Индивидуальный проект. Этап №2

Отчёт к этапу индивидуального проекта

Зайцева Анна Дмитриевна, НПМбд-02-21

Table of Contents

Цель работы

Цель работы — Установка DVWA.

Задание

- Установить DVWA в гостевую систему к Kali Linux из репозитория: https://github.com/digininja/DVWA.

Теоретические выкладки

DVWA - это уязвимое веб-приложение, разработанное на PHP и MYSQL.

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 четыре уровня 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.

Выполнение этапа индивидуального проекта

1) В установленной при выполнении предыдущего этапа индивидуального проекта операционной системе настроим DVWA. Это происходит на нашем локальном хосте, поэтому нужно перейти в директорию `/var/www/html`. Затем клонирую нужный репозиторий GitHub (Рис. [-@fig:001]):

Клонирование репозитория

2) Проверяю, что файлы склонировались правильно и повышаю права доступа к этой папке до 777 (Рис. [-@fig:002]):

Изменение прав доступа

3) Для настройки DVWA, нужно перейти в каталог `/dvwa/config`. Проверяю содержимое каталога (Рис. [-@fig:003]):

Перемещение по директориям

4) Создаем копию файла, используемого для настройки DVWA `config.inc.php.dist` с именем `config.inc.php`. Копируем файл, а не изменяем его, чтобы имелся запасной вариант на всякий случай (Рис. [-@fig:004]):

Создание копии файла

5) Открываю файл в текстовом редакторе (Рис. [-@fig:005]):

Открытие файла в текстовом редакторе

6) Изменяю данные об имени пользователя и пароле (Рис. [-@fig:006]):

Редактирование имени пользователя и пароля

7) По умолчанию в Kali Linux установлен mysql, поэтому можно его запустить без предварительного скачивания. Выполняю проверку, запущен ли процесс (Рис. [-@fig:007]):

Запуск mysql

Для выхода из режима проверки в консоли нажимаю q.

8) Авторизуюсь в базе данных от имени пользователя root (с паролем root). Появилась командная строка с приглашением “MariaDB”, в которой я создала нового пользователя, используя учётные данные из файла config.inc.php (Рис. [-@fig:008]):

Авторизация в бд

9) Теперь предоставим пользователю привилегии для работы с этой базой данных (Рис. [-@fig:009]):

Изменение прав

10) Необхоимо настроить сервер apache2. Для этого перехожу в соответствующую директорию и открываю файл ‘php.ini’, чтобы изменить в нём один параметр (Рис. [-@fig:010]):

Начало настройки сервера apache2

11) В раздееле ‘Fopen wrappers’ делаем так, чтобы параметры allow-url-fopen и allow-url-include были со значением On (Рис. [-@fig:011]):

Меняем параметры allow-url-fopen и allow-url-include

12) Запускаю службу веб-сервера apache и проверяю, запущена ли она (Рис. [-@fig:012]):

Запуск apache

13) DVWA, Apache и база данных настроены, поэтому теперь остаётся открыть браузер и запустить веб-приложение, введя в адресной строке 127.0.0.1/DVWA (Рис. [-@fig:013]):

Запуск веб-приложения

14) Нажимаю кнопку Create/Reset Database. У меня появилось пустое окно (Рис. [-@fig:014]):

Запуск веб-приложения

Но необходимо было авторизоваться с помощью предложенных по умолчанию данных (“admin // password”) и оказаться на домашней странице веб-приложения. На этом установка окончена.

Выводы

Приобрела практические навыки по установке уязвимого веб-приложения DVWA.

Библиография

1. Методические материалы курса