# **Введение**

Цель лабораторной работы – познакомиться с концепцией ролевого управления доступом и способами защиты программного обеспечения от существующих угроз.

Научиться разрабатывать приложения, которые используют ролевое управление доступом для разграничения полномочий пользователей. Получить навыки защиты разработанной программы от несанкционированного копирования и других угроз, которым может подвергаться программное обеспечение.

# **Краткие теоретические сведения**

Для выполнения данной лабораторной работы было использовано много технологий, в частности:

* Python3 как основной язык разработки.
* Django как фрэймворк для создания web приложения
* SQLite в качестве технологии для БД

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.  
 Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.  
 Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Есть реализация интерпретатора для JVM с возможностью компиляции, CLR, LLVM, другие независимые реализации. Проект PyPy использует JIT-компиляцию, которая значительно увеличивает скорость выполнения Python-программ.

Django **-** свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка **-** DRY (англ. Don't repeat yourself)

Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Архитектура Django похожа на «Модель-Представление-Контроллер» (MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется Представление, а презентационная логика Представления реализуется в Django уровнем Шаблонов. Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» (MTV).

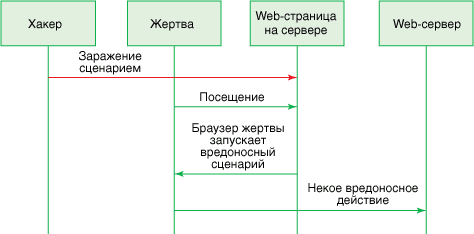
Первоначальная разработка Django как средства для работы новостных ресурсов достаточно сильно отразилась на его архитектуре: он предоставляет ряд средств, которые помогают в быстрой разработке веб-сайтов информационного характера. Так, например, разработчику не требуется создавать контроллеры и страницы для административной части сайта, в Django есть встроенное приложение для управления содержимым, которое можно включить в любой сайт, сделанный на Django, и которое может управлять сразу несколькими сайтами на одном сервере. Административное приложение позволяет создавать, изменять и удалять любые объекты наполнения сайта, протоколируя все совершённые действия, и предоставляет интерфейс для управления пользователями и группами (с пообъектным назначением прав).

В дистрибутив Django также включены приложения для системы комментариев, синдикации RSS и Atom, «статических страниц» (которыми можно управлять без необходимости писать контроллеры и представления), перенаправления URL и другое.

Также Django предоставляет множество возможностей по защите, в частности средства по шифрованию отправляемых web form для защиты от злоумышленников, которые могут попытаться послать неверные данные от лица пользователя.

**Возможные атаки и защита от них**

XSS (межсайтовый скриптинг) атаки – один из самых распространенных вариантов атаки на веб-сервер или компьютер пользователя, суть которой заключается во встраивании определенных html элементов и javaScript кода в отданную сервером разметку, который выполнится на стороне клиента и может запустить какой-либо вирус или же произвести несанкционированное действие от имени пользователя на сервере.



Если говорить немного подробнее, то можно выделить следующие виды XSS атак:

* Хранимые (Постоянные)
* Отражённые (Непостоянные)

Пример постоянных:

- Введённое злоумышленником специально сформированное сообщение в гостевую книгу (комментарий, сообщение форума, профиль) которое сохраняется на сервере, загружается с сервера каждый раз, когда пользователи запрашивают отображение этой страницы.

- Злоумышленник получил доступ к данным сервера, например, через SQL инъекцию (смешение двух типов атак – SQL injections и XSS атак), и внедрил в выдаваемые пользователю данные злонамеренный JavaScript код.

Пример непостоянных:

- На сайте присутствует поиск, который вместе с результатами поиска показывает что-то вроде «Вы искали: [строка поиска]», при этом данные не фильтруются должным образом. Поскольку такая страница отображается только для того, у кого есть ссылка на неё, то пока злоумышленник не отправит ссылку другим пользователям сайта, атака не сработает. Вместо отправки ссылки жертве, можно использовать размещение злонамеренного скрипта на нейтральном сайте, который посещает жертва.

Для предотвращения XSS-атак приложение должно шифровать выходные данные страницы перед передачей их конечному пользователю. При шифровании выходных данных разметка HTML заменяется альтернативными представлениями - объектами. Браузер отображает эти объекты, но не запускает их. Например, *<script>* преобразуется в *&lt;script&gt;.*

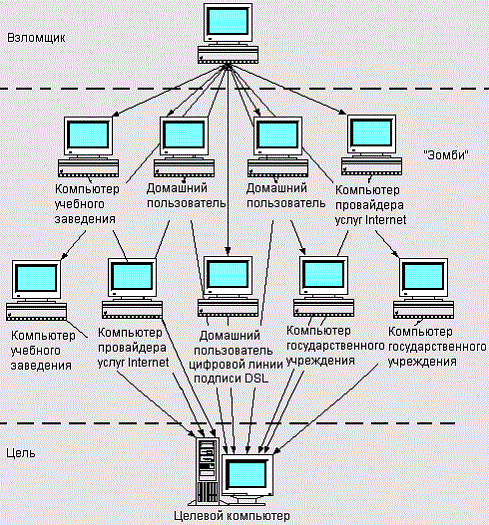
Когда браузер встречает объекты, они преобразуются обратно в HTML и распечатываются, но они не запускаются. Например, если атакующий внедрит в переменное поле Web-страницы сервера строку *<script>alert("вы атакованы")</script>*, то при использовании описанной стратегии сервер возвратит строку *&lt;script&gt;alert("вы атакованы")&lt;/script&gt;.*

