

Основы информационной безопасности

Индивидуальный проект. Этап № 5. Использование Burp Suite

Демидова Екатерина Алексеевна

Содержание

1	Постановка задачи	4
2	Теоретические сведения	5
3	Выполнение лабораторной работы	8
4	Выводы	13
	Список литературы	14

Список иллюстраций

3.1	Установка ПО	8
3.2	Создание проектка	8
3.3	Установка параметров	9
3.4	Включение Burp Proху	9
3.5	Настройка HTTP Proху браузера	10
3.6	Установка флага allow_hijacking_localhost	10
3.7	Перехват запроса на вход на сайт	11
3.8	Запрос на аутентификацию	11
3.9	Функция повторения запроса	11
3.10	Изучение ответа на запрос с функцией повторения запроса	12

1 Постановка задачи

Целью данной работы является использование Burp Suit для перехвата, изменения и изучения HTTP запросов и ответов.

2 Теоретические сведения

Damn Vulnerable Web Application (DVWA) – это веб-приложение PHP/MySQL, которое чертовски уязвимо[~ 1]. Его основная цель – помочь специалистам по безопасности проверить свои навыки и инструменты в правовой среде, помочь веб-разработчикам лучше понять процессы обеспечения безопасности веб-приложений, а также помочь студентам и преподавателям узнать о безопасности веб-приложений в контролируемом классе. обстановка помещения.

Цель DVWA – отработать некоторые из наиболее распространенных веб-уязвимостей различного уровня сложности с помощью простого и понятного интерфейса. В этом программном обеспечении существуют как задокументированные, так и недокументированные уязвимости.

DVWA имеет три 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.

Burp Suite — это интегрированная платформа для тестирования безопасности веб-приложений как в ручном, так и в автоматических режимах[~ 2].

Пакет состоит из набора утилит, среди которых есть инструменты для сбора и анализа информации, моделирования разных типов атак, перехвата запросов и ответов сервера и так далее.

- Target – создает карту сайта с подробной информацией о тестируемом приложении. Показывает, какие цели находятся в процессе тестирования, и позволяет управлять процессом обнаружения уязвимостей.
- Proxy – находится между браузером пользователя и тестируемым веб-приложением. Перехватывает все сообщения, передаваемые по протоколу HTTP(S).
- Spider – автоматически собирает данные о функциях и компонентах веб-приложения.
- Clickbandit – моделирует кликджекинг-атаки (clickjacking attacks), при которых поверх страницы приложения загружается невидимая страница, подготовленная злоумышленниками.
- DOM Invader – проверяет веб-приложение на уязвимость DOM-based меж-сайтовому скриптингу (основанному на объектной модели документа), внедрению вредоносного кода на страницу.
- Scanner (в профессиональной и корпоративной редакциях) — автоматически сканирует уязвимости в веб-приложениях. Также существует в бесплатной версии, но, предоставляет только описание возможностей. Intruder – проводит автоматические атаки различного типа, от перебора открытых веб-директорий до внедрения SQL-кода.

- Repeater – утилита для ручного манипулирования и повторной выдачи отдельных HTTP-запросов и анализа ответов приложения. Отправить запрос в Repeater можно из любой другой утилиты Burp Suite.
- Sequencer – анализирует качество случайности в выборке элементов данных. Можно использовать для тестирования сеансовых маркеров приложения или других важных элементов данных, которые должны быть непредсказуемыми, например маркеров анти-CSRF, маркеров сброса пароля и так далее. Decoder— преобразовывает закодированные данные в исходную форму или необработанные в различные закодированные и хешированные формы. Способен распознавать несколько форматов кодирования, используя эвристические методы. Comparer – предоставляет функцию визуального сравнения различий данных.

3 Выполнение лабораторной работы

Intercept HTTP traffic with Burp Proxy

Установим Burp Suit с официального сайта(рис. 3.1)

```
(root@eademidova)~[/home/eademidova]
# chmod +x burpsuite_community_linux_v2024_5_5.sh
# ./burpsuite_community_linux_v2024_5_5.sh
Unpacking JRE ...
Starting Installer ...
```

Рис. 3.1: Установка ПО

Откроем приложение и создадим временный проект с параметрами по умолчанию(рис. 3.2).

