**Руководство по выполнению Лабораторной работы № 1 “Анализ безопасности веб-приложений, разработанных на основе CMS платформы”**

**Цель работы**: Научиться использовать стандартное сканирующее ПО в целях проверки безопасности веб-приложения. Эмулировать настоящую атаку на веб-приложение.

**Ход работы:**

1. Проанализировать сайт с помощью утилиты whatweb
2. Просканировать сайт специальным сканером уязвимостей (в зависимости от типа CMS)
3. Провести автоматический перебор директорий с помощью dirb.
4. Провести ручной анализ сайта и найти контрольные значения
5. Произвести атаку полного перебора для пароля одного из пользователей (используя hydra, patator, Burp Suite или другое)
6. Произвести эксплуатацию известных уязвимостей и описать их

**Отчёт:** По каждому пункту составить подробное описание с изображениями (скриншотами) и свести всё в единый отчёт, стиль оформления отчёта – научно-технический.

**Краткая теория:** В данной краткой теории описаны методы использования инструментов, для более обширного описания данных инструментов вы можете обратиться к текстам лекций, либо к любым другим источникам.

**1. Проанализировать сайт с помощью утилиты whatweb.**

Данная утилита была подробна рассмотрена в одной из лекций про безопасность веб-приложений. Данный инструмент используется для автоматизирования сбора общей информации о тестируемой цели, может облегчить понимание функционирования веб-приложения и сформировать некоторые векторы атак.

Пример использования приведён на Рис. 1.

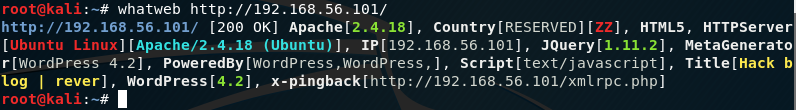


Рисунок 1 - Пример использования whatweb.

Исходя из полученных данных с помощью данной утилиты, необходимо сделать выводы об исследуемом приложении и записать их в отчёт по выполнению лабораторной работы.

**2. Просканировать сайт специальным сканером уязвимостей (в зависимости от типа CMS)**

Исходя из полученных данных при выполнение прошлого пункта, необходимо выбрать подходящий автоматический сканер уязвимостей для исследуемой цели. Автоматические сканеры уязвимостей также рассматривались в лекциях по безопасности веб-приложений. Например, для популярной CMS платформы – Wordpress существует открытый сканер wpscan, который входит в состав дистрибутива Kali Linux. Пример использования показан на Рис. 2.

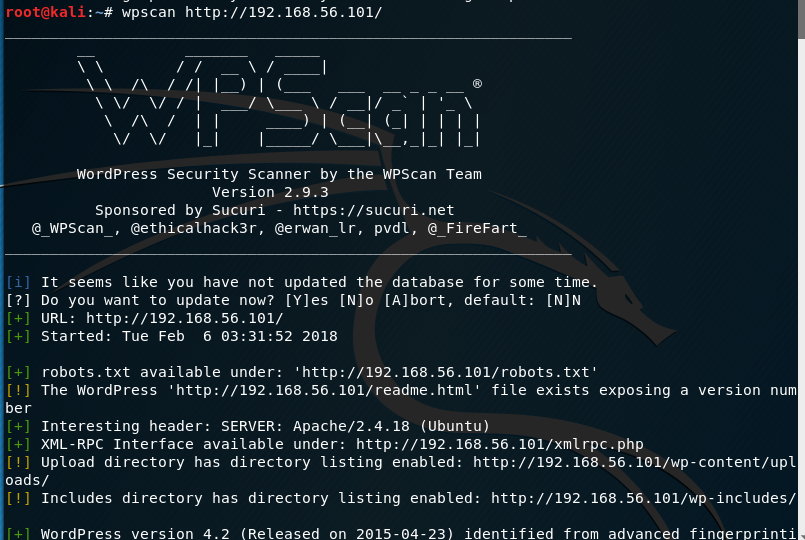


Рисунок 2 - Пример использования сканера wpscan.

