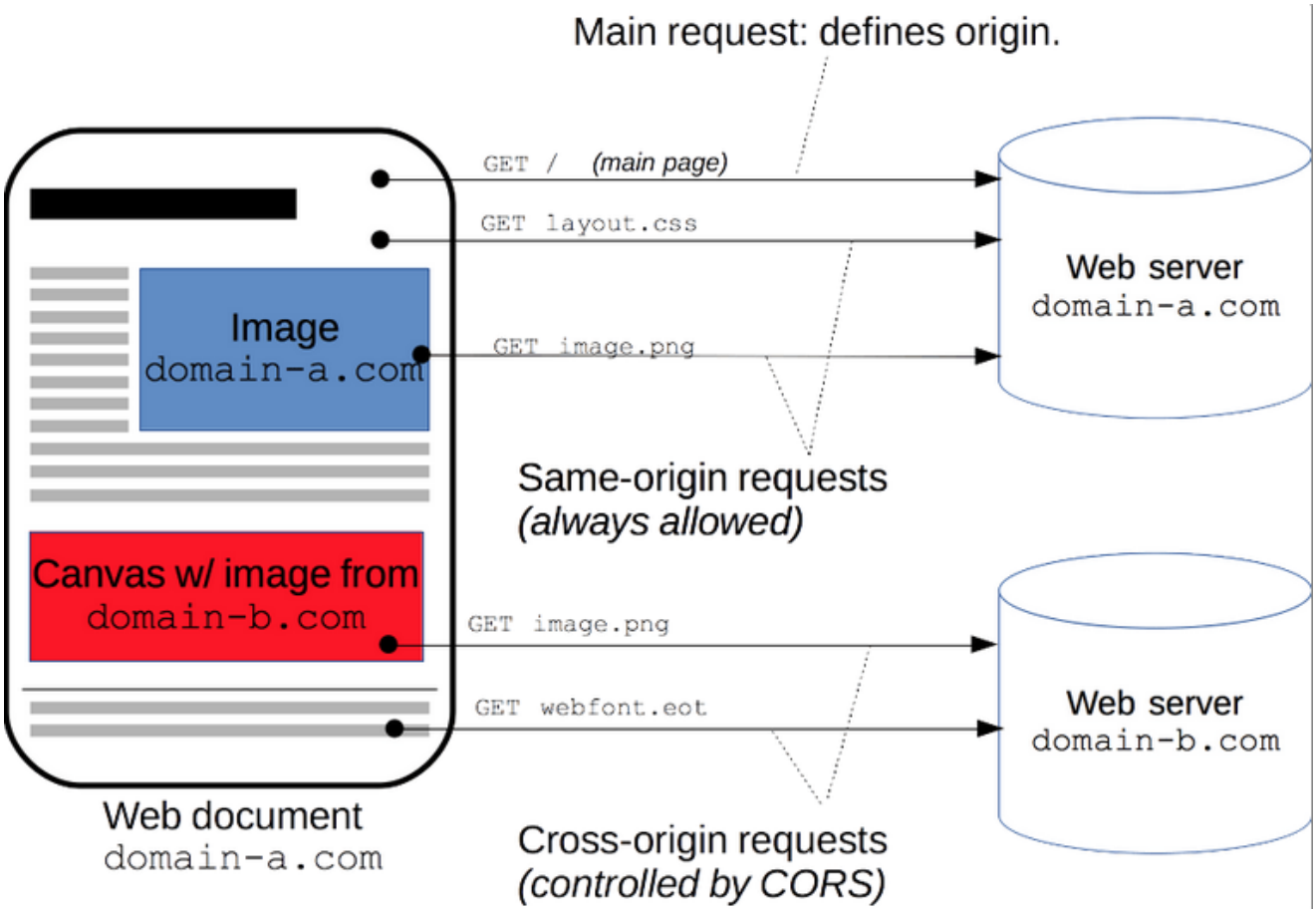
Урок 7

Same Origin Policy

Заметки

- Cookie - небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя. Веб-клиент (веб-браузер ...) всякий раз при попытке открыть страницу соответствующего сайта пересылает этот фрагмент данных веб-серверу в составе HTTP-запроса. Применяется для сохранения данных на стороне пользователя, на практике обычно используется для:
  - аутентификации пользователя;
  - хранения персональных предпочтений и настроек пользователя;
  - отслеживания состояния сеанса доступа пользователя;
  - хранения сведений статистики о пользователях.
- SOP (Same Origin Policy) для Cookie заключается в том что, например, домен a.com не может поставить Cookie на домен b.com, потому что у них разные origin. Но с Cookie не все просто, так например, дочерний домен может поставлять их на родительский.
- Cross-Origin Resource Sharing (CORS) — механизм, использующий дополнительные HTTP-заголовки, чтобы дать возможность агенту пользователя получать разрешения на доступ к выбранным ресурсам с сервера на источнике (домене), отличном от того, что сайт использует в данный момент. Говорят, что агент пользователя делает запрос с другого источника **(cross-origin HTTP request),** если источник текущего документа отличается от запрашиваемого ресурса доменом, протоколом или портом.
- В целях безопасности браузеры ограничивают cross-origin запросы, инициируемые скриптами. Например, `XMLHttpRequest` и Fetch API следуют *политике одного источника* (same-origin policy). Это значит, что web-приложения, использующие такие API, могут запрашивать HTTP-ресурсы только с того домена, с которого были загружены, пока не будут использованы CORS-заголовки.



