Лабораторная работа №6

БАЗОВЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ СТРУКТУРЫ WEB ПРИЛОЖЕНИЙ

1. Цель и задачи работы

Ознакомится с основными методами анализа структуры WEB приложений, и методами оценки результатов анализа.

2. Теоретические положения

На первом этапе развития WEB предполагалось, что все необходимый для работы WEB приложения файлы, должны располагаться в одном каталоге, расположенном на сервере.

Данный каталог получил название «виртуальный каталог», а процесс размещения приложения(файлов необходимых для работы WEB приложений) «публикация приложения».

При этом, не смотря на развитие WEB приложений, данная базовая особенность их размещения осталось неизменной.

При этом приложение, занимающиеся обработкой запросов клиентов и управляющее доступом к файлам виртуального каталога, получило название WEB сервис.

Основными функциями Web сервиса являются.

1. Регистрация места расположения виртуального каталога в файловой системе ОС, на которой развёрнут сервер. Чаще всего оно задано в настройках сервера и настройках каталога.
2. Обслуживание каталога и управление привилегиями доступа к файлам и подкаталогам, расположенных в виртуальном каталоге приложений.
3. Трансляция URL адресов, используемых браузером в пути к файлам, используемыми файловой системой.
4. Назначение процессов обработчиков, отвечающих за обслуживание виртуального каталога и настройки пула потоков, созданных для обслуживания данного приложения.
5. Обработка заголовков протоколов Http и Https и преобразование переданными с их помощью данных, в вид удобный для работы WEB приложения. По сути сделать так, чтобы все полученные данные могли бы использоваться в WEB, написанном на языке программирования, который поддерживает сервер.
6. Трансляция и отображение ошибок, возникающих при обработке запросов WEB приложений.
7. Администрирование самого WEB сервиса.
8. Прочие функции, связанные с поддержкой работы WEB приложений. Например, перезапуск пула потоков, обслуживающих приложение. Рестарт приложения, ограничение максимального количества запросов, автоматическое перенаправление запросов, при необходимости,

организация ftp сервера, базовая настройка фильтров, поддерживаемых сервером.

Наибольшее распространение получили следующие WEB серверы.

1. Apache. Языки программирования PHP, Python.
2. IIS (Microsoft). ASP.NET, ASP.NET MVC /C#. Прочие языки программирования, поддержка которых может быть добавлена при помощи дополнений, например, для языка PHP.
3. Kestrel (Microsoft/Linux). ASP.NET Core, ASP.NET MVC Core.
4. OWIN. ASP.NET Core, ASP.NET MVC Core.
5. Apache Tomcat(Servlet API). Java.

Как правило, большинство из данных серверов имеют каталоги По-умолчанию, в которых могут быть расположен виртуальные каталоги WEB приложений. Но крайне часто, администраторы настраивают иные каталоги расположения виртуальных каталогов, но при этом каталоги по умолчанию могут сохраняться.

Наличие WEB серверов, позволило упростить и немного упростить и частично стандартизировать процесс разработки приложений, за счёт создания различных библиотек-наборов модулей, используемых для построения приложений, получивших название фреймворки.

Дальнейшее развитие фреймворков привело к появлению стандартных структур построения приложений, которые отличаются в зависимости от использования различного языка программирования и фреймворка построения приложений.

На основе данных стандартных структур построения приложений, в дальнейшем были созданы готовые приложения, которые при помощи, более меньшего объёма кода, или вообще без написания кода, позволяют создать типовое, или близкое к типовому приложению и получили называние CMS(система управления контентом).

При этом, в случае, если необходимая функциональность приложения не может быть реализована на CMS, то как правило, разрабатывается собственная CMS, построенная на основе конкретного фреймворка.

Основные преимущества использования CMS.

1. Упрощение процесса администрирования WEB приложения. Не требуется использоваться услугами программиста для изменения содержимого страниц сайта(контентной части сайта).
2. Упрощение процесса построения защиты сайта. Частично обеспечение безопасности WEB приложений переложено на CMS и частично на программиста, в случае использование заведомо не безопасного кода, или отключение модулей защиты, используемых в CMS, вероятность отражения атаки будет стремиться к 0.
3. Ускорение времени разработки WEB приложений. По сути основной функционал WEB приложения уже реализован в CMS, но для обеспечения потребностей объекта информатизации, функционал CMS подвергается доработкам.