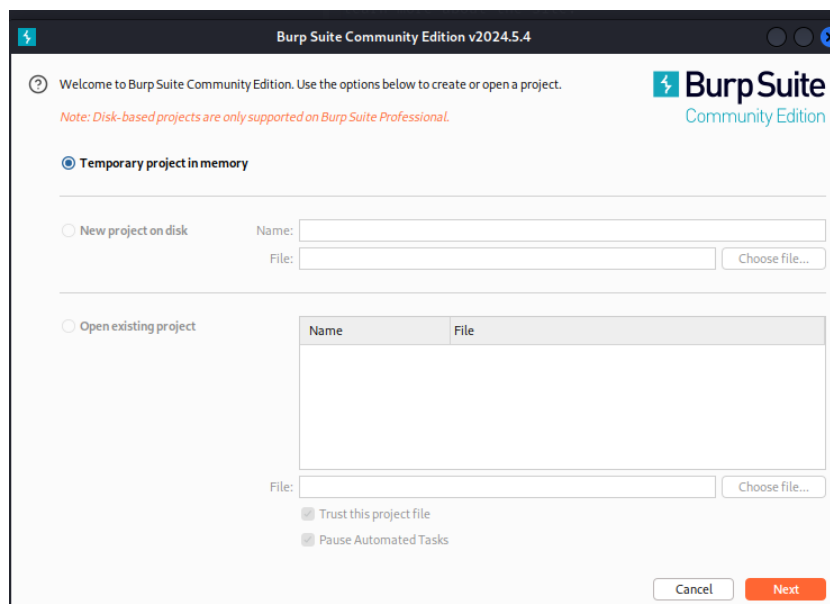


Рис. 3.2: Создание проектк

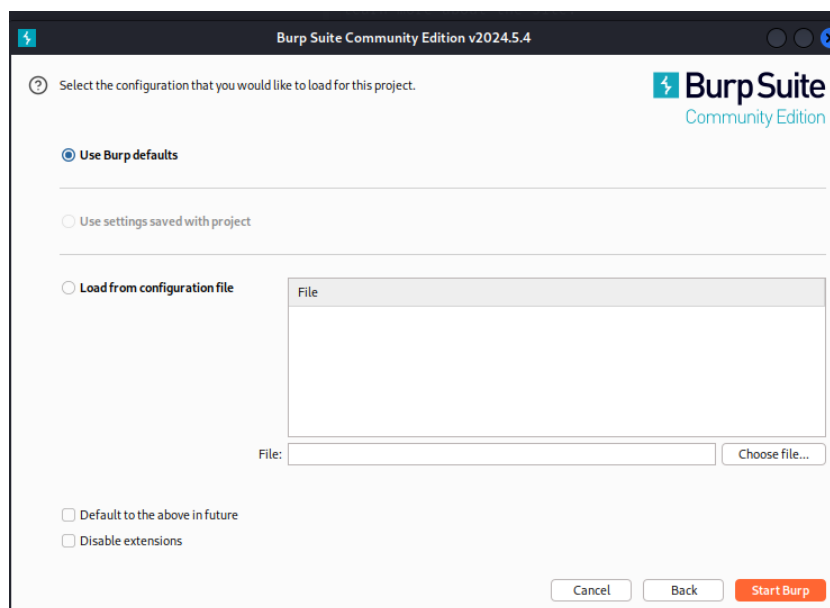


Рис. 3.3: Установка параметров

Теперь попробуем перехватить http запрос с помощью Burp Proxy. Включим перехват, а в браузере включим прокси и укажем для него адрес локального хоста, а также установим параметр, разрешающий перехват запросов локального хоста(рис. 3.4 - 3.6).

