МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Информационных технологий и систем безопасности

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(дипломная работа)

На тему: «Управление безопасностью сайта»

Исполнитель Черняк Алексей Сергеевич

(фамилия,имя,отчество)

Руководитель Бурлов Вячеслав Георгиевич

(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой _____

(подпись)

_____

(ученаястепень,ученое звание)

_____

(фамилия,имя,отчество)

«_____»_____2023г.

Санкт–Петербург

2023

ОГЛАВЛЕНИЕ

Введение

Internet – всемирная система объединённых компьютерных сетей для хранения и передачи информации. В большинстве своём, информационные системы создаются в виде веб-сайтов. В связи с этим существуют различные атаки и угрозы несанкционированного доступа к персональным данным пользователям и базам данных различных веб-ресурсов. Атаки и угрозы распространены в сети и веб-технологиях, из-за несовершенства информационных технологий, отсутствие защиты ПО и других факторов.

Веб-серверы и Веб-сайты - это объекты, которые постоянно подвергаются опасности и уязвимостям со стороны злоумышленников. Веб-серверы являются самым уязвимым элементов в функционировании Веб-сайта, к которым хакер может получить доступ к конфиденциальным данным, размещенным на сервере и изменить их содержимое, тем самым нарушить целостность системы, а также вывести из строя сервер с помощью распределенной атаки (DDoS-атака), которая приведет к отказу в доступе пользователям.

Данная совокупность факторов определяет актуальность настоящей ВКР, целью является выбрать и обосновать условия обеспечения информационной безопасности Веб-сайта в условиях глобальной сети интернет на основе разработки аналитической и математической модели.

Для достижения этой цели в работе поставлена и решена следующая задача - осуществить анализ существующих угроз, стандартных средств защиты клиент-серверных связей и создать на основе полученных результатов прототип безопасного веб-ресурса.

1. Основы обеспечение безопасности Веб-сайта

Информационная среда – это среда, в которой реализуется совокупность информационных процессов и средств для обслуживания объектов инфраструктуры территориальных образований в интересах обеспечения баланса спроса и предложения на рынке информационных услуг.

В современных условиях информационная среда является одним из основных компонентов системы деятельности территориального образования. Рынку информационных услуг присуща конкурентная борьба и другие процессы, порождающие деструктивные воздействия на информационную среду. Особое место в такой деятельности в настоящее время занимают веб-технологии. Однако в силу деструктивных воздействий результаты деятельности специалистов по информационным технологиям не оправдывают их ожидания. То есть информационные процессы не достигают поставленной цели. Такая ситуация породила проблему обеспечения ИБ.

Для рационального использования информационного ресурса при применении веб-технологий в условиях деструктивных воздействий среды необходимо уметь формировать информационные процессы с наперёд заданными свойствами [1]. Для этого целесообразно использовать естественно-научный подход, разработанный научной школой «Системная интеграция процессов государственного управления». Эта научная школа зарегистрирована в Реестре ведущих научно-образовательных школ Правительства СПБ [2-3].

Процесс обеспечения ИБ строится на основе системной интеграции

- целевого информационного процесс;
- процесса образования угрозы;
- процесса идентификации угрозы;
- процесса нейтрализации угрозы;
- показателя информационной безопасности.

Показателем безопасности является вероятность того, каждая угроза идентифицируется и нейтрализуется для данного целевого информационного процесса.

Технология обеспечения ИБ строится на основе следующих рассуждений:

- задаётся требуемый показатель уровня безопасности (вероятность Р)
- целевой информационный процесс;
- процесс образования угроз.

В результате системной интеграции формируется условие существования процесса обеспечения ИБ. Это условие в итоге позволяет сформировать одно уравнение с двумя неизвестными:

- обобщённая характеристики процесса идентификации угрозы;
- обобщённая характеристика нейтрализации угрозы.

Формирование обобщённых характеристик этих двух процессов исходя из требуемого уровня ИБ позволяет удовлетворить условие существования процесса обеспечения ИБ и тем самым обеспечить требуемую ИБ. В целом, этот результат позволил перейти к разработке аналитической математической

модели обеспечения ИБ веб-сайта в условиях деструктивных воздействий среды.