4. Упрощение процесса обновления приложений. По сути обновление в большинстве случаев будет сводиться к обновлению версии CMS. Но в случае, наличия существенных доработок, возможно, потребуется участие разработчика в процессе обновлении CMS.

Вне зависимости от используемой CMS, часто возникает задача индексирования сайта поисковой системой, такой как Яндекс, Google и др, для решение которой, как правило требуется разрабатывать сайт на основе структуры, совместимой с поисковой машиной. Например, обязательное наличие карты сайта, файл sitemap.xml, наличие файла управляющего процессом индексации страниц и уровнем доступа к ним, robots.txt и другие требования, которые, как правило, размещены на сайте поисковой системы.

Наличие определённых стандартов при написании WEB приложений, приводит к наличию возможности автоматизированного анализа состава файлов виртуального каталога WEB приложения.

При этом в виртуальном каталоге WEB приложения, могут быть размещены следующие типы файлов, представляющих интерес.

1. Статические файлы.
    a. Файлы таблиц стилей CSS.
    b. Файлы, в которых расположены клиентские скрипты, необходимые для работы приложений.
    c. Каталоги и файлы картинок, используемых для вёрстки сайтов и отображения контента.
    d. Видеофайлы, используемые для отображений на страницах сайта.
2. Файлы настройки CMS, или фреймворка, например, .htaccess, web.config, app.config и др. Крайне важны, так как могут дать информацию о логинах и паролях для сайта.
3. Файлы настройки сред развёртывания WEB приложений. Возможно окажется путь к репозиторию Git в котором расположен исходных код сайта.
4. Забытая служебная информация, не индексированная поисковой системой. Например, предыдущая версия файла конфигурации развёртывания, бакапы баз данных, забытые, или недоделанные страницы сайта.
5. Информация о страницах сайта. Файлы с расширением, характерным для конкретного языка программирования и фреймворка. Например, *.php, *.html, *.aspx, *.aspx.cs, *.cs, *.sql(срипты для работы с БД), и др.
6. Другая информация, например, файлы формата *.xlsx, в которых хранятся отчёты и прочая служебная информация. Файлы логов и.т.д.

Методы анализа содержимого виртуальных каталогов. Как правило методика анализа сводится к банальному перебору по словарю стандартных страниц и адресов сайта и в зависимости от кода ошибки

Http принятия решения о наличии, или отсутствии данного подкаталога на сайте.

При этом наиболее важными для обеспечения корректности данного процесса являются следующие ошибки HTTP.

1. 404 – данная страница, или раздел, отсутствует.
2. 403 – доступ к данному каталогу ограничен. Каталог, или страница существует на сайте, но средство анализа не смогло получить доступ к ней.
3. 500 – ошибка выполнения Http запроса. Страница, или раздел существует, но при его работе возникает ошибка. Как правило крайне интересная страница для анализа. В случае, если ошибки обработаны не корректно, может многое сказать о структуре построения сайта и его уязвимостях.
4. 301 -303 – перенаправления. По сути WEB сервер, или приложение, настроено таким образом, чтобы перенаправлять запросы, поступившие по данному адресу на другой адрес, чаще всего на страницы ошибок, или новую версию сайта.
5. 200 – не ошибка. Просто запрос был выполнено успешно, раздел существует и был обнаружен средством анализа.

При этом в зависимости от используемого средства анализа структуры WEB приложение, возможно использование следующих фильтров.

1. Адрес цели. Адрес WEB приложения, подлежащего анализу.
2. Фильтр по расширению файла.
3. Фильтр по времени выполнения запроса.
4. Фильтр по словарю. Быстрый, средний, полный.
5. Возможность рекурсивной фильтрации. По сути, если средство анализа обнаруживает подкаталог, оно автоматически просматривает его содержимое и анализирует его.

В зависимости от предпочтений аналитика и требуемой производительности. Наибольшую популярность получили следующие средства.

1. DirBuster https://kmb.cybber.ru/web/owasp_db/main.html или https://spy-soft.net/dirbuster-linux-windows/. Встроен в ОС Kali Linux, имеет графический интерфейс, относительно удобен в использовании, поддерживает несколько вариантов словарей, но, к сожаления, в некоторых случаях имеет недостаточную производительность.
2. DirSearch, упрощённый аналог DirBuster обладающий консольным интерфейсом и немного улучшенной производительностью https://vk.com/@thntofff-skryt-ne-poluchitsya-nahodim-skrytye-veb-direktorii-na-serve .
3. GoBuster, аналог DirBuster и DirSearch, написанный на языке программирования Go, с целью улучшения производительности https://itsecforu.ru/2019/08/09/%F0%9F%94%8E-сканирование-сайтов-на-наличие-инт/ .