Когда браузер загрузит зашифрованный сценарий, он преобразует его к виду *<script>alert("вы атакованы")</script>* и отобразит сценарий в составе Web-страницы, но не запустит его.

В случае приложения, реализованного в данной лабораторной работе, защита была реализована прямо на сервере gunicorn, который благодаря вышеописанной процедуре не позволяет запустить вредоносный скрипт на стороне пользователя.

SQL инъекции – также один из очень распространенных видов атак на приватные пользовательские данные или же непосредственно на устройство пользователя, а также и на сам сервер с целью уничтожения или распространения данных. Суть заключается в том, что некоторые веб формы могут быть связаны с SQL запросами и если пользователь введет в такую форму SQL запрос, он может исполниться прямо на стороне сервера и принести очень много проблем.

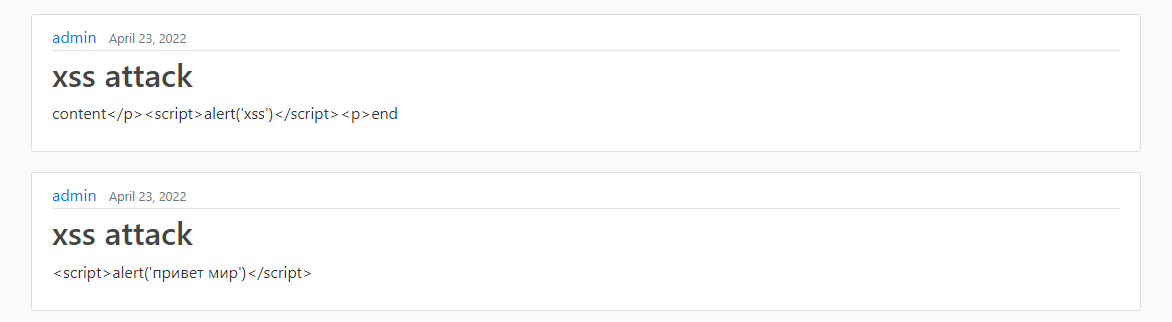
DoS атаки также популярный вид атак, который заключается в перегрузке сети запросами. По сути, это  хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых добросовестные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т. д.). Но чаще это мера экономического давления: потеря простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.



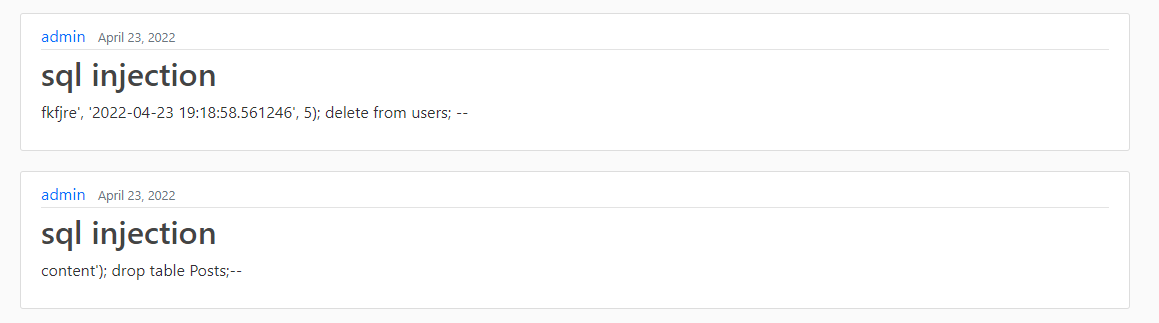
Защититься от DoS атаки является очень трудной задачей. Для защиты от сетевых атак применяется ряд фильтров, подключенных к интернет-каналу с большой пропускной способностью. Фильтры действуют таким образом, что последовательно анализируют проходящий трафик, выявляя нестандартную сетевую активность и ошибки. В число анализируемых шаблонов нестандартного трафика входят все известные на сегодняшний день методы атак, в том числе реализуемые и при помощи распределённых бот-сетей. Фильтры могут реализовываться как на уровне маршрутизаторов, управляемых  свичей, так и специализированными аппаратными средствами. Также возможна установка задержек на запросы от пользовательских машин (такой способ и был реализован в лабораторной работе), однако при большом количестве устройств данная защита начинает заметно проседать.

# **Демонстрация работы программы**

Отраженная XSS атака:



Отраженная SQL injection атака:



Отраженная DDOS атака:

