

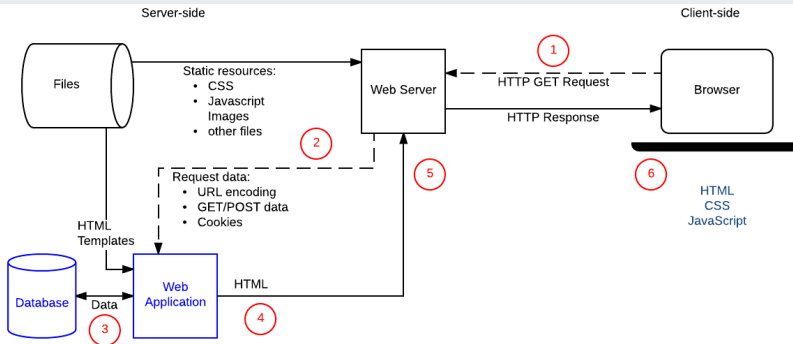
Практикум 2

Поиск уязвимостей в специально предназначенном для этого веб-приложении DVWA.

Что это такое?

Web-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

Архитектура web-приложения

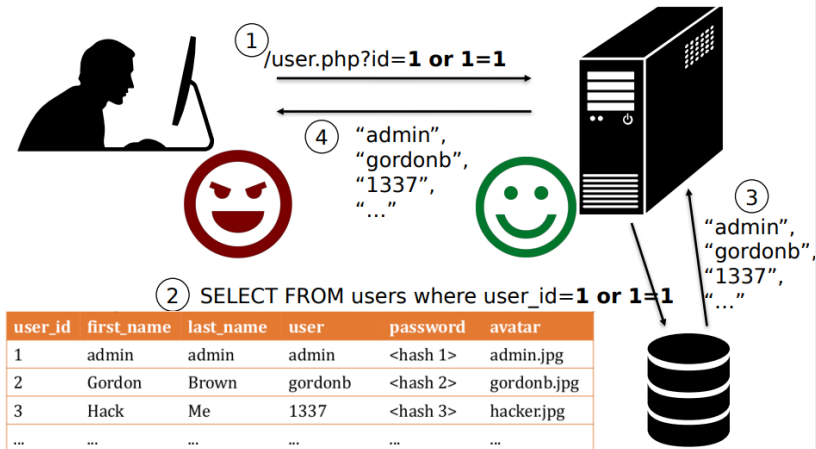


Web-сервер, предоставляя большие возможности, является "желаемой" целью для атаки злоумышленников из-за своей общедоступности и "распределенной природы".

Как следствие есть уязвимости и угрозы

- 1. Угрозы со стороны клиента
 - 2. Угрозы со стороны сервера
 - 3. Угроза одного пользователя другому
 - 4. Угроза от стороннего сервера
-
- 1. **Иньекции** - внедрение и исполнение кода на сервере
 - 2. **Brute Force** -> Получение логина/пароля администратора
 - 3. **Command Injection** – внедрение вредоносной(или не ояень) команды в уязвимую систему

Пример SQL-инъекции



Что это такое?

Damn Vulnerable Web Application (DVWA) – это веб-приложение PHP / MySQL, которое **очень и очень** уязвимо.


Для чего оно нужно?

Основная цель DVWA – помочь веб-разработчикам лучше понять процессы обеспечения безопасности веб-приложений и помочь студентам и преподавателям узнать о безопасности веб-приложений в контролируемой среде.

Внимание!

DVWA очень и очень уязвим! Не загружайте его в общедоступную html-папку хостинг-провайдера или на любые серверы, обращенные к Интернету, поскольку они будут скомпрометированы. Рекомендуется установить данное приложение для тестирования на виртуальную машину или Kali Linux, В случае виртуальной машины мы можем предоставить приложению любое требуемое ему окружение без ущерба для других компонентов.

Так выглядит основное меню DVWA, слева атаки.



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

DVWA Security

PHP Info

About

Logout

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to **practice some of the most common web vulnerabilities**, with **various levels of difficulty**, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are **both documented and undocumented vulnerability** with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users!)

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! **Do not upload it to your hosting provider's public html folder or any Internet facing servers**, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can downloading and install [XAMPP](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

Какие уязвимости имеет DVWA:

- **Brute Force:** Brute Force HTTP формы страницы входа; используется для тестирования инструментов по атаке на пароль методом грубой силы и показывает небезопасность слабых паролей
- **Исполнение (внедрение) команд:** выполнение команд уровня операционной системы
- **Межсайтовая подделка запроса (CSRF):** позволяет «атакующему» изменить пароль администратора приложений
- **Внедрение (инклюд) файлов:** позволяет «атакующему» присоединить удалённые/локальные файлы в веб-приложение.
- **SQL внедрение:** позволяет «атакующему» внедрить SQL выражения в HTTP из поля ввода, DVWA включает слепое и основанное на ошибке SQL внедрение
- **Межсайтовый скриптинг (XSS):** «атакующий» может внедрить свои скрипты в веб-приложение/базу данных. DVWA включает отражённую и хранимую XSS

Введём в поле User ID: `%' or 0=0 union select null, version()`
Получим версию базы данных.