- Примеры из скрипта:

```
cd /var/www/html && sudo nano l-7-3.html
<script>
        localStorage.setItem("userinfo", "Pavel Statsenko, geekbrains
...");
</script>
```

```
cd /var/www/html && sudo nano l-7-3.html
<script>
        localStorage.setItem("userinfo", "Pavel Statsenko, geekbrains
...");
        alert(localStorage.getItem("userinfo"));
        localStorage.removeItem("userinfo");
</script>
```

```
cd /var/www/html && sudo nano l-7-3.html
<script>
        localStorage.setItem("userinfo", "Pavel Statsenko, geekbrains
...");
        sessionStorage.setItem("session", "test");
</script>
```

Nginx & bWAAP:

- `cd /etc/nginx/sites-enabled && sudo nano default`
- `sudo nginx -s reload`
- `sudo nginx -t`
- `ps aux | grep ngninx`
- `/etc/nginx/sites-available`
- `sudo ln -s /etc/nginx/sites-available/default default
- `unzip bWAPP.zip`
- `mv bWAPP /var/www/html`
- `sudo apt install mysql-server`
- `sudo mysql`
- `sudo mysql -u bug -p`
  - `CREATE USER 'bug'@'localhost' IDENTIFIED BY 'bug';` `
  - `mysql -u bug -p`
  - `'bug'@'localhost' IDENTIFIED BY 'bug';`
- `/var/www/html/bWAPP/admin/settings.php`
  - `$db_server = "localhost";`
  - `$db_server = "bug";`
  - `$db_server = "bug";`
  - `$db_name = "bWAPP";`
  - `sudo chmod 777 passwords/`
  - `sudo chmod 777 images/`
  - `sudo chmod 777 documents/`
  - `sudo chmod 777 logs/`
- `localhost/bWAPP/install.php ...YES ...Login ...Portal`

Задание

1. Это задание выполняется на домене `attacker.com`. Прочитать куки домена `attacker.com` и вывести их. Попробовать прочитать и вывести куки домена `victim.com`.

В конфиге добавляем строки для куки:

- `server_name attacker.com;`
- `attacker.com (test1)`
- `victim.com (test2)`

```
GNU nano 6.2                          /etc/nginx/sites-enabled/default *
        #server_name _;
        #server_name localhost;
        server_name attacker.com;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
        # add security
                # add_header Content-Security-Policy "script-src none"
        # add Cokiess
        add_header 'Set-Cookie' 'test1=attacker-to-attacker; Max-age=3600; Domain=attacker.com';
        add_header 'Set-Cookie' 'test2=attacker-to-victim; Max-age=3600; Domain=victim.com';

                try_files $uri $uri/ =404;
        }

        # pass PHP scripts to FastCGI server
```

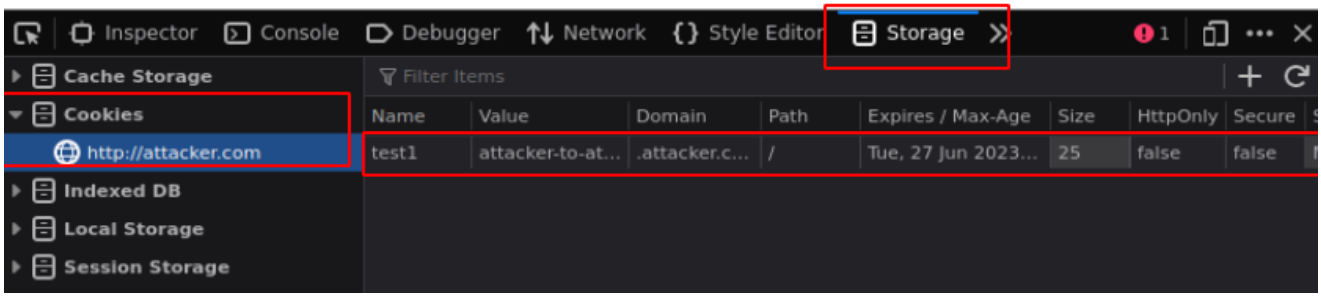
- `sudo nginx -s reload`

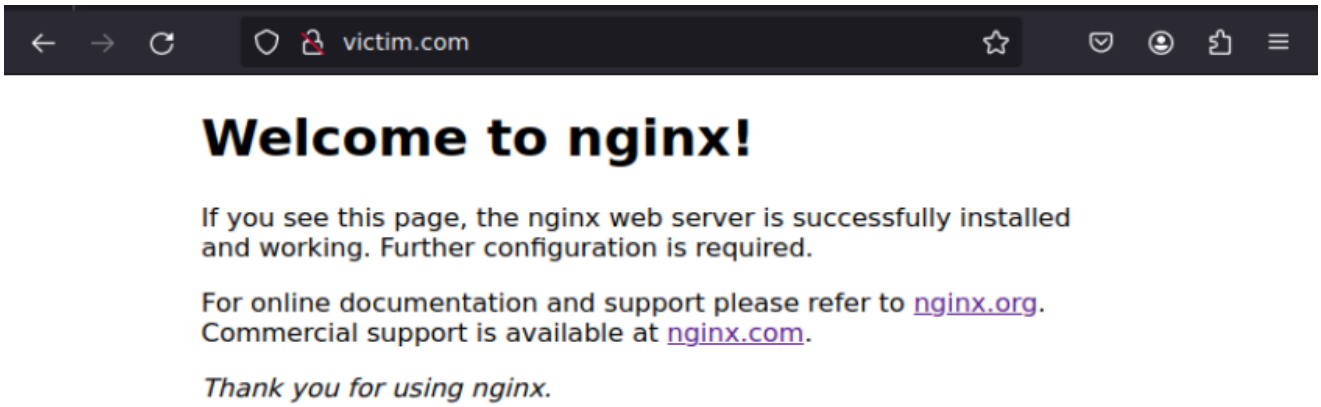
Welcome to nginx!