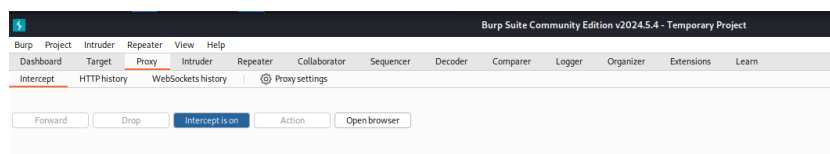


Рис. 3.4: Включение Burp Proxy

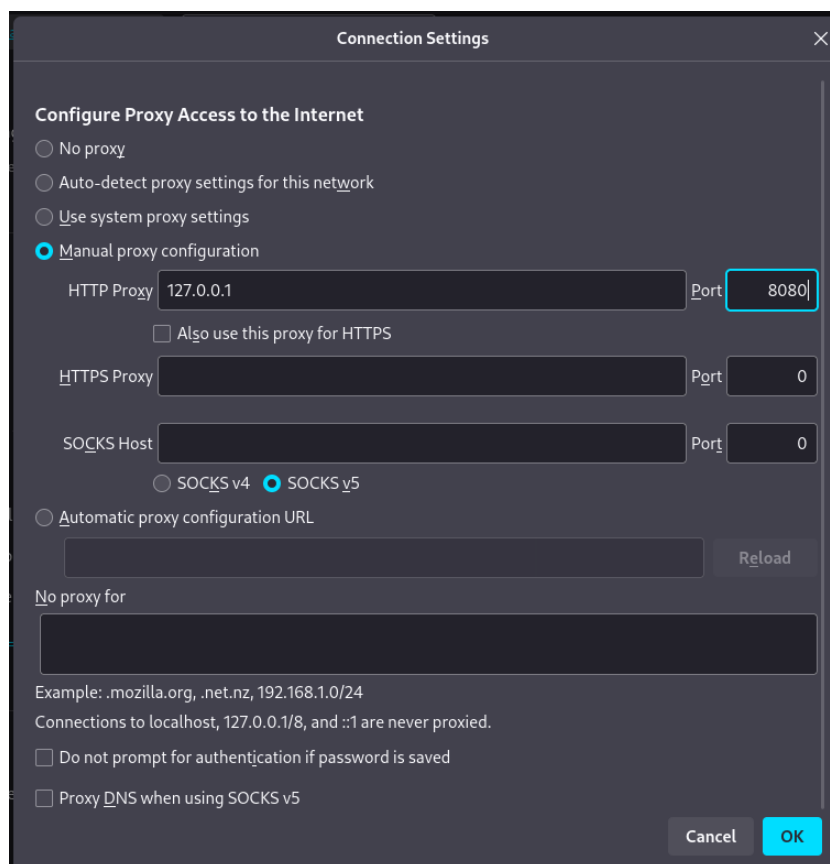


Рис. 3.5: Настройка HTTP Proxy браузера

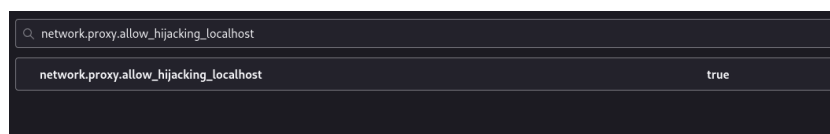


Рис. 3.6: Установка флага allow_hijacking_localhost

Можем увидеть первый перехваченный запрос: вход на сайт DVWA. Указаны адрес локального хоста, версия браузера, ОС устройства и другая информация(рис. 3.7):

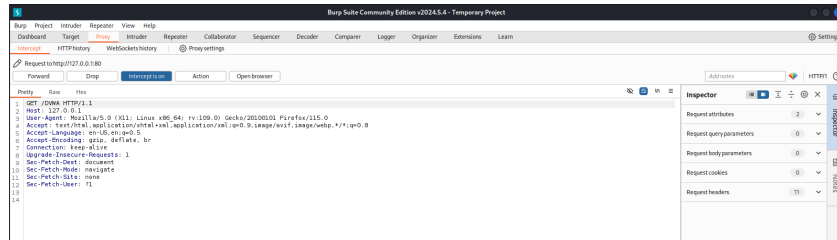


Рис. 3.7: Перехват запроса на вход на сайт

Рассмотрим перехват запроса аутентификации(рис. 3.8):