Предпринимая первые шаги к проектированию, созданию и использованию нового сайта — в первую очередь следует обеспечить защиту наилучшего уровня безопасности сервера, на котором он будет размещаться.

Веб-сервер формируется из некоторого ПО, каждый элемент которых уязвим и подвержен множеством способов атаки, как показано на рисунке 1. Любой из элементов может стать целью атаки:

Рисунок 1 – Основные элементы функционирования web-сайта.

Веб-сайт будет не доступен для вирусных и хакерских атак, если все 3 элемента организованы надлежащим образом. В случае если хотя бы один из них будет слабым, веб-сайт станет уязвимым для хакинга.

1.1. Процесс функционирования веб-сайта

В большей части глобальной сети используется клиент-серверная модель. В этой модели, устройства, которые используют пользователи – обмениваются данными через сеть с центрально расположенными серверами для получения необходимых данных, а не друг с другом. Устройства конечных пользователей, такие как ноутбуки, смартфоны и персональные компьютеры, считаются «пользователями» серверов. Пользовательские устройства отправляют запросы на серверы для веб-страниц или приложений, а серверы отправляют ответы.

Клиент-серверная модель используется потому, что сервера обычно более мощные и надежные, чем пользовательские устройства. Они также постоянно поддерживаются и хранятся в контролируемой среде, чтобы быть уверенными, что они всегда включены и доступны, хотя отдельные серверы могут выходить из строя, обычно их поддерживают другие серверы. Между тем пользователи могут включать и выключать свои устройства, и это никак не скажется на сервис глобальной сети для других пользователей.

Серверы могут обслуживать несколько пользовательских устройств одновременно, и каждое такое устройство отправляет запросы нескольким серверам в ходе доступа к глобальной сети и просмотра его страниц.

На рисунке 2 представлен принцип взаимодействия нескольких клиентов и серверов.

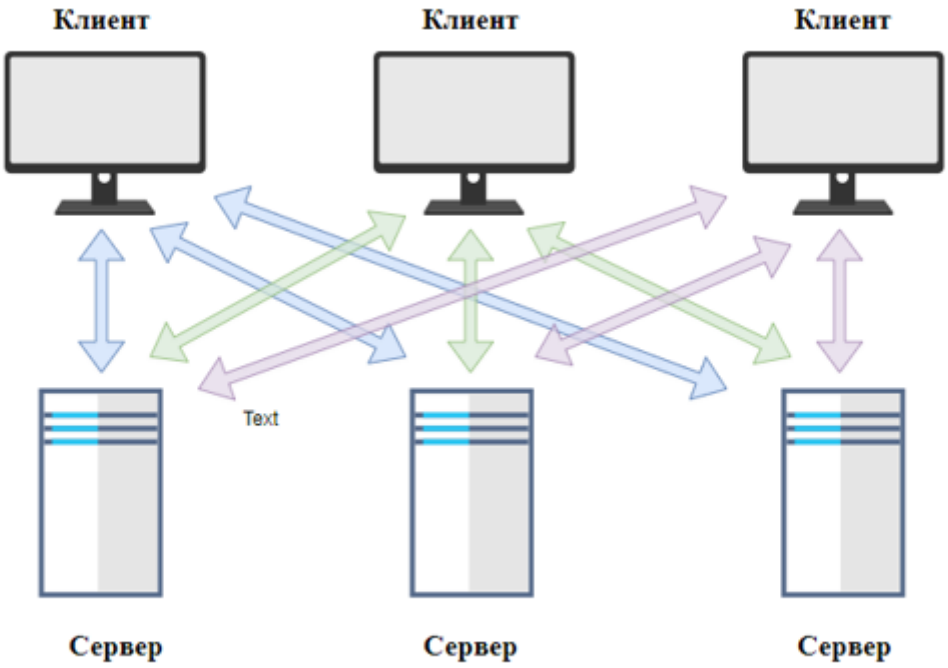

Рисунок 2 – принцип взаимодействия нескольких клиентов и серверов.

Каждый клиент будет общаться с несколькими серверами, и наоборот.