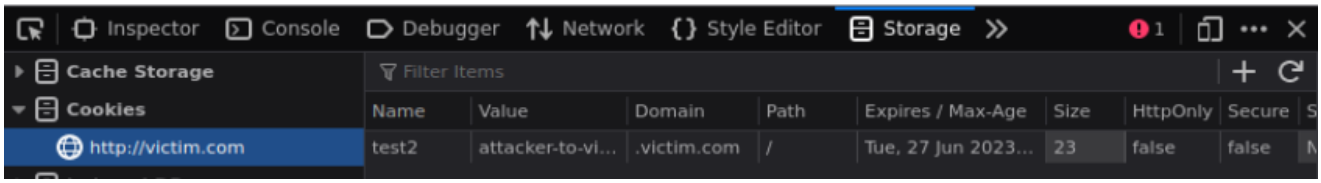
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



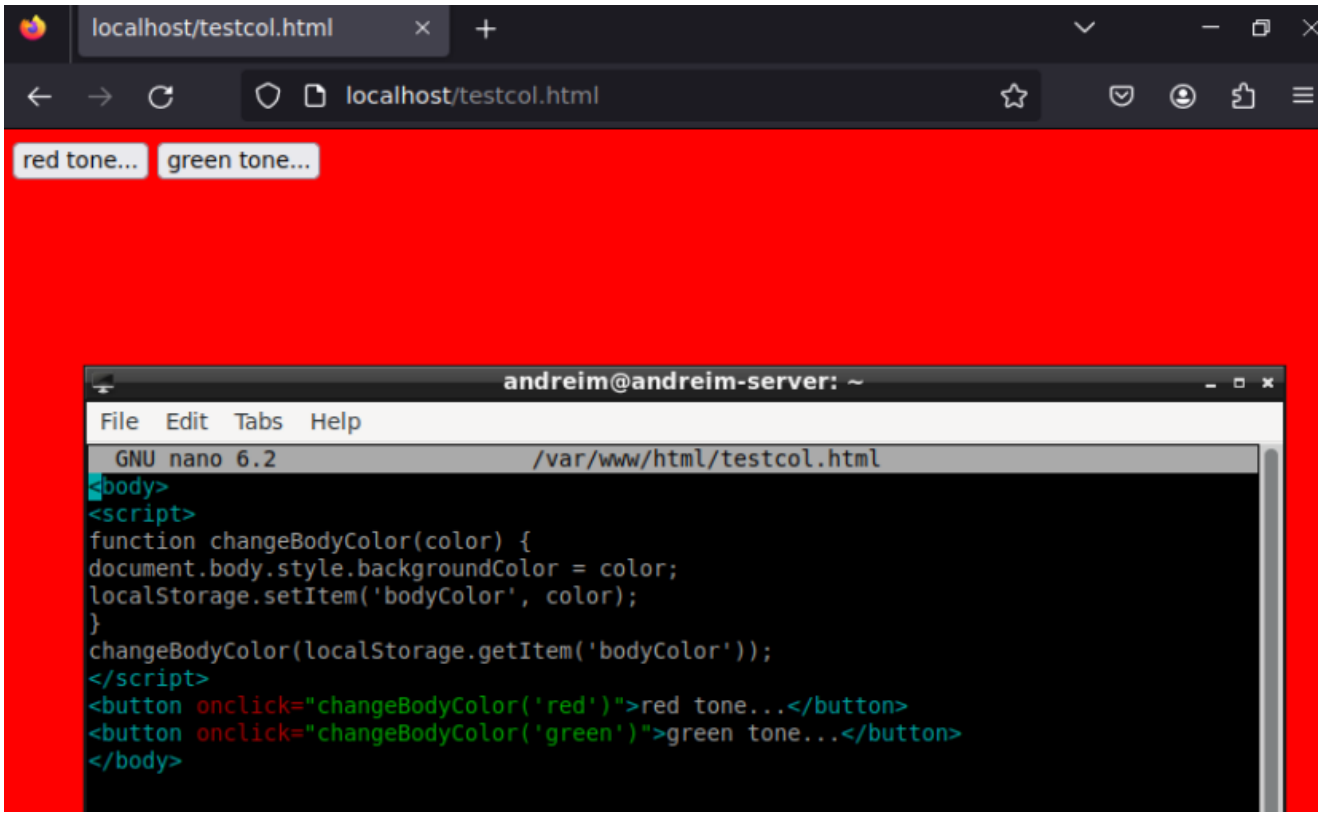


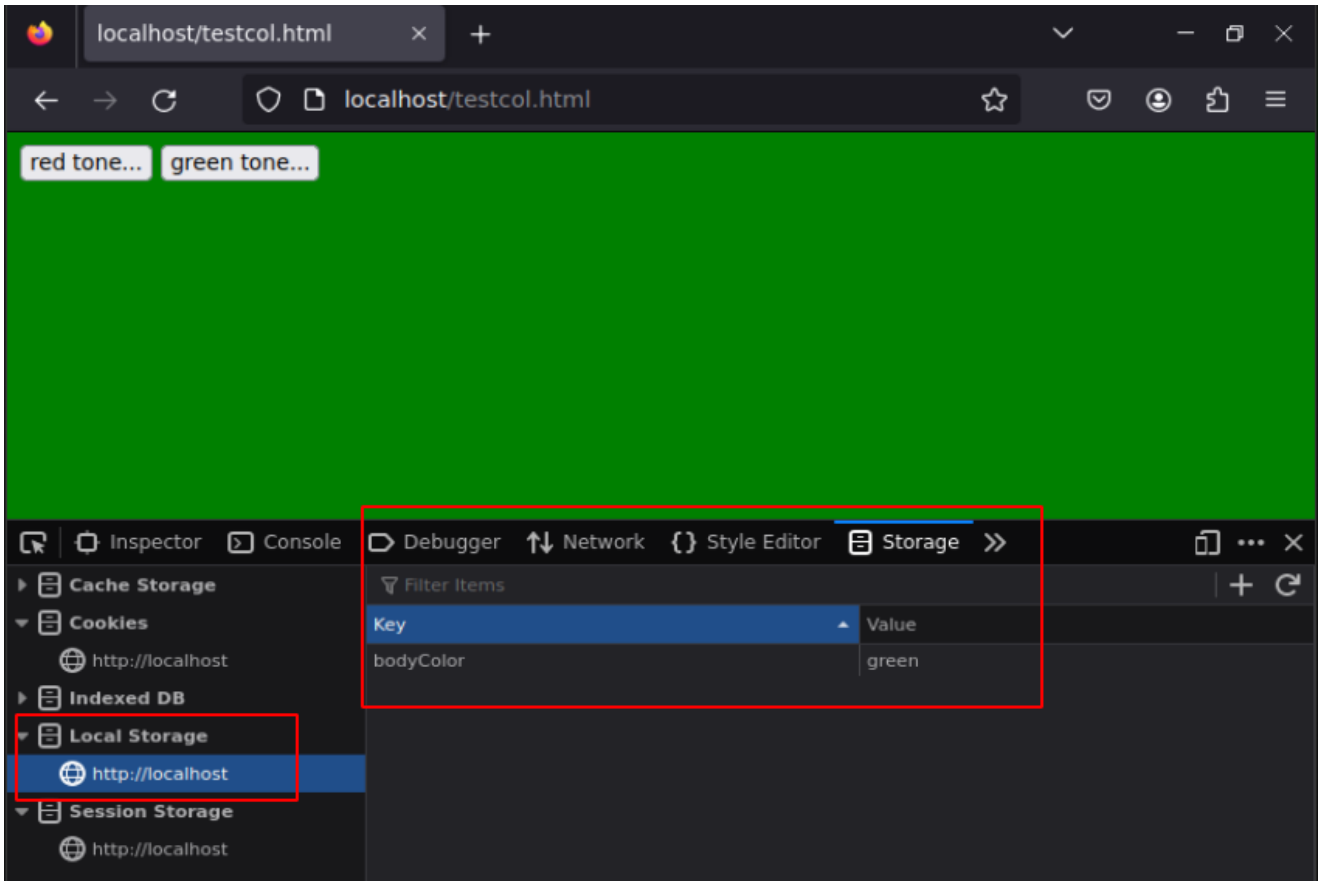


Куки не прописались, так как это разные домены разные.

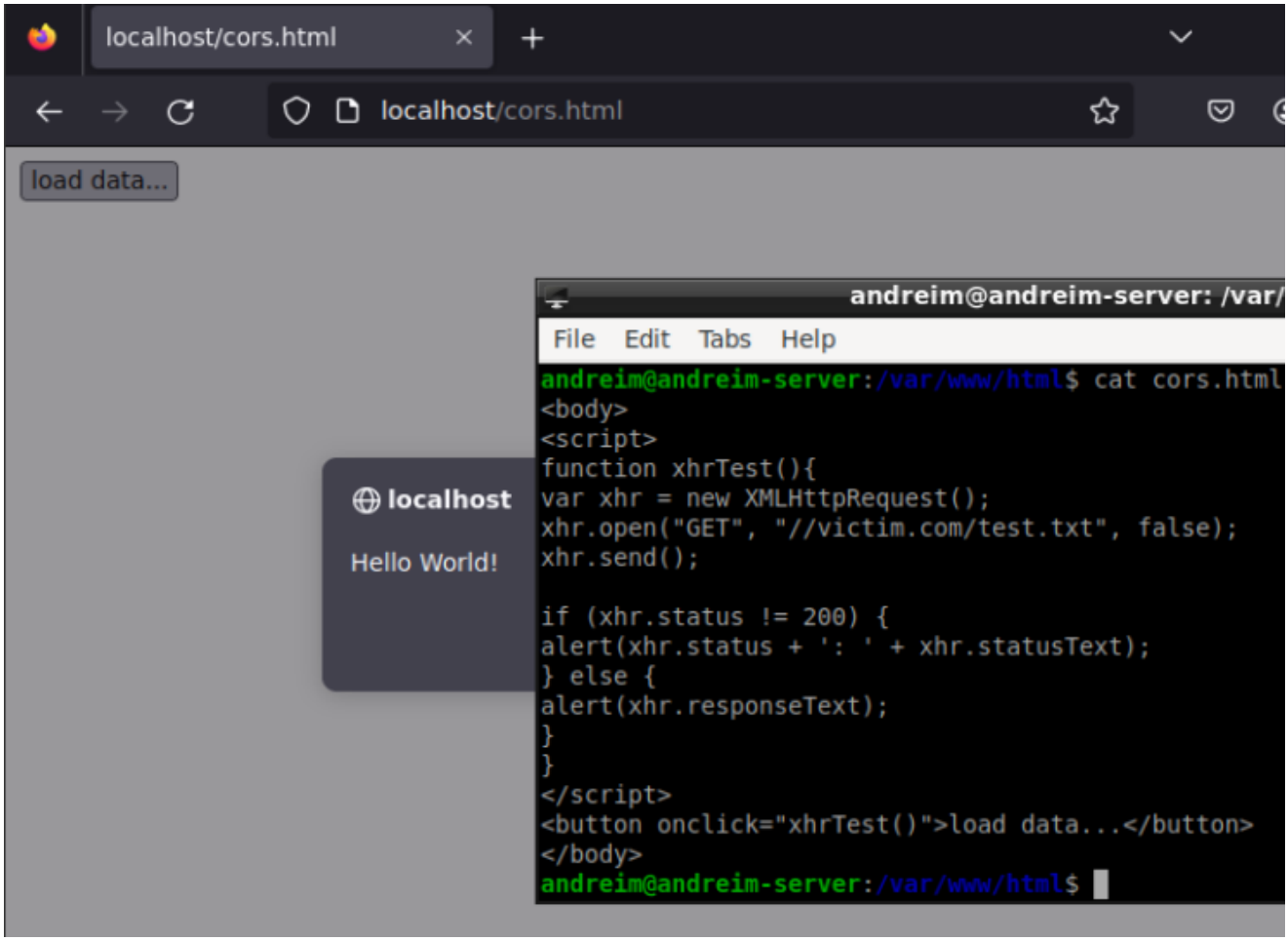
2. Дан сайт, который при нажатии на кнопку меняет цвет фона. Дописать, чтобы при открытии сайта JS обращался в web storage за цветом фона и восстанавливает его.

```
<body> <script> function changeBodyColor(color) {
document.body.style.backgroundColor = color; } </script> <button
onclick="changeBodyColor('red')">Make it hell!</button> <button
onclick="changeBodyColor('green')">Make it grass!</button> </body>
```





