Vectors Hack Wordpress

Индексирование сайта

Первым этапом любого теста обычно бывает сбор информации о цели.

И тут очень часто помогает неправильная настройка индексирования сайта, которая позволяет неавторизованным пользователям просматривать содержимое отдельных разделов сайта и, например, получить информацию об установленных плагинах и темах, а также доступ к конфиденциальным данным или резервным копиям баз данных.

Чтобы проверить, какие директории видны снаружи, проще всего воспользоваться Гуглом. Достаточно выполнить запрос Google Dorks типа site:example.com intitle:«index of» inurl:/wp-content/.

В операторе inurl: можно указать следующие директории:

```
/wp-content/
/wp-content/languages/plugins
/wp-content/languages/themes
/wp-content/plugins/
/wp-content/themes/
/wp-content/uploads/
```

Если сможешь просмотреть /wp-content/plugins/, следующий шаг по сбору информации об установленных плагинах и их версиях значительно упрощается. Естественно, запретить индексирование можно с помощью файла robots.txt.

Так как по умолчанию он не включен в установочный пакет WordPress, его необходимо создать самому и закинуть в корневую директорию сайта. Мануалов по созданию и работе с файлом robots.txt довольно много, поэтому оставлю эту тему для самоподготовки. Приведу лишь один из возможных вариантов:

```
User-Agent: *
Disallow: /cgi-bin
Disallow: /wp-login.php
Disallow: /wp-admin/
Disallow: /wp-includes/
Disallow: /wp-content/
Disallow: /wp-content/plugins/
Disallow: /wp-content/themes/
Disallow: /?author=*
Allow: /
```

Если в файлах, хранящихся в папке uploads, имеются сведения конфиденциального характера, добавляем к этому списку строчку: Disallow: /wp-content/uploads/. С другой стороны, в файле robots.txt не рекомендуется размещать ссылки на директории, которые были созданы специально для хранения чувствительной информации. Иначе этим самым ты облегчишь злоумышленнику задачу, так как это первое место, куда обычно все заглядывают в поисках «интересненького».

Определение версии WordPress

Еще один важный шаг — идентификация версии CMS. Иначе как подобрать подходящий сплоит? Существует три быстрых способа для определения используемой на сайте версии WordPress:

1. Найти в исходном коде страницы. Она указана в метатеге generator: /> или же в тегах : <link rel='stylesheet' id='twentyfifteen-style-css' href='http://.../wordpress/wp-content/themes/twentyfifteen/style.css?ver=4.1.1'… />.
2. Найти в файле readme.html (рис. 1), который входит в состав установочного пакета и находится в корне сайта. Файл может иметь и другие названия типа readme-ja.html.
3. Найти в файле ru_RU.po (рис. 2), который входит в состав установочного пакета и расположен по адресу /wp-content/languages/: «Project-Id-Version: WordPress 4.1.1\n».

