Лабораторная работа № 4

Тестирование веб-сайтов

В этой работе Вы должны познакомиться с технологиями тестирования на проникновение, инструментами для проведения такого тестирования, рекомендациями по разработке надёжных веб-приложений и веб-сервисов.

Проект OWASP

Большая часть современных приложений используют сетевые технологии и по сути являются или веб-приложениями (динамические сайты) или веб-сервисами. Для разработчиков таких приложений одним из наиболее значимых вопросов является безопасность как всей инфраструктуры (обычно обеспечивается хостинг-провайдером), так и скриптов (обеспечивается разработчиком).

Проблема осложняется тем, что для написания сайтов широко используются готовые решения в виде различных фреймворков и CMS, где уровень безопасности поставляется «из коробки», и повлиять на него очень проблематично. Стоит отметить, что каждый этап рефакторинга, добавления/изменения функционала на сайте зачастую снижает уровень безопасности. Если на этапе запуска проекта уровень защиты был протестирован и находился на допустимом уровне, то впоследствии могут появиться критические уязвимости, обнаруживающие себя слишком поздно.

Список возможных уязвимостей веб-приложений и веб-сервисов весьма обширен, так как сайты и сервисы используют очень разнообразный стек технологий, который, в свою очередь, развивается достаточно интенсивно. Ограниченная версия этого списка выглядит примерно так:

- PHP-инъекция;
- PHP-инъекция через загрузку файлов;
- SQL-инъекция;
- межсайтовый скриптинг (XSS);
- взлом сервисов аутентификации и управления сеансами;
- небезопасные прямые ссылки на объект;
- некорректная конфигурация безопасности инфраструктуры;
- доступность конфиденциальных данных;
- отсутствие контроля доступа на уровне функции;
- межсайтовая подделка запроса (CSRF);
- использование компонентов с известными уязвимостями;
- непроверенные перенаправления и переадресация;
- демонстрация ошибок пользователю;
- доступность данных о характеристиках системы пользователю;
- доступность данных о программном коде пользователю;
- возможность задания глобальных переменных;

С 2001 года существует открытый проект обеспечения безопасности веб-приложений (Open Web Application Security Project) —OWASP. Сообщество OWASP включает в себя корпорации, образовательные организации и частных лиц со всего мира, OWASP состоит примерно из 190 местных отделений, располагающихся по всему миру (Российское представительство OWASP находится здесь: https://www.owasp.org/index.php/Russia) и тысяч участников в листах рассылки проекта. Сообщество работает над созданием статей, учебных пособий, документации, инструментов и технологий, находящихся в свободном доступе.

OWASP не аффилирован ни с одной компанией, занимающейся разработкой технологий, он поддерживает грамотное использование технологий безопасности. Проект избегает аффилирования, так как полагает, что свобода от влияния со стороны других организаций может облегчить распространение беспристрастной, полезной и дешевой информации о безопасности приложений.

Участники сообщества OWASP делают приложения безопаснее, учитывая человеческий фактор и технологический уровень. Наиболее востребованные документы, опубликованные OWASP, включают в себя:

Руководство OWASP — www.owasp.org/index.php/OWASP_Guide_Project;

Обзорное руководство по программированию — www.owasp.org/index.php/Category:OWASP_Code_Review_Pro

Руководство по тестированию OWASP — https://www.owasp.org/index.php/OWASP_Testing_Project;

Проект Топ-10 OWASP — www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

Руководство по Разработке OWASP даёт практические советы и содержит примеры кода на J2EE, ASP.NET и PHP. Руководство по Разработке охватывает обширный массив вопросов безопасности для уровня приложений, таких как SQL-инъекции, фишинг, обработка кредитных карт, фиксация сессий, подделка межсайтовых запросов, согласование и конфиденциальность.

OWASP создал стандарт OWASP Application Security Verification Standard (ASVS). Основная цель OWASP ASVS — это стандартизация диапазона охвата и уровня строгости доступных на рынке приложений, обеспечивающих безопасность, а также создание набора коммерчески успешных открытых стандартов, приспособленных для специализированных веб-технологий. Сборник для веб-приложений уже опубликован, сборник для веб-сервисов находится в процессе разработки.

Самыми распространёнными инструментами OWASP являются тренировочная среда WebGoat (https://www.owasp.o прокси-анализатор Zed Attack Proxy (https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project) и .NET инструменты (https://www.owasp.org/index.php/Category:OWASP_.NET_Project).