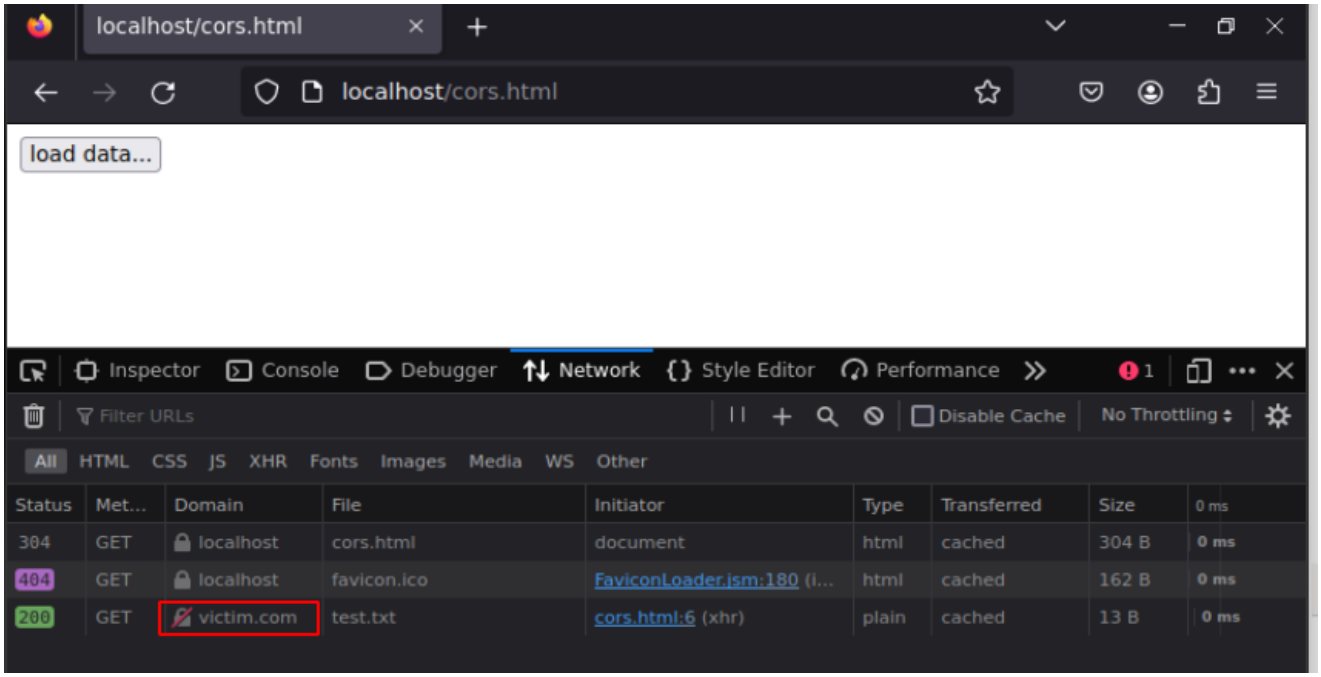
3. (_) Самостоятельно настроить CORS на http://victim.com. Разрешить http://localhost с помощью CORS делать запросы к http://victim.com.



