**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

Институт компьютерных технологий и защиты информации

### КАФЕДРА СИСТЕМ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

### к выполнению Лабораторной работы № 1

на тему

**«**Анализ уязвимостей веб-ресурсов, сетевых угроз**»**

Казань 2021

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**Название работы:**

Анализ уязвимостей веб-ресурсов, сетевых угроз

**Цель работы:**

Ознакомиться с принципами работы сканеров обнаружения уязвимостей веб-ресурса веб-ресурсов.

Теоретический материал:

**ОБЩИЕ СВЕДЕНИЯ**

1. **Уязвимость -** недостаток в системе, используя который, можно нарушить её целостность и вызвать неправильную работу.

Уязвимость может быть результатом ошибок программирования, недостатков, допущенных при проектировании системы, ненадежных паролей, вирусов и других вредоносных программ, недостаточной проверки данных, вводимых пользователем, что позволяет вставить в интерпретируемый код произвольные команды (SQL-инъекция).



1) SQL-инъекции – встраивание вредоносного кода в запросы к базе данных – наиболее опасный вид атак . С использованием SQL-инъекций злоумышленник может не только получить закрытую информацию из базы данных, но и, при определенных условиях, внести туда изменения. Уязвимость по отношению к SQL-инъекциям возникает из-за того, что пользовательская информация попадает в запрос к базе данных без должной обработке: чтобы скрипт не был уязвим, требуется убедиться, что все пользовательские данные попадают во все запросы к базе данных в экранированном виде.

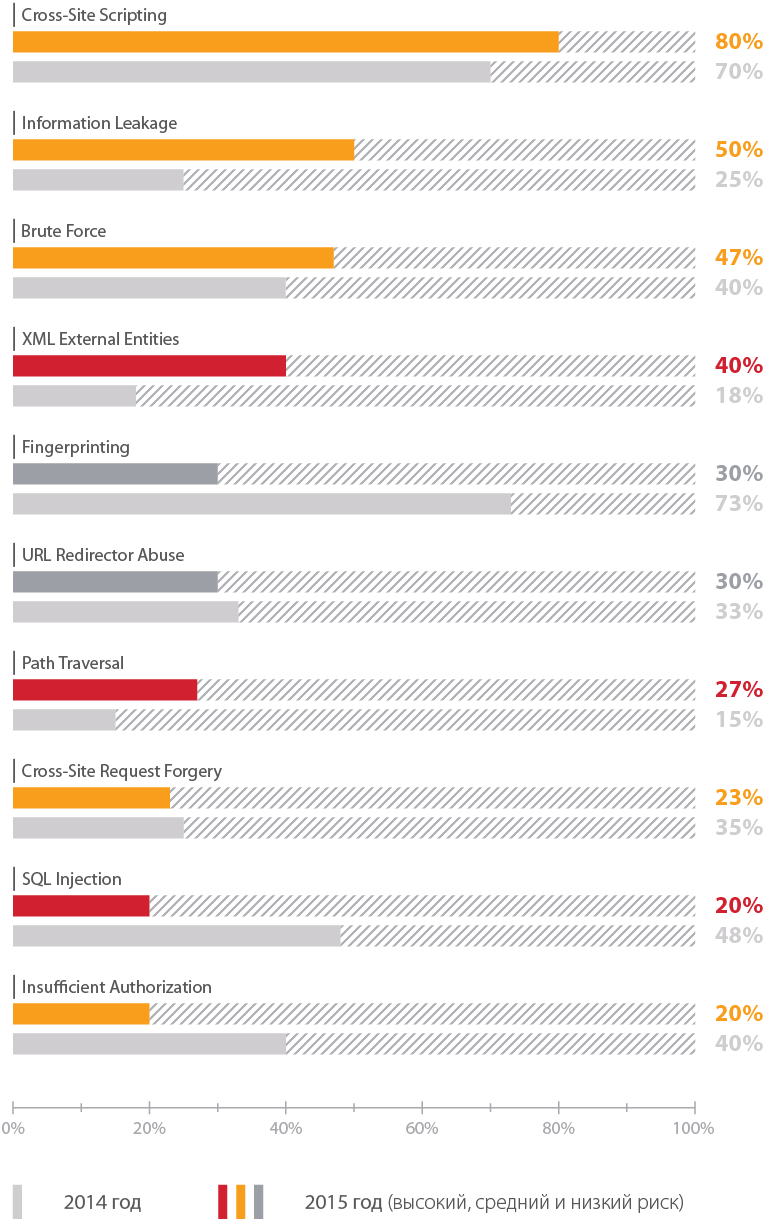
2) PHP-инъекция (англ. PHP injection) – один из способов взлома веб-сайтов, работающих на PHP, заключающийся в выполнении постороннего кода на серверной стороне [3]. Потенциально опасными функциями являются: \* eval(), \* preg\_replace() (с модификатором «e»), \* require\_once(), \* include\_once(), \* include(), \* require(), \* create\_function(). PHP-инъекция становится возможной, если входные параметры принимаются и используются без проверки.