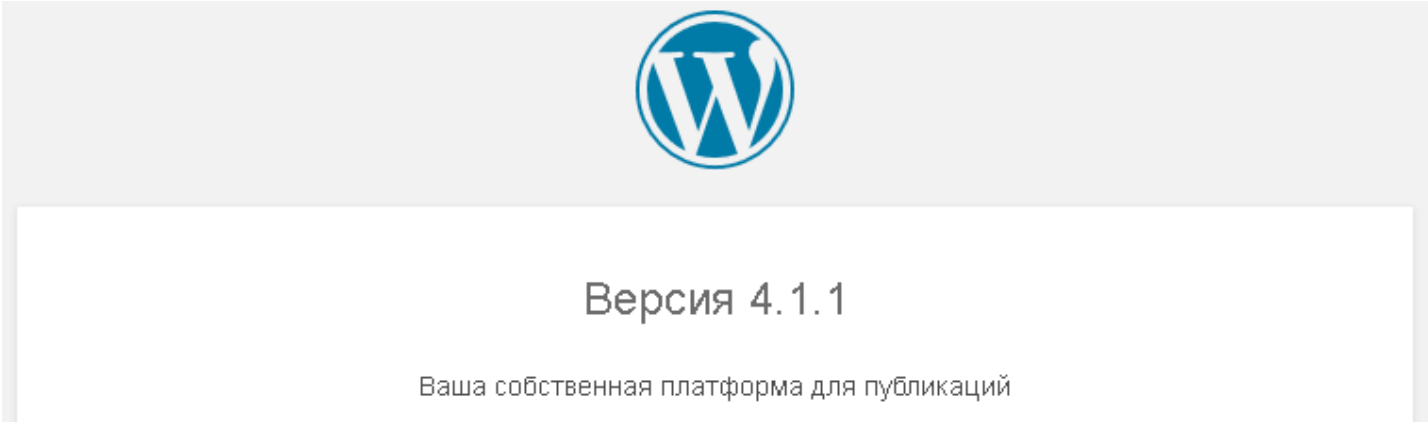

Рис. 1. Версия WordPress в файле readme.html

```
# Translation of 4.1.x in Russian
# This file is distributed under the same license as the 4.1.x package.
msgid ""
msgstr ""
"PO-Revision-Date: 2015-03-26 16:18:23+0000\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"
"Plural-Forms: nplurals=3; plural=(n%10==1 && n%100!=11 ? 0 : n%10>=2 && n%10<=4 && (n%100<10 || n%100>=20) ? 1 : 2);\n"
"X-Generator: GlotPress/1.0-alpha-1000\n"
"Project-Id-Version: 4.1.x\n"
```

Рис. 2. Подсматриваем версию WordPress в файле ru_RU.po

Автоматизация процесса тестирования

Исследованием безопасности WordPress занялись не вчера, поэтому существует достаточное количество инструментов, позволяющих автоматизировать рутинные задачи.

Nmap:

— определение версии и темы с помощью скрипта http-wordpress-info

```
nmap -sV --script http-wordpress-info <ip>
```

— подбор пароля по словарям

```
nmap -p80 --script http-wordpress-brute --script-args
'userdb=users.txt,passdb=passwords.txt' example.com
```

Metasploit:

— модуль для определения версии: auxiliary/scanner/http/wordpress_scanner;
— модуль для определения имени пользователя auxiliary/scanner/http/wordpress_login_enum.

WPScan:

— перечисление установленных плагинов: wpscan --url www.exmple.com --enumerate p;
— перечисление установленных тем: wpscan --url www.exmple.com --enumerate t;
— перечисление установленного timthumbs: wpscan --url www.example.com --enumerate tt;
— определение имени пользователя: wpscan --url www.example.com --enumerate u;
— подбор пароля по словарю для пользователя admin: wpscan --url www.example.com --wordlist wordlist.txt --username admin;
— подбор пароля с использованием связки имя пользователя / пароль с числом потоков, равным 50: wpscan --url www.example.com --wordlist wordlist.txt --threads 50.

Определение установленных компонентов

Теперь давай соберем информацию об установленных плагинах и темах независимо от того, активированы они или нет. Прежде всего такую информацию можно выудить из исходного кода HTML-страницы, например по JavaScript-ссылкам, из комментариев и ресурсов типа CSS, которые подгружаются на страницу. Это самый простой способ получения информации об установленных компонентах. Например, строчки ниже указывают на используемую тему twentyeleven:

```
<link rel="stylesheet" type="text/css" media="all" href="http://example.com/wp-content/themes/twentyeleven/style.css" />
<script src="http://example.com/wp-content/themes/twentyeleven/js/html5.js" type="text/javascript"></script>
```

Далее, HTTP-заголовки, такие как X-Powered-By, могут указывать на наличие плагина (например, на плагин W3 Total Cache).

Так как информация о плагинах не всегда отображается в исходном коде HTML-страницы, то обнаружить установленные компоненты можно с помощью утилиты WPScan (см. врезку). Только не забывай, что перебор путей плагинов зафиксируется в логах веб-сервера. Получив данные об установленных компонентах, уже можно приступать к поиску уязвимостей своими силами либо найти общедоступные эксплойты на ресурсах типа rapid7 или exploit-db.