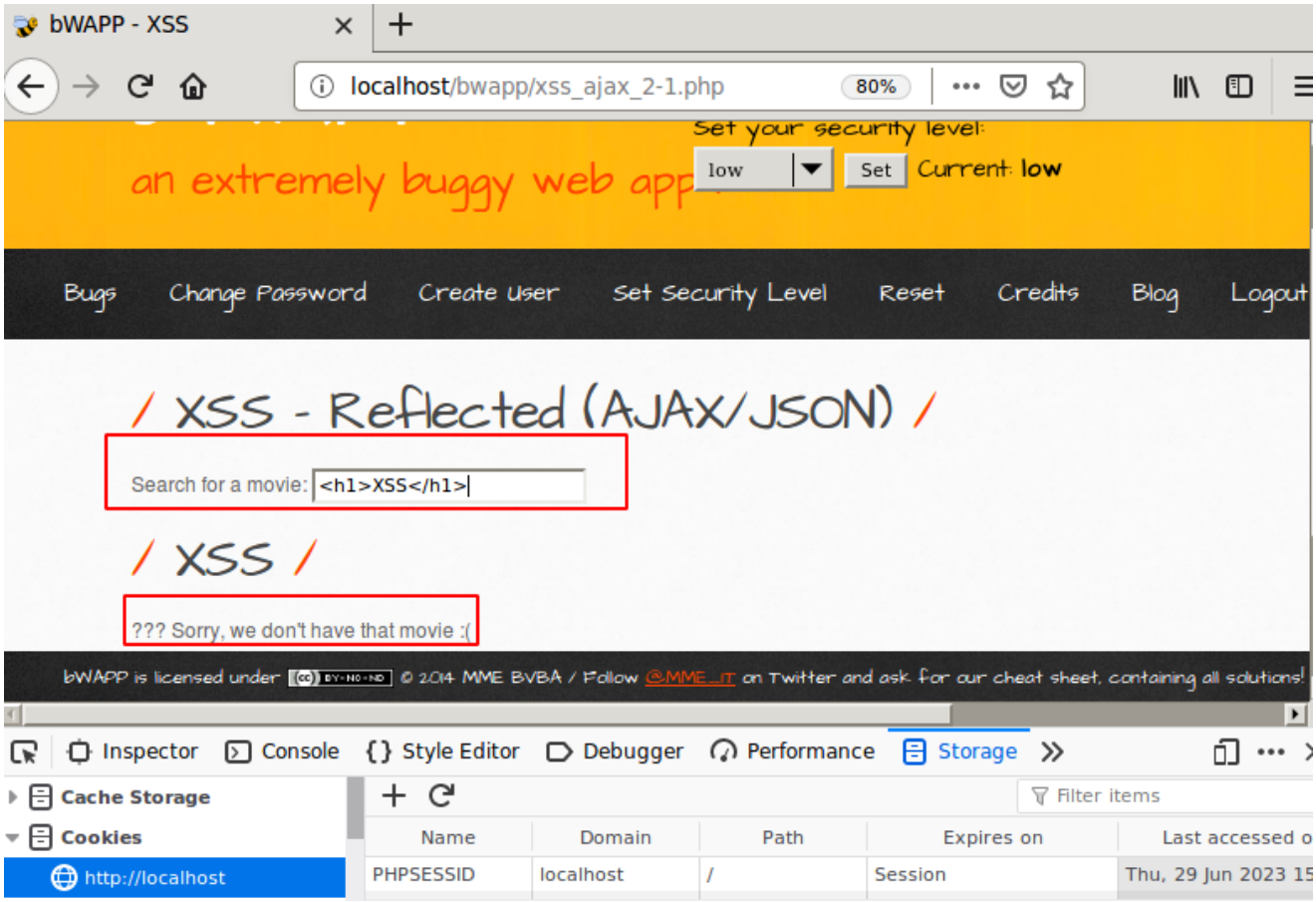
- добавляем CORS
  `server_name victim.com;... location ... add_header "Access-Control-Origin" "http://localhost";`

