|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**Отчет**

**по домашнему заданию № 1**

**Название:** Выявление рисков нарушения целостности, доступности и конфиденциальности информации в автоматизированной системе

**Дисциплина:** Защита информации

Студент гр. ИУ6-85Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.Д. Шульман**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.А. Миков**

(Подпись, дата) (И.О. Фамилия)

# Выбор автоматизированной системы

В качестве предмета изучения выбрана тема дипломной работы «Программная система моделирования искусственной жизни».

Данная автоматизированная система представляет собой веб-приложение, доступное множеству пользователей в сети Интернет. Архитектурная схема программного обеспечения рассматриваемой системы представлена на рисунке 1. Согласно данной схеме будут выявляться риски и возможные уязвимости рассматриваемой системы, а также будут рассмотрены возможные способы их решения.



Рисунок – Архитектура программного обеспечения рассматриваемой автоматизированной системы

В представленной системе присутствуют следующие компоненты:

* Прокси-сервер nginx;
* Frontend-сервер Node.js;
* Backend-сервер Golang;
* Веб-сервис моделирования Golang;
* База данных PostgreSQL.

Для каждого из перечисленных компонентов будут рассмотрены возможные уязвимости.

# Выявление потенциальных угроз и возможных уязвимостей

## Нарушение доступности

Нарушение доступности может произойти вследствие атаки на ряд компонентов системы.

### Атака на proxy-server

Нарушения доступности системы может произойти в первую очередь из-за вывода из строя сервера nginx, ввиду того, что он представляет из себя единую точку входа в систему.

Нарушение работы nginx может произойти по следующим причинам:

* падение из-за высокой нагрузки и недостатка ресурсов;
* падение под воздействием DDoS атаки (формирование большого количества запросов);
* падение под натиском большого количества медленных клиентов (один из вариантов DDos атаки, приводит к быстрому росту необходимых поддерживаемых соединений);
* падение под тяжестью «толстых клиентов» (необходимо проксировать недопустимо большие объемы данных).

### Потенциальные угрозы для frontend-сервера

Преднамеренное большое количество запросов на получение веб-приложения может привести к увеличению среднего времени отклика, что приведет к дополнительному времени ожидания загрузки сайта и понижению уровня доступности приложения в целом.

### Высокая нагрузка на backend-server

Большинство проблем, которые возникают на стадии проксирования, также актуальны и для этого компонента, если они не были решены раннее.

## Нарушение целостности рассматриваемой автоматизированной системы

Нарушение целостности может быть вызвано выводом из строя любого из компонентов программного обеспечения данной автоматизированной системы.

### Внедрение вредоносного кода и sql-инъекции

Внедрение в веб-приложение вредоносного кода и запросов может привести к целому ряду проблем, в том числе и к нарушению целостности из-за появления ошибок при работе отдельных компонентов и выхода их из строя.

### Целенаправленное создание высокой нагрузки на связующие элементы системы

Создание нагрузки на компоненты системы, обеспечивающие жизнеспособность всей системы в целом может привести не только к проблемам доступности всей системы, но и к нарушению корректности работы связанных элементов и даже к выводу их из строя.

Может иметь место, если система спроектирована как сильносвязанная, каскадный эффект, который приведет к полной неработоспособности всей системы.

### Уничтожение физических серверов

Уничтожение физических серверов, если система не имеет резервов, неизбежно ведет к нарушению целостности.

## Нарушение конфиденциальности информации

Рассматриваемая система является веб-приложением, которое доступное в сети Интернет множеству пользователей, что означает, что существует множество аккаунтов пользователей, которые не должны иметь доступ к данным чужих профилей и модифицировать их.

### SQL-инъекции в базе данных для получения доступа к приватным данным

Внедрение SQL, в зависимости от типа используемой СУБД и условий внедрения, может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

Атака типа внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах.

### Внедрение XSS в frontend-приложение

XSS не так опасна для сервера, но опасна для пользователя. Она работает в браузере пользователя, и поэтому позволяет только украсть его данные. XSS или Cross-Site Scripting работает в JavaScript. Злоумышленник передает специальную строку, в строке содержится JS код, далее браузер думает, что этот код отправлен сайтом и выполняет его, а код может быть любым.

### Уязвимость междоменного взаимодействия CSRF

Атака CSRF или Cross-Site Request Forgery позволяет злоумышленнику заставить браузер жертвы отправить определенный HTTP запрос, включая куки, файлы сеанса и любую другую, автоматически включаемую информацию в уязвимое веб-приложение.

## Резюме

Основным источником угрозы является пользователь-злоумышленник (внешняя преднамеренная угроза).

Основными объектами угрозы является аккаунты пользователя и их приватная информация.

Возможные способы несанкционированного доступа – похищение пароля пользователя, перехват сессии, внедрение вредоносного кода в компоненты программной системы.

# Составление модели нарушителя

Т.к. в ходе выявления потенциальных угроз было выяснено, что большая часть из них относится к воздействию на систему из вне, то было принято решение построить модель для такого нарушителя.