Исходя из полученных данных с помощью данной утилиты, необходимо сделать выводы об исследуемом приложении и записать их в отчёт по выполнению лабораторной работы. Также необходимо указать общее количество уязвимостей и количество крайне критичных уязвимостей.

**3. Провести ручной анализ сайта и найти контрольные значения**

Выполнение данного пункта заключается в ручном анализе сайта и поиска на сайте контрольных значений, представленных в форме специального слова и заключённого в фигурные скобки контрольного значения в формате md5-хеша. Ручной анализ может заключаться в просмотре исходного кода страниц, анализа доступных страниц сайта, нахождение учётных записей пользователей. При проведении анализа необходимо обнаружить как можно больше контрольных значений и внести их в виде скриншотов и описания к ним, каким образом они были получены.

**4. Произвести атаку полного перебора для пароля одного из пользователей (используя hydra, patator, Burp Suite или другое).**

Выполнение данного пункта не обязательно должно заканчиваться компрометацией одного из аккаунтов веб-приложения, так как атаки полного перебора довольно часто занимают большой объём времени. Поэтому в данном пункте необходимо производить атаку по словарю, что ускорит работу атаки полного перебора.

Для простого перебора можно использовать уже применяемую нами ранее утилиту – wpscan которая также может производить перебор паролей. Стоит отметить, что для осуществления перебора необходимо знать имя учётной записи, что в прочем находится без особо труда применительно к Wordpress. Пример команды для атаки полного перебора приведён на Рис. 3.



Рисунок 3 - Запуск атаки полного перебора.

После выполнения своих стандартных функцию wpscan начнёт производить атаку полного перебора по заданному словарю.

Ключи, используемые при запуске:

-u – URL цели.

--threads – количество потоков для выполнения атаки

--wordlist – путь до словаря, файла с паролями

--username – имя пользователя для которого проводится атака.

В случае нахождения пароля будет выдано сообщение об успешном проведении атаки, а также в конце напечатана таблица, если вы проводили атаку для нескольких пользователей Рис. 4.

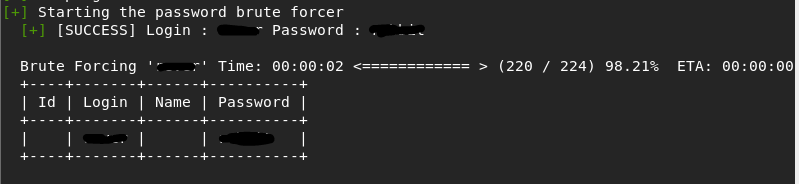


Рисунок 4 - Успешная реализация атаки.

При выполнение данного пункта необходимо провести атаку полного перебора с использованием предложенных инструментов (при использовании инструментов отличных от wpscan, даже при неудачном проведение атаки — это будет учитываться более весомо, чем удачная атака с помощью wpscan). В отчёт должны войти скриншоты запуска атаки и её результатов, а также выводы по проделанной атаке. Успешность атаки не считается важным критерием.

**5. Произвести эксплуатацию известных уязвимостей и описать их.**

При помощи данных полученных на этапе №2 лабораторной работы, необходимо произвести попытки эксплуатации одной из найденных уязвимостей.

При этом требуется описать данную уязвимость, отметить, существует ли для неё эксплоит в открытом доступе, что требуется для успешной реализации данной уязвимости, если реализовать её в данных условиях не представляется возможных – то пояснить принципиальные причины невозможности эксплуатации.

Все действия и анализ необходимо сопровождать скриншотами и логическими пояснениями. Выбор открытой уязвимости производится вами лично, однако он должен быть отличен от выбора вашего соседа.

**Оформление отчёта:** Отчёт должен содержать изображения и описание проведённых вами действий при исследовании приложения; выводы, производимые на различных этапах анализа, могут быть вынесены в финальные выводы, также финальные выводы должны содержать оценки безопасности тестируемого веб-приложения и рекомендации для устранения ошибок безопасности, если они имели место быть.