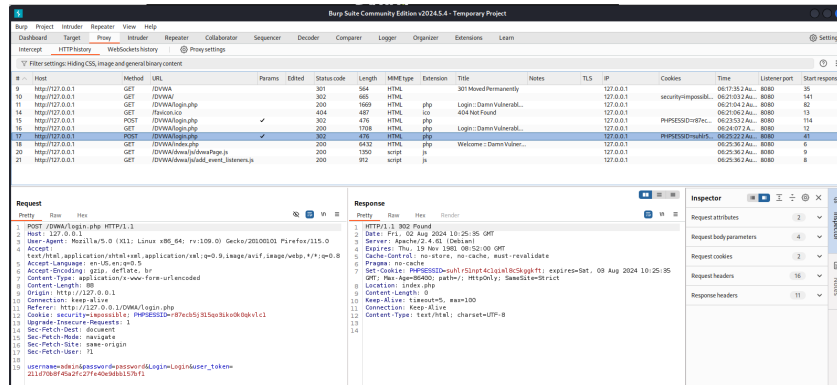


Рис. 3.8: Запрос на аутентификацию

Здесь дополнительно указываются куки запроса, а также выдается сам запрос с указанием введенного имени пользователя и пароля.

Кроме того уже совершенный запрос можно отпправить на повтор для того чтобы изучить ответы(рис. 3.9):

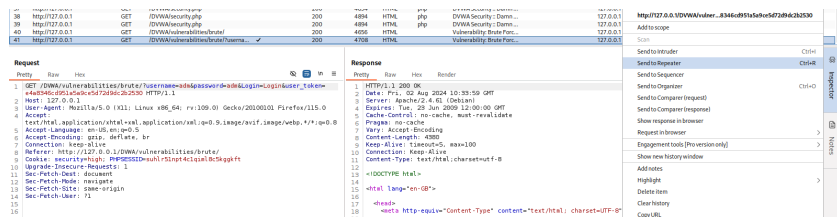


Рис. 3.9: Функция повторения запроса

В запросах можно изменять вводимую нформацию и сравнивать ответы(рис.

3.10):

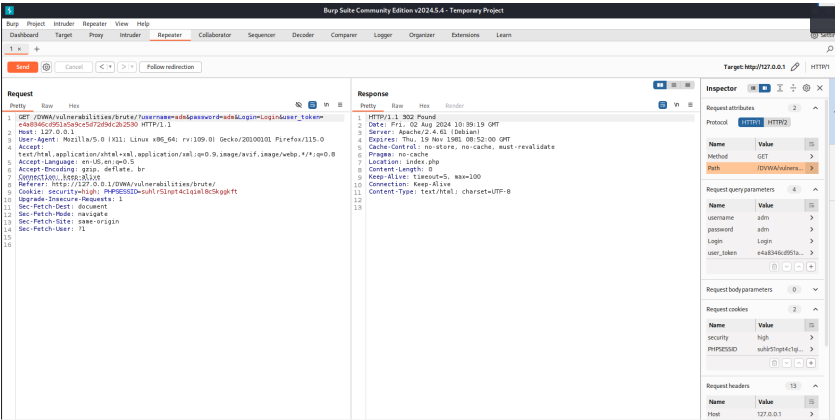


Рис. 3.10: Изучение ответа на запрос с функцией повторения запроса

4 Выводы

В результате выполнения работы научились на практике использовать ПО Burp Suit для перехвата, изменения и изучения HTTP запросов и ответов.

Список литературы

1. DVWA [Электронный ресурс]. GitHub, Inc, 2024. URL: <https://github.com/digininja/DVWA>.
2. Burp Suit [Электронный ресурс]. PortSwigger Ltd., 2024. URL: <https://portswigger.net/burp>.