3) Cross-Site Scripting – это вид уязвимости программного обеспечения (Веб-приложений), при которой, на генерированной сервером странице, выполняются вредоносные скрипты, с целью атаки клиента. Сейчас XSS составляют около 15 % всех обнаруженных уязвимостей. Долгое время программисты не уделяли им должного внимания, считая их неопасными. Однако это мнение ошибочно: на странице или в HTTP-Cookie могут быть весьма уязвимые данные (например, идентификатор сессии администратора). На популярном сайте скрипт может устроить DoS-атакy.

4) CSRF (англ. Сross Site Request Forgery – «Подделка межсайтовых запросов», также известен как XSRF) – вид атак на посетителей веб-сайтов, использующий недостатки протокола HTTP. Если жертва заходит на сайт, созданный злоумышленником, от её лица тайно отправляется запрос на другой сервер (например, на сервер платёжной системы), осуществляющий некую вредоносную операцию (например, перевод денег на счёт злоумышленника).

Таким образом, уязвимости в Веб-приложениях по-прежнему остаются одним из наиболее распространенных недостатков обеспечения защиты информации. Проблема защищенности Веб-приложений усугубляется еще и тем, что при разработке Веб-приложений, зачастую не учитываются вопросы, связанные с защищенностью этих систем от внутренних и внешних угроз, либо не достаточно внимания уделяется данному процессу. Это в свою очередь порождает ситуацию, в которой проблемы ИБ попадают в поле зрения владельца системы уже после завершения проекта. А устранить уязвимости в уже созданном Веб-приложении является более расходной статьей бюджета, чем при его разработке и внедрении.

Статистика выявленных уязвимостей за 2015-2014 год показана на (Рис 1).



*Рис 1.*

1. **Сканеры защищенности веб-ресурсов.**

Сканеры защищенности веб-сайтов — это программные (программно-аппаратные) средства, осуществляющие поиск дефектов веб-приложений (уязвимостей), которые приводят к нарушению целостности системных или пользовательских данных, их краже или получению контроля над системой в целом. С помощью сканеров защищенности веб-сайтов можно обнаружить уязвимости следующих категорий:

уязвимости этапа кодирования;

уязвимости этапа внедрения и конфигурирования веб-приложения;

уязвимости этапа эксплуатации веб-сайта.

К уязвимостям этапа кодирования относятся уязвимости, связанные с некорректной обработкой входных и выходных данных (SQL-инъекции, XSS). К уязвимостям этапа внедрения веб-сайта относятся уязвимости, связанные с некорректными настройками окружения веб-приложения (веб-сервера, сервера приложений, SSL/TLS, фреймворк, сторонние компоненты, наличие DEBUG-режима и т. п.). К уязвимостям этапа эксплуатации веб-сайта относятся уязвимости, связанные с использованием устаревшего ПО, простых паролей, хранением архивных копий на веб-сервере в общем доступе, наличием в общем доступе служебных модулей (phpinfo) и т.п.

**Принцип работы сканеров защищенности веб-ресурсов**

В общем случае принцип работы сканера защищенности веб-ресурса заключается в следующем:

Сбор информации об исследуемом объекте.

Аудит программного обеспечения веб-ресурса на предмет уязвимостей по базам уязвимостей.

Выявление слабых мест системы.

Формирование рекомендаций по их устранению.

**Категории сканеров защищенности веб-ресурсов**

Сканеры защищенности веб-сайтов, в зависимости от их предназначения, можно разделить на следующие категории (типы):

Сетевые сканеры — данный тип сканеров раскрывает доступные сетевые сервисы, устанавливает их версии, определяет ОС и т. д.

Сканеры поиска уязвимостей в веб-скриптах — данный тип сканеров осуществляет поиск уязвимостей, таких как SQL inj, XSS, LFI/RFI и т. д., или ошибок (не удаленные временные файлы, индексация директорий и т. п.).

Средства поиска эксплойтов — данный тип сканеров предназначен для автоматизированного поиска эксплойтов в программном обеспечении и скриптах.

Средства автоматизации инъекций — утилиты, которые конкретно занимаются поиском и эксплуатацией инъекций.

Дебаггеры  — средства для исправления ошибок и оптимизации кода в веб-приложении.

Существуют также и универсальные утилиты, которые включают в себя возможности сразу нескольких категорий сканеров.

**Порядок выполнения работы:**

1. Изучить теоретический материал.
2. Выполнить задание:
   1. Найти статистические данные инцидентов выявленных уязвимостей веб-ресурсов за последние 2 года и сравнить из с данными 2015 года (Рис 1). Проанализировать полученные данные. Сделать выводы.
   2. Разработать модель работы сканера уязвимостей веб-ресурса с использованием методологии IDEF0. Детализацию представить до уровня А3.
3. Ответить на контрольные вопросы.

Контрольные вопросы:

1. Что такое уязвимость веб-ресурса?
2. В связи с чем возникают уязвимости?
3. Что такое SQL-инъекции?
4. Что такое PHP-инъекция?
5. Что такое Cross-Site Scripting?
6. Что такое сканер уязвимостей веб-ресурса?
7. Опишите принцип работы сканера уязвимостей веб-ресурса.
8. Перечислите категории сканеров защищенности веб-ресурсов.
9. Опишите сетевые сканеры.
10. Что такое дебаггер?