МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Анализ web-сервисов на наличие XSS-уязвимостей

АВТОРЕФЕРАТ

дипломной работы

студента 6 курса 631 группы

специальности 10.05.01 Компьютерная безопасность

факультета компьютерных наук и информационных технологий

Ковалева Александра Сергеевича

Научный руководитель

к. п. н, доцент _____ А. С. Гераськин

23.01.2021 г.

Заведующий кафедрой

д. ф.-м. н., доцент _____ М. Б. Абросимов

23.01.2021 г.

Саратов 2021

ВВЕДЕНИЕ

Обеспечение информационной безопасности вычислительных систем является одной из главных задач для каждой организации, в хозяйственной деятельности которой применяются алгоритмы сбора, обработки, хранения, передачи информации. Из-за распространения web-приложений стали возможны множество угроз информационной безопасности. 10 лет назад преобладали статические web-приложения, они не имели интерактивных интерфейсов взаимодействия с пользователями. Следовательно, почти не было уязвимостей, которые могли бы быть использованы нарушителями. Это позволяло разработчикам игнорировать вопросы, связанные с безопасностью. На данный момент, практически все web-сайты и приложения являются динамическими, в них огромное количество новых технологий, используемых web-браузерам. Новейшие технологии позволяют подключать к web-приложениям все возможные модули, которые позволяют посетителю по максимуму использовать web-ресурсы (например, доски объявлений, формы обратной связи и т.д.). Но, к сожалению, пользователь может столкнуться с проблемами. Технологии, функционирующие в динамических web-сайтах, обеспечивают хорошую платформу нарушителям для проведения XSS-атак. В 9 из 10 приложениях актуальна угроза атак на клиентов с помощью межсайтового скриптинга[1]. С помощью внедренного кода нарушитель может получить несанкционированный доступ к конфиденциальной информации пользователя и совершать противоправные действия, как на локальных компьютерах пользователей, так и в сетевом оборудовании компании, меняя

[1] Positive Technologies [Электронный ресурс]: Угрозы и уязвимости веб-приложений в 2019 году. URL: https://www.ptsecurity.com/ru-ru/research/analytics/web-vulnerabilities-2020/?sphrase_id=81237 (дата обращения: 13.11.2020). - Загл. с экрана. – Яз. рус.

конфигурацию сети и программного обеспечения[2]. Отсутствие должных мер по соблюдению правил и норм информационной безопасности приводит к появлению угроз, которые можно реализовать с помощью компьютерных атак, эксплуатирующих уязвимости, связанные с внедрением вредоносного кода.

Целью данной работы является разработка программного обеспечения для поиска XSS-уязвимостей. Для решения поставленной цели необходимо выполнить следующие задачи:

1. провести исследование видов XSS-уязвимостей;
2. исследовать алгоритмы поиска XSS-уязвимостей;
3. провести анализ существующих программных продуктов для поиска XSS-уязвимостей;
4. разработать программное обеспечение по поиску XSS-уязвимостей.

Дипломная работа состоит из введения, 4 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 53 страницы, из них 32 страницы – основное содержание, включая 22 рисунка, список использованных источников из 20 наименований.

[2] Носиров, З. А. Обнаружение XSS-уязвимостей на основе анализа полной карты веб-приложения: / З. А. Носиров, И. М. Ажмухамедов // Системы управления, связи и безопасности. 2018. №1. URL: http://sccs.intelgr.com/archive/2018-01/03-Nosirov.pdf (дата обращения: 13.11.2020). - Загл. с экрана. - Яз. рус.

КРАТКОЕ СОДЕРЖАНИЕ

Первый раздел дипломной работы посвящен исследованию видов XSS-уязвимостей. В разделе представлено определение XSS-уязвимостей и их виды; рассмотрены примеры XSS-уязвимостей каждого вида. Также в конце раздела представлено определение XSS-инъекции и описан список инъекций, которые использовались для тестирования разработанного программного обеспечения. В результате были выделены 3 вида XSS-уязвимостей:

1. непостоянные (отраженные) осуществляются, когда данные, предоставляемые web-клиентом, тут же используются серверными скриптами для генерации страницы с результатами для этого самого клиента. Если пользовательские данные некорректны и содержатся внутри страницы с результатами без кодирования HTML - это позволяет внедриться клиентскому коду в динамическую страницу. После этого внедренный код может быть выполнен на стороне сервера, например, на странице с поисковыми результатами или на странице с ошибкой или любой другой странице, которая появляется в ответ на запрос пользователя. Эта страница будет включать в себя часть входных данных, передаваемых серверу как часть запроса;
2. постоянные (хранимые) - в этом случае вредоносный код хранится на web-сервисе (в базе данных, файловой системе или в другом месте), а затем отображаются посетителю web-страницы без кодирования с использованием специальных символов HTML;
3. локальные XSS-уязвимости (основанные на DOM), заключаются в том, что злоумышленник меняет данные на стороне клиента во время запроса страницы с сервера. Если участок кода JavaScript имеет доступ к параметру URL и формирует некий HTML-код на странице, используя эту информацию, которая не закодирована с использованием спецсимволов HTML – возможно, присутствует XSS-уязвимость, поскольку записанные данные будут заново

интерпретированы браузерами, так как HTML-код может содержать дополнительный клиентский скрипт. Единственным отличием уязвимости модели DOM от других типов является то, что сервер не возвращает результатов запроса; наоборот, происходит локальная обработка данных при помощи функций DOM и вредоносный сценарий исполняется с такими же правами, что и web-браузер на машине жертвы атаки.

Второй раздел содержит в себе описание алгоритмов поиска для каждого из видов XSS-уязвимостей. Алгоритмы, приведенные в этом разделе были использованы для разработки программного обеспечения для поиска XSS-уязвимостей, описанного в четвертом разделе.

Для поиска непостоянных уязвимостей, необходимо провоцировать отправку данных, включая в отправляемые данные вредоносный код и получая ответ в HTML формате. Пришедшее сообщение затем анализируется на наличие уязвимости следующим образом: если страница содержит ожидаемый от инъекции результат, то страница помечается как содержащая потенциальную угрозу соответствующего типа, иначе страница помечается как безопасная и не добавляется в итоговый список уязвимостей.

Алгоритм поиска хранимых уязвимостей во многом схож с алгоритмом поиска отраженных XSS-уязвимостей, за исключением того, что необходимо производить отправку формы и ждать ответа от сервера. В случае, если скрипт выполнился, страница помечается как уязвимая.

В ходе выполнения алгоритма поиска уязвимостей основанных на DOM осуществляется анализ кода страницы на наличие скриптов, спрятанных в HTML тегах. После нахождения содержимого всех скриптов на странице в найденных данных осуществляется поиск вызовов методов объектной модели документа таких как: запись чистого HTML; прямая модификация модели документа (в том числе события Dynamic HTML); прямое выполнение скриптов.

Третий раздел содержит в себе описание программных продуктов для поиска XSS-уязвимостей, существующих на рынке на данный момент. В данном разделе рассмотрено 5 программ, позволяющих анализировать web-сервисы на наличие уязвимостей: NetSparker, Wapiti, Tenable.io, OWASP ZAP и Burp Suite.

Среди рассмотренного программного обеспечения можно выделить OWASP ZAP. Данное приложение позволяет просканировать ресурс на наличие не только XSS-уязвимостей, выдает предложения по решению найденных уязвимостей и распространяется бесплатно. Из недостатков данного приложения можно выделить сложность в настройке сканирования и неполную локализацию. Сложность в настройке не позволяет обычному пользователю быстро просканировать ресурс на наличие уязвимостей и устранить их.

По результатам исследования были выявлены следующие недостатки существующего на рынке программного обеспечения:

- не все ПО осуществляет поиск XSS в закрытой части web-ресурса;
- нахождение не всех XSS-уязвимостей в WEB-ресурсе;
- высокая стоимость программных продуктов;
- не все ПО формирует удобные для чтения отчеты;
- не у всех приложений есть удобный графический интерфейс.

Четвертый раздел содержит описание и тестирование разработанного приложения для поиска XSS-уязвимостей на web-ресурсе. Программа написана на языке Python с использованием дополнительных библиотек Mechanize и PySimpleGUI. Библиотека Mechanize позволяет имитировать работу браузера и позволяет с помощью функций выполнять действия на web-ресурсе. Библиотека PySimpleGUI использовалась для разработки интерфейса приложения. Для формирования отчетов в виде Excel файлов использовалась библиотека xlswriter.