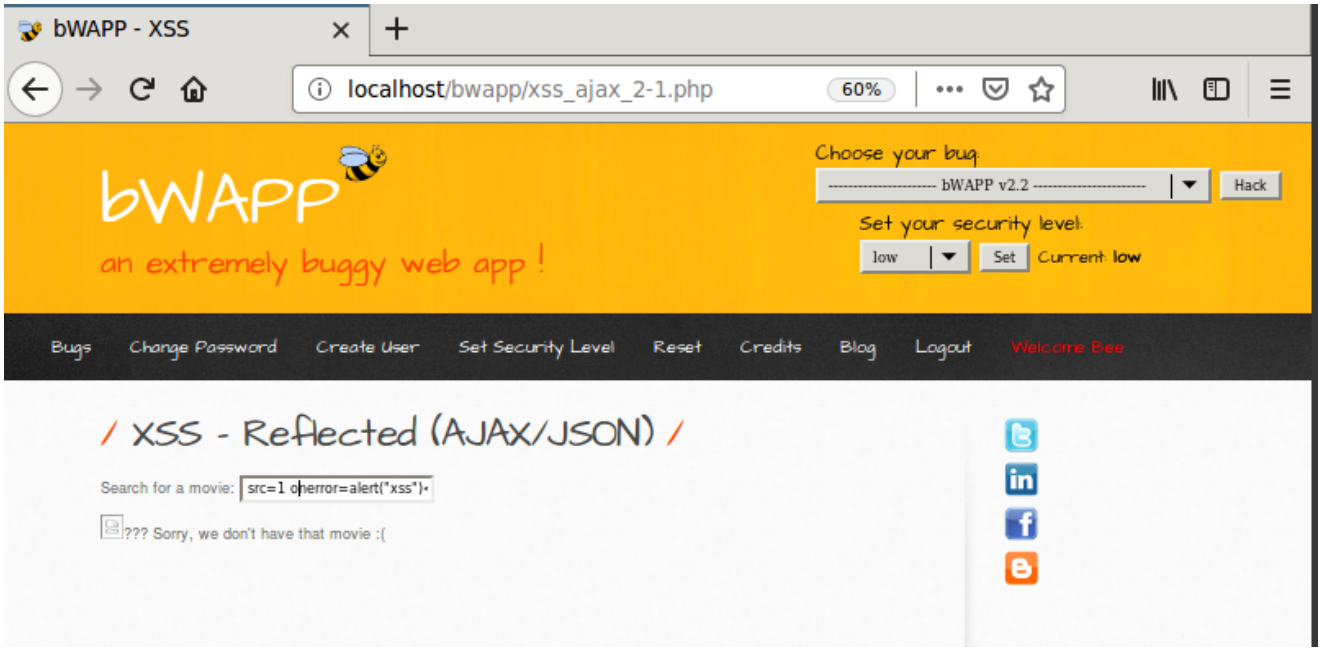
```
 GNU nano 6.2                /etc/nginx/sites-available/default *
        # Add index.php to the list if you are using PHP
        index index.html index.htm index.nginx-debian.html;

        #server_name _;
        #server_name localhost;
        #server_name attacker.com;
        server_name victim.com;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
        # add security
                # add_header Content-Security-Policy "script-src none"
        # add Cokiess
#       add_header 'Set-Cookie' 'test1=attacker-to-attacker; Max-age=3600; Doma>
#       add_header 'Set-Cookie' 'test2=attacker-to-victim; Max-age=3600; Domain>
add_header 'Access-Control-Allow-Origin' 'http://localhost';

                try_files $uri $uri/ =404;
        }
```

