Индивидуальный проект. Этап 2

Установка DVWA

Парфенова Елизавета Евгеньевна

Содержание

# 1 Цель работы

Установка DVWA в гостевую систему к Kali Linux.

# 2 Теоретическое введение

Damn Vulnerable Web Application (DVWA) — это веб-приложение на PHP/MySQL, которое очень сильно уязвимо. Его главная цель — помочь профессионалам по безопасности протестировать их навыки и инструменты в легальном окружении, помочь веб-разработчикам лучше понять процесс безопасности веб-приложений и помочь и студентам и учителям в изучении безопасности веб-приложений в контролируем окружении аудитории. [1]

Некоторые из уязвимостей веб приложений, который содержит DVWA:

* *Брутфорс*: Брутфорс 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 (инклуд) 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.
* *Небезопасная выгрузка файлов*: Позволяет «атакующему» выгрузить вредоносные файлы на веб сервер.
* *Межсайтовый скриптинг (XSS)*: «Атакующий» может внедрить свои скрипты в веб приложение/базу данных. DVWA включает отражённую и хранимую XSS.
* *Пасхальные яйца*: раскрытие полных путей, обход аутентификации и некоторые другие.

DVWA имеет несколько уровней безопасности, они меняют уровень безопасности каждого веб приложения в DVWA:

* *Невозможный* — этот уровень должен быть безопасным от всех уязвимостей. Он используется для сравнения уязвимого исходного кода с безопасным исходным кодом.
* *Высокий* — это расширение среднего уровня сложности, со смесью более сложных или альтернативных плохих практик в попытке обезопасить код. Уязвимости не позволяют такой простор эксплуатации как на других уровнях.
* *Средний* — этот уровень безопасности предназначен главным образом для того, чтобы дать пользователю пример плохих практик безопасности, где разработчик попытался сделать приложение безопасным, но потерпел неудачу.
* *Низкий* — этот уровень безопасности совершенно уязвим и совсем не имеет защиты. Его предназначение быть примером среди уязвимых веб приложений, примером плохих практик программирования и служить платформой обучения базовым техникам эксплуатации.

# 3 Выполнение лабораторной работы

Установку DVWA будем выполнять согласно инструкции в репозитории гитхаба (в том числе видео-инструкции) [2]

Начнем установку с клонирования указанного репозитория на наш компьютер. Для этого в терминале введм команду *git clone https://github.com/digininja/DVWA.git*. Далее по рекомендации перенесем созданный каталог в директорию /var/www/html командой *sudo mv DVWA /var/www/html* (зайдем с правами администратора) и проверим, что все получилось корректно. (рис. 1).

![Клонирование репозитория DVWA](data:application/octet-stream;base64,)

Рис. 1: Клонирование репозитория DVWA

Далее попробуем зайти на сервер по ссылке http://localhost, но увидим, что попытка не удалась. (рис. 2)

![Неудачная попытка входа](data:application/octet-stream;base64,)

Рис. 2: Неудачная попытка входа

Запустм стандартный сервер apache2 командой *sudo service apache2 start*, (рис. 3) и попробовав перезагрузить страницу, увидим, что все сработало и открылась стартовая страница сервера apache2. (рис. 4)

![Запуск сервера apache2](data:application/octet-stream;base64,)

Рис. 3: Запуск сервера apache2

![Удачное открытие страницы](data:application/octet-stream;base64,)

Рис. 4: Удачное открытие страницы

Далее попробуем перейти на страницу http://localhost/DVWA/ (рис. 5), но увидим, что это вышло не совсем корректно. Как и требует текст высветившейся ошибки, поработаем с файлом config.

![Неудачная попытка открытия DVWA](data:application/octet-stream;base64,)

Рис. 5: Неудачная попытка открытия DVWA

Для этого перейдем в директорию DVWA и посмотрим, что лежит в директории config командой *ls*. Далее скопируем дист-версию конфигурационного файла в файл config.inc.php (для большей безопасности именно скопируем, а не перенесем) (рис. 6), а затем прочитаем файл командой *cat config/config.inc.php* (рис. 7). Из этого файла нам особенно понадобятся учетные данные для базы данных, которые мы не изменяем.

![Копирование конфигурационного файла](data:application/octet-stream;base64,)

Рис. 6: Копирование конфигурационного файла

![Прочтение конфигаруционного файла](data:application/octet-stream;base64,)

Рис. 7: Прочтение конфигаруционного файла

Увидим, что мы смогли войти на страницу http://localhost/DVWA/setup.php (рис. 8), но при попытке создать базу данных мы ничего не получаем

![Открытие страницы http://localhost/DVWA/setup.php](data:application/octet-stream;base64,)

Рис. 8: Открытие страницы http://localhost/DVWA/setup.php

Запустим стандартную базу данных mariadb командой *service mariabd start* (рис. 9)

![Запуск базы данных](data:application/octet-stream;base64,)

Рис. 9: Запуск базы данных

Далее в другом окне терминала, подключившийсь к БД от имени пользователя root командами *sudo -su* и *mysql*, создадим нового пользователя следующими команды (рис. 10), проверяя, чтобы вывод после них не свидетельствовал о какой-либо ошибке:

* *MariaDB [(none)]> create database dvwa;*
* *MariaDB [(none)]> create user dvwa@localhost identified by ‘p@ssw0rd’;*
* *MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;\*
* *MariaDB [(none)]> flush privileges;*

При введении первых трех команд ориентируемся на данные из конфигурационного файла, а последней командой мы перезапускаем БД.

![Создание нового пользователя базы данных (dvwa)](data:application/octet-stream;base64,)

Рис. 10: Создание нового пользователя базы данных (dvwa)

Проверим корректность создания нового пользователя bd с помошью команды *mysql -u dvwa -pp@sw0rd*, открыв новыое окно терминала. (рис. 11) Этой командой мы входим в базу данных по “учетной записи” созданного пользователя. Все получается корректно.

![Проверка создания нового пользовтаеля](data:application/octet-stream;base64,)

Рис. 11: Проверка создания нового пользовтаеля

После выполненных манипуляций попробуем снова создать базу данных через сайт. Для этого проскролим сайт вниз и нажмем на кнопку “Create/Reset Database”. Нас перекидывает на страницу входа, значит все получилось верно. Вводим в пустые поля стандартные “admin” и “password” (рис. 12) и попадаем на страницу DVWA (рис. 13).

![Данные входа на страницу](data:application/octet-stream;base64,)

Рис. 12: Данные входа на страницу

![Успешный вход на страницу DVWA](data:application/octet-stream;base64,)

Рис. 13: Успешный вход на страницу DVWA

На этом стандартная установка DVWA окночена.

# 4 Выводы

Мы установили DVWA в гостевую систему к Kali Linux.

# Список литературы

1. Damn Vulnerable Web Application (DVWA) [Электронный ресурс]. Инструменты Kali Linux, 2024. URL: <https://kali.tools/?p=1820>.

2. DVWA [Электронный ресурс]. GitHub, Inc., 2024. URL: <https://github.com/digininja/DVWA>.