DVWA

Vulnerability: SQL Injection

User ID:

ID: `%' or 0=0 union select null, version()` #
First name: admin
Surname: admin

ID: `%' or 0=0 union select null, version()` #
First name: Gordon
Surname: Brown

ID: `%' or 0=0 union select null, version()` #
First name: Hack
Surname: Me

ID: `%' or 0=0 union select null, version()` #
First name: Pablo
Surname: Picasso

ID: `%' or 0=0 union select null, version()` #
First name: Bob
Surname: Smith

ID: `%' or 0=0 union select null, version()` #
First name:
Surname: 10.1.26-MariaDB-0+deb9u1

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavtuna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

Версия базы данных (**10.1.26-MariaDB-0+deb9u1**), и она показана на скриншоте выше.

До этого SQL-инъекцию мы делали, используя браузер. Однако мы можем воспользоваться **cURL**.

cURL

cURL — (client for URL) кроссплатформенная служебная программа командной строки, позволяющая взаимодействовать с множеством различных серверов по множеству различных протоколов с синтаксисом URL.

*информацию об опциях curl мы всегда можем узнать в **man curl***

1

Заходим в командную строку linux.

2

Нам нужно ввести команду: **curl -v <URL атаки в DVWA в кавычках> -b "PHPSESSID=<YourCOOKIE>;security=low"**

3

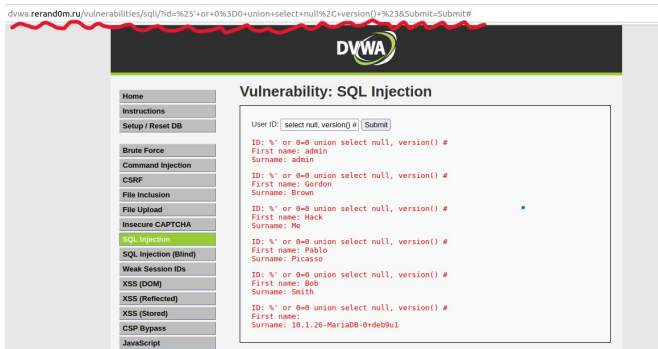
YourCOOKIE - это ваши куки на сайте DVWA. Каких их узнать?

4

Если у вас Firefox, то заходим в другие инструменты -> инструменты веб-разработчика -> хранилище -> куки.
В других браузерах примерно похожий алгоритм.

Теперь разберёмся как узнать **URL атаки в DVWA**.

Осуществляем SQL-инъекцию, видим следующую картинку. Нам понадобится то, что выделено красной волнистой линией.



Внимание!

Не забывайте про кавычки!

Если всё сделали правильно, то увидим:

```
</form>
<pre>ID: ' ' or 0=0 union select null, version() #<br />First name: admin<br />Surname: admin</pre><pre>ID: ' ' or 0=0 union select null, version() #<br />First name: Gordon<br />Surname: B
om</pre><pre>ID: ' ' or 0=0 union select null, version() #<br />First name: Hack<br />Surname: Mes</pre><pre>ID: ' ' or 0=0 union select null, version() #<br />First name: Pablo<br />Surname: Picasso</pre>
<pre>ID: ' ' or 0=0 union select null, version() #<br />First name: Bob<br />Surname: Smith</pre><pre>ID: ' ' or 0=0 union select null, version() #<br />First name: <br />Surname: 10.1.26-MariaDB-0+deb9ui
</pre>
```

Если всё сделали правильно, то в ответе увидим версию базы данных.

User-Agent

Кроме того, в curl мы можем указать конкретное значение User-Agent, определённым ключом.

Реализуйте две любые атаки из трёх предложенных на <http://dvwa.rerand0m.ru>:

- **SQL-инъекция**
- **Brute Force** - перебор пароля из списка предложенных
- **Исполнение (внедрение) команд**

Атаки должны быть реализованы с помощью команды curl. При этом User-Agent должен принимать следующие значения:

- В случае **SQL-инъекции** User-Agent: **<фамилияио>-sql**i
- В случае **Brute Force** User-Agent: **<фамилияио>-bf**
- В случае **Исполнение (внедрение) команд** User-Agent: **<фамилияио>-cj**

В **Brute Force** предполагается, что админ известен. Достаточно перебрать с десятков паролей. В случае **исполнения команд** не стоит вызывать команды, удаляющие что-то.

В качестве результата вам нужно будет прислать две отфильтрованные трассы, соответствующие веб-атакам. Название трасс:

- В случае **SQL-инъекции** название: **<фамилияио>-sqli.pcap**
- В случае **Broute Force** название: **<фамилияио>-bf.pcap**
- В случае **Исполнение (внедрение) команд** название: **<фамилияио>-ci.pcap**

Две сохранённые трассы нужно объединить в архив **<фамилияио>-<группа>- p2.zip** и отправить на почту, указав в теме письма **<вуз>-<группа>- p2**.

Две сохранённые трассы нужно объединить в архив **<фамилияио>-<группа>- p2.zip** и отправить на почту, указав в теме письма **<вуз>-<группа>- p2**.