Определение имени пользователей

По умолчанию в WordPress каждому пользователю присваивается уникальный идентификатор, представленный в виде числа: example.com/?author=1. Перебирая числа, ты и определишь имена пользователей сайта. Учетная запись администратора admin, которая создается в процессе установки WordPress, идет под номером 1, поэтому в качестве защитной меры рекомендуется ее удалить.

Брутфорс wp-login

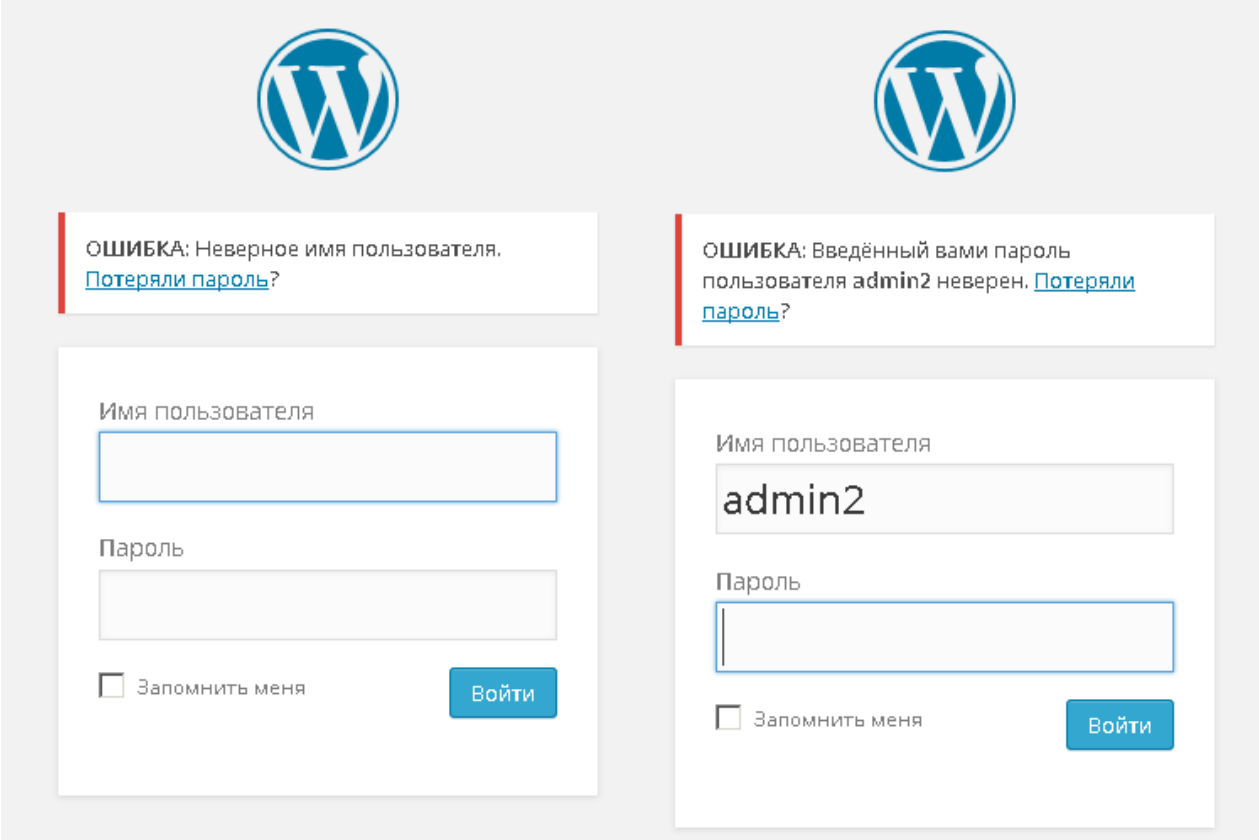

Рис. 3. Ошибки при аутентификации пользователя

Зная имя пользователя, можно попробовать подобрать пароль к панели администрирования. Форма авторизации WordPress на странице wp-login.php весьма информативна (рис. 3), особенно для злоумышленника: при вводе неправильных данных появляются подсказки о неверном имени пользователя или пароле для конкретного пользователя. Разработчикам известно о данной особенности, но ее решили оставить, так как подобные сообщения удобны для пользователей, которые могли забыть свой логин и/или пароль.