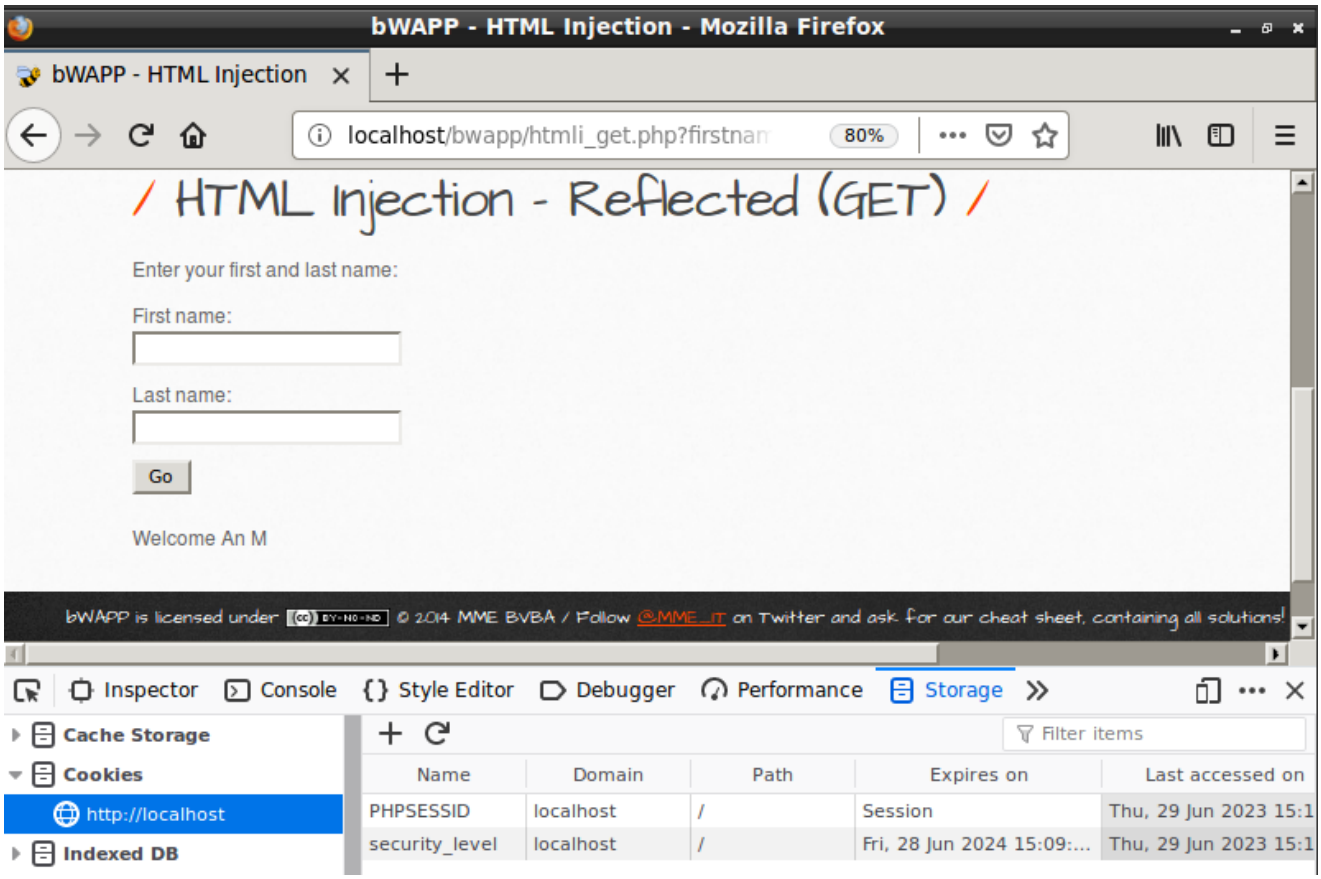
- `sudo nginx -s reload`

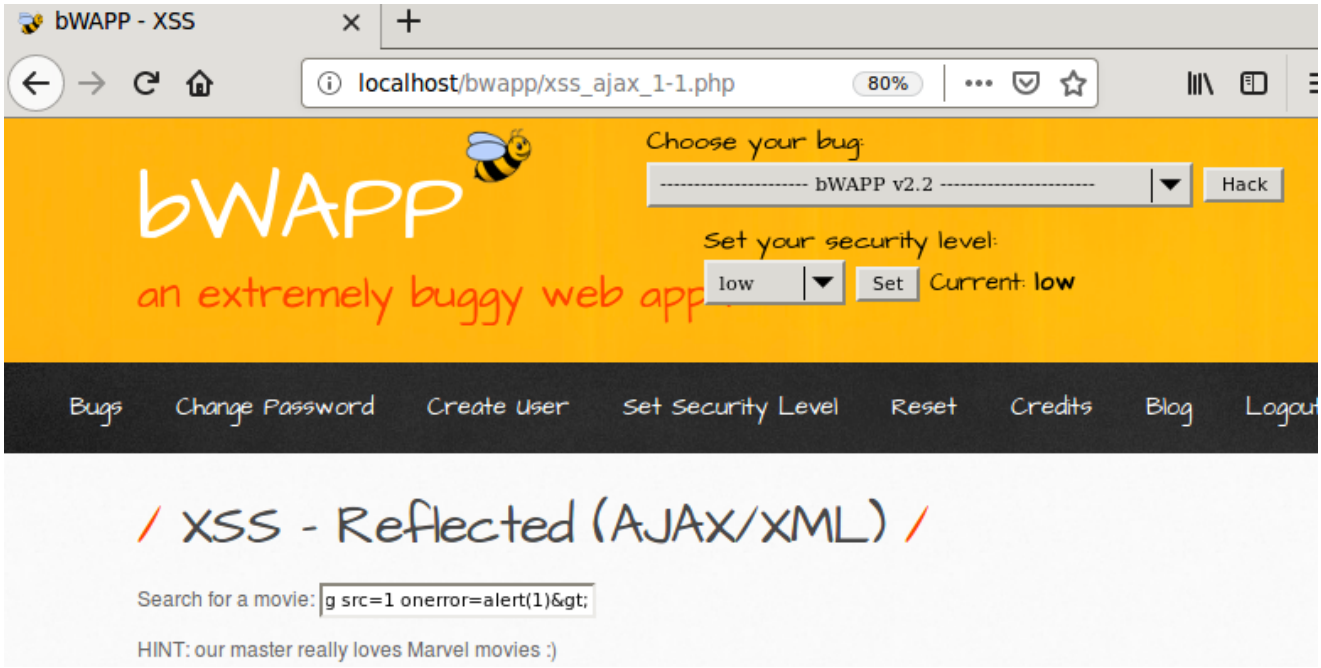


4. (_) Решить как можно больше XSS на уровне low в bWAPP.


bWAPP - XSS
localhost/bwapp/xss_ajax_2-1.php
80%
an extremely buggy web app
Set your security level:
low
Set
Current: low
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
/ XSS - Reflected (AJAX/JSON) /
Search for a movie: <h1>XSS</h1>
/ XSS /
??? Sorry, we don't have that movie :(
bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions!
Inspector
Console
Style Editor
Debugger
Performance
Storage
Cache Storage
Cookies
http://localhost
Filter items
Name
Domain
Path
Expires on
Last accessed o
PHPSESSID
localhost
/
Session
Thu, 29 Jun 2023 15


bWAPP - XSS
localhost/bwapp/xss_ajax_2-1.php
60%
bWAPP
an extremely buggy web app !
Choose your bug:
bWAPP v2.2
Hack
Set your security level:
low
Set
Current: low
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
Welcome Bee
/ XSS - Reflected (AJAX/JSON) /
Search for a movie: src=1 onerror=alert("xss")
??? Sorry, we don't have that movie :(


bWAPP - HTML Injection - Mozilla Firefox
bWAPP - HTML Injection
localhost/bwapp/htmli_get.php?firstnan
80%
/ HTML Injection - Reflected (GET) /
Enter your first and last name:
First name:
Last name:
Go
Welcome An M
bWAPP is licensed under © 2014 MME BVBA / Follow @MME_IT on Twitter and ask for our cheat sheet, containing all solutions!
Inspector
Console
Style Editor
Debugger
Performance
Storage
Cache Storage
Cookies
http://localhost
Indexed DB
Filter items
Name
Domain
Path
Expires on
Last accessed on
PHPSESSID
localhost
/
Session
Thu, 29 Jun 2023 15:1
security_level
localhost
/
Fri, 28 Jun 2024 15:09:...
Thu, 29 Jun 2023 15:1


bWAPP - XSS
localhost/bwapp/xss_ajax_1-1.php
80%
bWAPP
an extremely buggy web app
Choose your bug:
bWAPP v2.2
Hack
Set your security level:
low
Set
Current: low
Bugs
Change Password
Create User
Set Security Level
Reset
Credits
Blog
Logout
/ XSS - Reflected (AJAX/XML) /
Search for a movie: g src=1 onerror=alert(1)>
HINT: our master really loves Marvel movies :)