Предположим, пользователь просматривает Интернет и вводит «www.google.com» в свою панель браузера. Это приводит к запросу DNS-серверов * на получение IP-адреса * www.google.com, а DNS-серверы отвечают на этот запрос, передавая IP-адрес браузеру. Затем браузер пользователя отправляет запрос на сервера Google (используя IP-адрес) для содержимого, отображаемого на странице, такого как уменьшенные изображения фильмов, логотип amediateka и панель поиска. Серверы amediateka доставляют это в браузер, и браузер загружает страницу на клиентское устройство.

В веб-разработке «сторона клиента» относится ко всему в веб-приложении, которое отображается или происходит на клиенте (устройстве конечного пользователя). Это включает в себя то, что видит пользователь, например, текст, изображения и остальную часть пользовательского интерфейса, а также любые действия, которые приложение выполняет в браузере пользователя.

** IP-адреса – это Интернет-протокол, который представляет собой набор правил, позволяющих устройствам взаимодействовать через Интернет. Поскольку миллиарды людей получают доступ к Интернету каждый день, уникальные идентификаторы необходимы для отслеживания того, кто что делает. Интернет-протокол решает эту проблему, назначая IP-номера каждому устройству, имеющему доступ в Интернет.*

** DNS-сервер – это система, которая управляет всем этим. Он работает как телефонная книга для IP-адресов, так что пользователи могут получать доступ к веб-службам, используя понятные для человека доменные имена. Когда пользователь вводит доменное имя, в свое окно браузера,*

начинается DNS-запрос, который в конечном итоге приводит к тому, что DNS-сервер переводит доменное имя в IP-адрес.

Во всем мире пользователи Интернета используют веб-системы, которые следуют парадигме клиент-сервер. Веб-браузер действует как клиент для веб-сервера, который предоставляет услугу. Эти веб-системы полагаются на протокол HTTP для связи, но это протокол без сохранения состояния. Поэтому разработчикам приложений приходится использовать альтернативные методы для идентификации и аутентификации пользователя и сохранения состояния пользователя. Механизмы без сохранения состояния и с отслеживанием состояния могут использоваться с HTTP для отслеживания сеансов и запоминания информации, специфичной для пользователя. Сеансы сохраняют пользовательские переменные и состояние посредством последовательных запросов страниц. Сеансы обычно используются для обеспечения соблюдения ограничений безопасности и для инкапсуляции информации о состоянии во время выполнения. Критической проблемой веб-безопасности является возможность привязки проверки подлинности пользователя и управления доступом к уникальным сеансам.

Управление сеансом позволяет веб-системам создавать сеансы и поддерживать данные сеанса, специфичные для пользователя, так что пользователю не нужно повторно проходить аутентификацию при выполнении действий. На первом этапе выполняется процесс аутентификации, чтобы проверить, имеет ли пользователь право доступа к ресурсу. После аутентификации пользователю предоставляется уникальный идентификатор сеанса. Таким образом, управление сеансом достигается за счет сохранения уникального идентификатора сеанса на стороне клиента и сервера, и браузер должен отправлять идентификатор сеанса с каждым новым запросом.

К сожалению, существующие методы управления сеансами изначально предназначены для надежной среды, которой больше не является Интернет, и поэтому они не могут

Интерфейс веб-сайта — это все, что вы видите и с чем можете взаимодействовать с помощью браузера. Итак, создание этой визуальной части называется фронтенд-разработкой. Можно даже сказать, что дизайнеры, создающие пользовательские интерфейсы и планирующие опыт, также являются фронтенд-разработчиками, поскольку они совместно работают над одной и той же частью проекта.

Для создания внешнего интерфейса инженеры используют комбинацию HTML (для базовой структуры страницы и содержимого), CSS (для визуального редактирования) и JavaScript (для создания интерактивных веб-сайтов). Тот же набор инструментов используется для создания прогрессивных веб-приложений — мобильных приложений, которые выглядят и работают как нативные, но создаются с использованием фронтенд-технологий. Подробнее об этом в статье по ссылке.

С другой стороны, бэкенд — это все, что происходит, ну, за кулисами. Он содержит серверы, на которых расположены ваши веб-страницы, и базовую логику, которая управляет функциями и процессами веб-сайта. У нас есть подробное описание внутренней работы веб-приложений, если вы хотите его проверить.