Задание 1. Использование IBM Security AppScan Standard для сканирования веб-приложений на наличие уязвимостей безопасности

Доступная версия «IBM Security AppScan Standard» позволяет сканировать только сайт `demo.testfire.net`. Для этого сайта есть доступ в один аккаунт — имя пользователя: `jsmith` пароль: `demo1234`.

Сканирование «AppScan» состоит из двух этапов: анализ и тестирование.

Анализ — сайту отправляются запросы для сбора данных о нем и его структуре в автоматическом режиме или вручную. AppScan анализирует ответы, выполняет поиск потенциальных уязвимостей и создает тестовые запросы.

Тестирование — AppScan отправляет тысячи пользовательских тестовых запросов. Он записывает и анализирует ответы приложения, обнаруживая неполадки защиты, определяя уровень риска и предлагая рекомендации.

В ходе полностью автоматического сканирования после завершения первой фазы анализа и тестирования AppScan переходит к новой фазе для обработки информации, полученной в ходе тестирования. Сканирование завершается после выполнения настроенного числа фаз сканирования.

Анализ веб-приложения или веб-службы перед проверкой с помощью AppScan можно выполнить тремя способами:

- С помощью AppScan: продукту AppScan передаются начальный URL и идентификационные данные для автоматического анализа. Кроме того, можно вручную проверить сайт, чтобы предоставить AppScan доступ к областям, для обращения к которым требуются специальные действия пользователя.
- С помощью внешнего устройства для анализа веб-служб RESTful или других веб-служб, отличных от SOAP или служб SOAP, не требующих конвертов защиты — запросы отправляются на сайт с помощью мобильного телефона, симулятора или эмулятора; AppScan настраивается как записывающий прокси-сервер.
- С помощью GSC (Generic Service Client, общий клиент служб): при наличии файла WSDL интегрированный клиент (GSC) может создать интерфейс, в котором отображаются службы и можно ввести параметры и просмотреть результаты.

Результаты сканирования удобно представить в виде отчёта. В «AppScan» можно управлять содержимым и макетом отчётов, доступны отчёты различных типов:

Отчёт о защите — перечисленыhtpasswd -c .htpasswd captain обнаруженные уязвимости защиты.

Промышленный стандарт — указывает, соответствует ли приложение требованиям выбранных стандартов (PCI, OWASP Top 10, SANS или WASC).

Соблюдение требований законодательства — указывает, соответствует ли приложение требованиям законодательства (например, HIPAA, GLBA, SOX, California SB 1386 и AB 1950).

Разностный анализ — сравнивает два набора результатов сканирования и отображает различия URL и/или неполадок защиты.

Шаблоны отчётов — позволяет создать отчёты с пользовательскими данными и параметрами форматирования в виде файлов MS Word (.doc).

1. Зайдите на сайт `demo.testfire.net`. Посмотрите разделы сайта. Попробуйте зайти в аккаунт `jsmith` (пароль: `demo1234`). Посмотрите доступные для этого пользователя ресурсы.
2. Запустите «IBM Security AppScan Standard». Закройте окно приветствия.
3. Создайте новое сканирование (▷ Файл ▷ Создать...).
4. Выберите шаблон обычного сканирования (Regular Scan).
5. В мастере настройки сканирования выберите метод анализа AppScan.
6. Настройте сканирование с помощью AppScan: начальный URL сканирования: `https://demo.testfire.net`.
7. Если работа выполняется в компьютерном классе университета, то необходимо настроить дополнительные параметры связи (поставить флажок в нижней части окна). На следующем шаге необходимо выбрать пользовательские настройки прокси-сервера, адрес:
   - в классах МИЭМИС: `proxy-sc.asu.ru`;
   - в остальных классах АлтГУ: `proxy.asu.ru` или `proxy-class.asu.ru` (можно посмотреть с помощью `netstat`).

   порт: 3168. Для прокси обязательно надо настроить имя пользователя, пароль (указав свои значения) и домен `stud`. Вернитесь на шаг назад и проверьте соединение с сервером (кнопка ).
8. Далее, выберите способ входа на сайт Автоматический и задайте параметры авторизации для входа на сайт:
   - логин: `jsmith`;
   - пароль: `demo1234`;
9. Выберите из шаблонов стратегию тестирования — Полный.
10. Далее, выберите запуск полностью автоматического сканирования.
11. Задайте файл для сохранения результатов сканирования. После этого должно запуститься сканирование.

    Сначала будет выполнено оценочное сканирование, после анализа его результатов будут выданы рекомендации по изменению параметров для оптимизации стратегии сканирования веб-сайта.
12. Выполните выданные рекомендации (измените параметры, настройки сканирования). Некоторые параметры среды исполнения веб-приложения можно выяснить изучая сайт, используя обычный браузер.

    Полный доступ к параметрам **конфигурации сканирования** можно получить через соответствующую пиктограмму или меню ▷ Сканирование ▶ Конфигурация сканирования (F10).

    Чем точнее Вы укажете известные данные, тем быстрее будет произведено тестирование и анализ сайта.

13. Просмотрите в разделе ТЕСТ выбранные Стратегии тестирования (кнопкой [+] раскрываются группы). Здесь можно узнать о сути тестируемой уязвимости (на вкладке Сообщение) и Рекомендации по исправлению этой уязвимости.

14. Примените рекомендации (помеченные). При этом запустится повторное сканирование, более глубокое, для полного экспертного анализа.

    В процессе тестирования возможны обнаружения уязвимостей, они будут скапливаться в разделе Неполадки (▷ Вид ▶ Защита). По умолчанию список Упорядочен по приоритету, По убыванию.

    Во время сканирования можно просмотреть найденные уязвимости (Информация о неполадке), их описание Сообщение) и рекомендации для их устранения (Рекомендации по исправлению).

    Во вкладке Запрос/Ответ можно просматривать описание теста, посылаемый запрос (начинается с `GET`) и ответ сервера (начинается с `HTTP`). Здесь можно выполнять поиск текста, открывать запрос в браузере. Через локальное меню ▷ Опции можно задать параметры выбранному тесту.

15. В разделе Данные (▷ Вид ▶ Данные приложения) можно просмотреть сгенерированные для тестирования данные (параметры, запросы, и т. д.). Обратите особое внимание на вкладку Требуется взаимодействие с пользователем, если в этом разделе есть данные, необходимо вручную ввести необходимые значения.

16. В разделе Задачи (▷ Вид ▶ Задачи исправления) можно посмотреть выданные рекомендации для устранения обнаруженных уязвимостей. В крайнем левом столбце представлена в виде дерева файловой системы логического каталога сайта (На основе URL) или связанного списка файлов физического каталога сайта (На основе содержимого). Выбирая ветки этого дерева можно видеть отфильтрованные рекомендации для выбранной ветки (папки или файла).

17. После окончания сканирования, просмотрите Протокол сканирования (▷ Вид ▶ Протокол сканирования).

18. Откройте в разделе XSS первый тест (`amCreditOffer (Cookie)`), прочитайте о нём информацию, просмотрите запрос этого теста в браузере, пометьте этот тест, как не имеющий уязвимостей.

19. Откройте мастер создания отчётов, добавьте в Макет верхний колонтитул со своей фамилией, инициалами и номером группы. Создайте 3 отчёта:

    - Защита — используйте шаблон Подробный отчёт, добавьте в содержание отчёта Запрос/Ответ.
    - Соответствие промышленным стандартам — *OWASP Top 10 2013*.
    - Соблюдение требований законодательства — выберите *Payment Application Data Security Standart*.

Задание 2. Сканирование сайта свободными инструментами

Сканирование произвольного сайта может повлечь административную и даже уголовную ответственность. В этом задании Вам предлагается протестировать специально созданный компанией ADN Digital Studio сайт hackme.adn.agency.

1. Скачайте виртуальную машину с *Kali Linux* с ftp://10.0.12.224/. Запустите Kali.

2. Изучите тестируемый сайт (как обычный пользователь, через браузер).

3. Просмотрите доступные установленные инструменты (через меню). Запустите «Sparta».

4. Задайте параметры прокси, как в задании 4.

5. Добавьте в «Sparta» IP-адрес тестируемого ресурса. Выполните тестирование.

6. Укажите в отчёте, какие службы и по каким портам доступны. Подкрепите ответ скриншотом служб.

7. Добавьте в отчёт протоколы работы используемых внешних инструментов.

8. Скачайте (в Kali) «Словари для брута» с ресурса ftp://10.0.12.224/ (можете найти в Сети и другие словари).

   Выполните подбор идентификационных данных для тех сервисов, для которых требуется аутентификация (отправьте их в Brute). Для этого могут пригодиться словари (загрузите их в «Sparta» в соответствующие разделы).

   Добавьте в отчёт протоколы взлома (Brute).

9. Какую ещё информацию об этом сайте Вы можете добавить? Можете использовать другие свободные инструменты. Добавьте эту дополнительную информацию в отчёт.