Для тестирования программы, было использовано приложение bWAPP – это PHP приложение, использующее базу данных MySQL, разработанное с

уязвимостями, для тестирования различных видов атак. Это приложение запускалось на виртуальной машине с операционной системой Linux.

Также в этом разделе представлено описание алгоритма работы программы. В начале работы программы составляется карта сайта. После составления карты сайта для каждой найденой страницы осуществляется 3 вида анализа страницы: путем поиска и отправки форм, путем инъекции в URL параметры, путем инъекции в HTML-заголовки. На основе выявленных уязвимостей составляется отчет, который отображается в интерфейсе приложения в поле «Отчет», а также сохраняется в указанной директории, если такой параметр был выбран. Отчет генерируется в формате Excel и содержит в себе общее количество найденных уязвимостей, дату, время выполнения инъекции, URL адрес уязвимой страницы, тип элемента, в который производилась инъекция и сама инъекция.

Далее приводятся примеры работы программы для нескольких уязвимых страниц. Тестируются инъекции в формы ввода данных, в HTTP-заголовки и в URL-параметры.

При тестировании страницы с уязвимой формой ввода данных, будут производиться инъекции в HTTP-заголовок «Referer», URL-параметры «firstname», «lastname» и в оба поля ввода. Для тестирования использовалась инъекция <script>alert(“1”)</script>. Результат исследования страницы представлен в поле отчет. В отчете видно, что была найдена уязвимость путем загрузки инъекции. В сгенерированном отчете можно увидеть, что уязвимость найдена именно в форме ввода данных.

При тестировании страницы с уязвимостью в использовании HTTP-заголовка будет использована инъекция <script>alert(“Injection”)</script>. В сгенерированном отчете видно, что уязвимость найдена именно путем внедрения инъекции в HTTP-заголовок «Referer».

При тестировании страницы, в которой при заполнении формы или передаче URL-параметров «firstname» и «lastname» генерируется страница с

введенными данными, используется инъекция %3Cscript%3Ealert%28%221%22%29%3C%2Fscript%3E. В сгенерированном отчете видно, что было найдено две уязвимости именно путем внедрения инъекций в URL-параметры «firstanme» и «lastname».

В результате тестирования программы были выделены определенные преимущества. Главным преимуществом программы является гибкость в настройках сканирования, возможность исследования страниц, для доступа к которым пользователь должен пройти авторизацию, а также возможность генерации отчетов с результатами исследования страниц. Также можно отметить, что большую часть времени работы программы занимает ожидание загрузки страниц, выполнение инъекций происходит быстро.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были установлены виды XSS-уязвимостей. Этими видами являются отраженные, хранимые и основанные на DOM XSS-уязвимости. Также были рассмотрены алгоритмы поиска этих уязвимостей, на основе которых было разработано программное обеспечение для поиска XSS-уязвимостей.

Также было рассмотрены существующие на рынке программные продукты. Более детально рассматривались программы NetSparker, Wapiti, Tenable.io, OWASP ZAP и Burp Suite. Среди рассмотренного программного обеспечения можно выделить OWASP ZAP. Данное приложение позволяет просканировать ресурс на наличие не только XSS-уязвимостей, выдает предложения по решению найденных уязвимостей и распространяется бесплатно. Из недостатков данного приложения можно выделить сложность в настройке сканирования и неполную локализацию. Сложность в настройке не позволяет обычному пользователю быстро просканировать ресурс на наличие уязвимостей и устранить их.

Также было разработано собственное приложение для поиска XSS-уязвимостей. Разработанное ПО позволяет производить поиск XSS-уязвимостей на сайтах производя инъекции в формы ввода данных, URL параметры и в HTTP заголовки запросов. ПО тестировалось на специальном приложении, разработанном с уязвимостями, были рассмотрены все возможные сценарии сканирования web-ресурса на наличие XSS-уязвимостей. Разработанное приложение отлично справилось с поиском уязвимостей на тестовом приложении.

Разработанное приложение может быть использовано собственниками web-ресурсов для поиска XSS-уязвимостей, чтобы возможность внедрения инъекций и нанесения ущерба посетителям своего web-ресурса.

Разработанное программное обеспечение имеет свои преимущества. Преимуществом является гибкость в настройках сканирования, простой

интерфейс, составление отчетов о найденных уязвимостях и возможность анализа страниц, доступных только авторизованным пользователям.