Серверная часть построена с использованием другого набора технологий, включая Java, PHP, Ruby, C# и иногда JavaScript, которые мы объясним в соответствующем разделе. Базовый набор инструментов для интерфейса четко определен: HTML, CSS и JavaScript. Однако технологии разработки внешнего интерфейса могут быть расширены с помощью менеджеров пакетов, препроцессоров CSS, фреймворков и многого другого.

Как и на стороне клиента, «сторона сервера», а именно серверная часть, означает все, что происходит на сервере, а не на клиенте. В прошлом почти вся бизнес-логика работала на стороне сервера, и это включало рендеринг динамических веб-страниц, взаимодействие с базами данных, идентификацию подлинности и push-уведомления.

Проблема с размещением всех этих процессов на стороне сервера заключается в том, что каждый запрос, связанный с одним из них, должен каждый раз проходить весь путь от клиента до сервера. Это вносит большую задержку. Одним из вариантов использования является рендеринг динамических веб-страниц в реальном времени путем запуска скриптов в браузере, которые вносят изменения в контент, который видит пользователь.

Как и в случае с «внешним интерфейсом» и «клиентской стороной», back-end* также является термином для процессов, происходящих на сервере, хотя back-end относится только к типам процессов, а на стороне сервера - к месту, где выполняются процессы.

Сценарии на стороне клиента просто означают запуск сценариев, таких как JavaScript, на клиентском устройстве, обычно в браузере. Все виды сценариев могут выполняться на стороне клиента, если они написаны на JavaScript, потому что JavaScript поддерживается повсеместно. Другие языки сценариев могут использоваться, только если их поддерживает браузер пользователя.

Серверные сценарии выполняются на сервере, а не на клиенте, часто для доставки динамического содержимого на веб-страницы в ответ на действия пользователя. Серверные сценарии не обязательно должны быть написаны на JavaScript, поскольку сервер может поддерживать различные языки. Принцип запуска скриптов представлен на Рисунке 3.

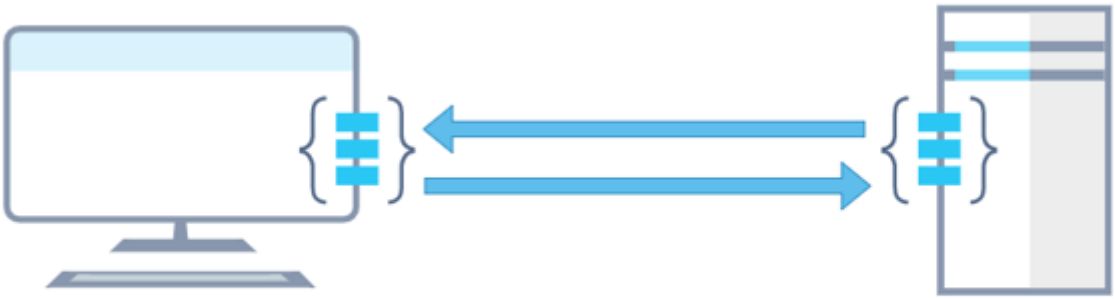

Рисунок 3 – Принцип запуска скриптов на стороне клиента и на стороне сервера.

По сути, фиксация сеанса представляет собой трехэтапную атаку. Этапы атаки фиксации сеанса. Чтобы провести атаку с фиксацией сеанса, злоумышленник создает ссылку на целевой веб-сайт вместе с выбранным идентификатором сеанса. Это первый шаг атаки фиксации сеанса. Иногда злоумышленнику необходимо инициировать сеанс с сервером перед аутентификацией и передать URL-адрес этого сеанса пользователю для аутентификации. В этих случаях злоумышленнику необходимо поддерживать сеанс ловушки через регулярные промежутки времени.