С учётом особенностей разрабатываемой системы, предполагается, что нарушителем может быть одиночка или небольшая группа авантюристов. Модели нарушителя такие, как «организация-конкурент» и «правительственная структура» не будут рассматриваться, т.к. разрабатываемая автоматизированная система не является коммерческим продуктом, а также не имеет под собой никакой политической подоплеки.

|  |  |  |
| --- | --- | --- |
| **Характеристика** | **Одиночка** | **Группа злоумышленников** |
| Вычислительная мощность технических средств | Персональный компьютер | ЛВС |
| Доступ к Интернету, тип каналов доступа | Модем или выделенная линия | Использование выделенных каналов с высокой пропускной способностью |
| Финансовые возможности | Сильно ограниченные | Ограниченные |
| Уровень знаний в области IT | Невысокий или средний | Средний или высокий. Множество компетенций |
| Используемые технологии | Готовые программы, самописные скрипты, использование популярных уязвимостей | Исследование атакуемой системы, поиск уязвимостей, изготовление вредоносных программ, внедрение троянов и вирусов |
| Знание о системе защиты атакуемой системы | Отсутствие знаний об особенностях системы | Могут предпринимать значительные усилия по получения информации о работе системы на протяжения длительного временного периода |
| Преследуемые цели | Эксперимент, получение реального опыта взлома, хулиганство, особые личные мотивы | Вымогательство, внесение искажений в работу системы, похищение личных данных пользователей |
| Характер действий | Скрытый | Скрытый |
| Глубина проникновения | Первое успешное воздействие или до обнаружения | До достижения поставленных целей, до обнаружения или возникновения непреодолимого препятствия. |

На основании построенных моделей нарушителей можно предугадать основные векторы атак, на основание чего разработать методы защиты системы от попыток нарушить конфиденциальность информации или целостность самой сиcтемы.

# Способы и средства защиты. Основные направления защиты

Для устранения выявленных уязвимостей предлагается предпринять ряд мер. Защита должна быть обеспечена на всех доступных уровнях в рамках разумного. Основная часть мер должна быть применена на тех элементах программной системы, на которые в первую очередь поступает на обработку пользовательский запрос.

## Обеспечение защиты от нарушения доступности и целостности системы в единой точке доступа к системе

На этапе проксирования запроса от пользователя на сервере nginx можно предпринять некоторое количество мер по защите системы.

### Блокировка DDoS источника по IP-адресу

Позволяет обеспечить частичную защиту от DDoS спама, однако из-за NAT не может быть использоваться постоянно. Для надежности защита должна обеспечиваться провайдером, т.к. он обладает большими ресурсами и знаниями о конфигурации сети Интернет для блокировки нарушителей.

### Ограничение количества запросов от пользователей в единицу времени

Данное ограничение позволяет значительно сэкономить ресурсы в случае, если некоторая группа юзеров совершает аномально большое количество запросов, в котором нет необходимости при штатном использовании сервисов, предоставляемых веб-приложением.

### Отключение медленных пользователей

Отключение медленных пользователей позволяет защититься от атак, которые используют медленное соединения для утилизации большого процента ресурсов на сервисе, на который совершается атака.

## Обеспечение защиты в приложении frontend-сервера

Для защиты пользователя от некорректного поведения на стороне веб-приложения в браузере необходимо обеспечить защиту от наиболее частых типов атак со стороны внешнего нарушителя.

### Защита от XSS

Для защиты от XSS (внедрение вредоносного кода) обеспечивается сериализация данных, которая призвана из вводимой информации удалять все типы символов, которые могут быть восприняты как исполняемый код при интерпретации браузером.

### Защита от CSRF

Запрос CSRF-токена – один из самых популярных и надежных на данный момент способов защиты от CSRF-атак. Обеспечивает защиту от несанкционированных действий при междоменном взаимодействии.

## Обеспечение защиты на backend-сервере

На данном программном компоненте стоит в первую очередь обратить внимание на защиту пользовательских данных, т.к. данный компонент напрямую взаимодействует с базами данных.

### Защита от XSS

Использование экранированных входных\выходных данных. Применение встроенных функций для очистки кода от вредоносных скриптов.

### Использование белых списков для доверенных хостов

Для взаимодействия со всеми компонентами программной системы должен использоваться подход «белых списков», когда разрешено взаимодействие только с теми хостами, которые были явно прописаны в списке разрешенных. В этом случае какие-либо взаимодействие с «чужими хостами» будет запрещено.

Это также обеспечивает дополнительную защиту, когда проводится попытка нарушения работоспособности системы изнутри.

## Обеспечение защиты данных в БД

Осуществляется непосредственно средствами реляционной базы данных postgreSQL.

### Вход по паролю

Для взаимодействия с базой данных требуется пройти идентификацию и аутентификацию.

### Разграничение прав доступа

Перед совершением какого-либо действия проводится авторизация пользователя.

### Защита полей таблиц

Можно явно заблокировать для пользователей БД операции по удалению и модификации данных, без предварительного резервного копирования изменяемых данных.

### Шифрования данных

Наиболее ценные данные в базе должны храниться в зашифрованном виде. Например, пароли.

Заключение

В результате выполнения домашнего задания были выявлены потенциальные угрозы и возможные уязвимости в рассматриваемой автоматизированной системе.

Были выявлены наиболее возможные нарушители рассматриваемой информационной системы, а также построены их модели.

На основании выявленных угроз и моделей нарушителей был принят ряд мер по обеспечению защиты от нарушения конфиденциальности информации, целостности и доступности автоматизированной системы.