Как правило, для сканирования сайта, при помощи любого из выбранных средств желательно придерживаться следующей последовательности действий.

1. Выбор средства анализ, исходя из личных предпочтений, или задания на выполнение работы.
2. Ввод корневого URL сайта без дополнительных путей, в случае необходимости полного анализа сайта в качестве цели анализа.
3. Настройка опций запуска приложения. Фильтр по расширению файлов.
4. Запустить средство сканирования.
5. Сохранить отчёт в требуемом формате.

3. Оборудование

Персональный компьютер с количеством процессорных ядер не менее 2, работающих на частоте не менее 2 GHz, работающий под управлением операционной системы Kali Linux, Ubuntu Linux с пакетом дополнений Forensic Tools, наличие установленных средств анализа, Windows 7, или более новая. Видеокарта с поддержкой технологий CUDA или Open CL. Не менее 4 GB оперативной памяти. Не менее 20GB свободного места на HDD.

4. Задание на работу

4.1 Установить Metasploitable 2 в комплекте с платформой DVWA.

4.2 Изучить документацию по основам работы с Metasplitable 2. https://docs.rapid7.com/metasploit/metasploitable-2/

4.3 Изучить методику исправления конфигурации приложения Mutillidae для работы с приложением Burp Suite https://cyberarms.wordpress.com/2015/05/01/mutillidae-database-errors-in-metasploitable-2/

4.4 Выполните привязку двух машин (Kali Linux и Metasploitable). Чтобы узнать адрес машины Metasploitable, используйте команду ifconfig. После проверьте работоспособность схемы с помощью команды: ping IP-адрес Metasploitable (по умолчанию ping 10.7.7.7).

4.5 Зайдите через браузер Kali Linux на Metasploitable.

4.6 Введите в строку запросов IP-адрес машины, на которую установлен Metasploitable.

4.7 Выбрать тренировочную площадку DVWA.

4.8 Просканировать площадку DVWA с помощью DirBuster и представить скриншот и отчёт.

4.9 Просканировать DVWA при помощи DirSearch предствить скриншот и отчёт.

4.10 Просканировать DVWA в рекурсивном режиме с анализом подкаталогов и предоставить скриншот и отчёт.

4.11 В качестве дополнительного задания просканировать сайт scanme.org

5. Порядок выполнения работы

5.1 Установить Kali Linux.

5.2 Настройте Burp Suite.

5.3 Установите Metasploitable 2.

5.4 Выполнить задания на работу в предложенном порядке.

5.5 Предоставить скриншот каждого этапа выполнения задания.

5.6 Сделать выводы о проделанной работе.

6. Оформление отчета

Отчет оформляется в тетради или листах формата А4 и должен содержать:

- название курса, название и номер лабораторной работы;
- цель работы и задание на исследование;
- базовая информация о структуре WEB приложения DVWA;
- скриншоты запросов определения структуры WEB приложения DVWA;
- краткие выводы о проделанной работе.

7. Контрольные вопросы

7.1 Типовые структуры построения WEB приложений?

7.2 Основные коды ошибок HTTP, используемые для поиска информации?

7.3 Методы анализа структуры WEB приложений, включая поиск скрытых файлов?

7.4 Состав и основные возможности приложения DirSearch?

7.5 Состав и основные возможности приложения DirBuster?

7.6 Состав и основные возможности приложения GoBuster?

7.7 Результаты сравнения данных средств на предмет скорости работы и удобство использования?

7.8 Правовая база использования подобных средств? **Напоминаю, использование данных средств без разрешения представителя объекта информатизации недопустимо, так как в некоторых случаях их использование может привести к «падению» WEB приложения, а так же нарушает доступность информации и нагружает северы, обслуживающие WEB приложение, что может быть расценено как целенаправленная атака.**

7.9 Какая юридическая ответственность наступит, или может наступить в случае анализа уязвимостей на сайте https://tulsu.ru/? **!!!Внимание!!!** Сканировать любые сайты, кроме настроенных в рамках данной работы, категорически запрещается!