На втором этапе злоумышленнику необходимо представить ловушку сеанса пользователю-жертве и исправить сеанс. Обычно злоумышленник размещает ссылку в общедоступном месте. Чтобы быть в большей безопасности, злоумышленник может использовать XSS-уязвимость целевого веб-сайта, если это возможно, и разместить ссылку на целевом сайте. По-видимому, пользователь-жертва может нажать на определенную ссылку и попытаться войти в систему. Это последний шаг атаки фиксации, когда жертва получает доступ к фиксированному сеансу. Поскольку созданная ссылка закодирована с помощью SID, веб-сервер не будет пытаться перезаписать ее своим собственным.

Следовательно, пользователь сможет войти в систему с указанным злоумышленником SID. Теперь сессия становится доступной для злоумышленника даже без аутентификации. Задача сервера состоит в том, чтобы быстро обрабатывать запросы клиентов. Для этого требуется ресурсы, с помощью которых сайт быстро обрабатывают данные. В первую очередь это хороший процессор, достаточная оперативная память, ширина канала к серверу и дисковое пространство.

1.2. Анализ угроз для функционирования Веб-сайта

Как и обычное приложение или операционная система, веб-сайты могут иметь недостатки безопасности. Это является серьезной проблемой, так как Веб-сайт использует конфиденциальные данные (пароли, номера банковских карт и т.д.). Даже на защищенном веб-сервере, работающем под управлением

известной безопасной операционной системы, могут сохраняться уязвимости безопасности, поскольку они в основном вызваны ошибками программирования самого приложения, а не сервера. Введем несколько основных терминов, которые будем использовать в работе:

Угроза безопасности информации – это совокупность условия и факторов, создающих потенциальную или реально существующую опасность, связанную с утечкой информации, несанкционированными или непреднамеренными воздействиями на нее [4].

Уязвимость - это случайная или преднамеренная ошибка, в конфигурации системы или в способе ее использования. Уязвимость может быть использована для создания вторжения.

Атака - это ошибка вредоносного взаимодействия, направленная на нарушение одного или нескольких свойств безопасности. Это внешняя ошибка, созданная с намерением причинить вред.

Вторжение - это злонамеренная ошибка внутреннего, но также внешнего происхождения, возникающая в результате атаки.

Самым первым шагом в устранении существующих хакерских атак и уязвимостей является идентификация угроз – это установление всех возможных и актуальных угроз, которые имеют место быть для Веб-сайта в процессе его создания и эксплуатации.

Атака с использованием межсайтовых сценариев — это разновидность атаки на стороне клиента. В XSS злоумышленник может установить токен сеанса, а также разместить вредоносный код на целевой веб-странице. Эта атака становится возможной в веб-системах, которые не выполняют надлежащую очистку и фильтрацию пользовательского ввода. Более того, иногда уязвимость вызвана процессом фильтрации, который не может отфильтровать некоторые типы ввода. Несмотря на то, что результат процесса фильтрации не является правильно отформатированным HTML-файлом, некоторые браузеры созданы для их обработки. Таким образом, иногда XSS-атаки успешны в некоторых конкретных браузерах. Например, в сеансе на основе файлов cookie XSS можно использовать для обхода проверок безопасности браузера. Браузер позволяет сайту читать или изменять файл cookie, если он был установлен сайтом в том же домене.

Используя межсайтовый скриптинг, злоумышленник может загрузить настроенную ссылку с вредоносным кодом на целевой веб-сайт, и когда жертва нажимает на ссылку, предоставленную злоумышленником, вредоносная программа запускается на стороне клиента. Поскольку код загружается на сайт, который установил файл cookie, браузер позволит вредоносному коду получить доступ к файлу cookie и изменить его. Злоумышленник может использовать программы для установки значения cookie по своему выбору.

Для сеанса, распространяемого с помощью данных POST, злоумышленник может создать форму со скрытыми элементами, содержащими определенный SID, и опубликовать код в общедоступном месте.

Для систем, использующих данные GET для отслеживания сеанса, XSS-атаку можно запустить, создав специально созданную ссылку, которая кодирует выбранный злоумышленником идентификатор сеанса.

В данной ВКР будут рассмотрены самые распространённые на данный момент времени угрозы безопасности при функционировании веб-ресурса. Ниже перечислены основные уязвимости:

1. Уязвимости внедрения и межсайтового скриптинга (XSS).
2. Межсайтовый запрос / подделка ссылок (CSRF)
3. Проблемы аутентификации и управления сессиями
4. Недостаточная защита транспортного уровня
5. SQL-инъекция
6. Распределенная атака типа «отказ в обслуживании» (DDoS)
7. Неправильная настройка безопасности

1.2.1Уязвимости внедрения и межсайтового скриптинга (XSS)

Инъекционные атаки бывают различных форм, включая внедрение в код SQL, операционную систему, электронную почту, и все они основаны на отправке вредоносных данных в приложение как часть запроса. Тщательно созданные данные могут заставить приложение выполнять нежелательные команды или получать доступ к неавторизованным данным.

Внедрение SQL происходит, когда хакеры получают контроль над сайтами, которые генерируют запросы SQL из предоставленных пользователем данных, без предварительной проверки достоверности данных. Эти хакеры могут отправлять вредоносные запросы SQL и передавать команды непосредственно в базу данных.

XSS (межсайтовый скриптинг) атакует целевых пользователей приложения путем внедрения кода, обычно клиентского скрипта (такого как JavaScript), в выходные данные веб-приложения. Как только отображается зараженный выход или страница, браузер выполняет код, позволяя злоумышленнику перехватить пользовательские сеансы, перенаправить пользователя на вредоносный сайт или просто повредить страницу. Атаки XSS возможны в содержимом динамически генерируемой страницы, с того момента, как приложение интегрирует данные, предоставленные пользователем, без проверки или правильного их преобразования.

Как правило, файл cookie хранится в клиентском браузере в файле, и информация в файле cookie легко читается и доступна клиентам, если она не зашифрована или не хеширована должным образом. Шифрование значения cookie предотвращает доступ злоумышленника к информации. Однако это также может создать уязвимость, предоставив злоумышленнику возможность узнать ключ шифрования. Если злоумышленник сможет вычислить ключ шифрования, он может нанести серьезный ущерб системе.[9] Основным недостатком механизма cookie является то, что злоумышленник может заставить клиента выполнять некоторые действия приложения от его имени. Вредоносный сайт может содержать ссылку на целевое веб-приложение в

поддельном ресурсе, а ссылка может содержать запрос на выполнение какого-либо вредоносного действия. Аутентифицированный, уже вошедший в систему пользователь целевого веб-приложения автоматически отправит файл cookie со ссылкой на ресурс-ловушку при посещении вредоносного сайта и выполнит указанное действие. Кроме того, злоумышленник может создать URL-адрес, выдающий себя за исходный веб-сайт, и заставить клиента отправлять файлы cookie злоумышленнику. Этот тип атаки известен как сбор файлов cookie.

1.2.2 Межсайтовый запрос / подделка ссылок (CSRF)

CSRF - это атака, которая заставляет пользователя предпринять какие-либо действия в уязвимом приложении без его ведома. Это может произойти, когда пользователь посещает веб-страницу, содержащую вредоносный запрос, который затем выполняет выбранное действие от имени клиента.

Файл cookie также уязвим для атаки с использованием межсайтовых сценариев, когда злоумышленник может получить доступ к файлу cookie с помощью вредоносного сценария. Однако для предотвращения этой атаки вводится дополнительный атрибут cookie HttpOnly. Когда этот атрибут присутствует в файле cookie, он недоступен через Javascript. Когда файл cookie передается по незащищенному каналу, он может быть перехвачен злоумышленником. Файлы cookie с установленным флажком «Безопасность» никогда не будут передаваться по незащищенному соединению. Еще одна серьезная проблема использования файлов cookie заключается в том, что их можно отключить в настройках браузера. Веб-приложение не сможет отслеживать сеансы пользователей, у которых отключены файлы cookie.

Обнюхивание сеанса — это своего рода перехват. Перехват возможен, когда злоумышленник может собрать данные, из которых он может вычислить SID. Реализация перехвата сеанса более сложна, чем прогнозирование сеанса. Перехват сеанса можно рассматривать как атаку «человек посередине», когда злоумышленник может перехватить конфиденциальный токен сеанса, передаваемый между пользователем и сервером, посредством перехвата трафика передачи между ними. После захвата токена сеанса злоумышленник может действовать как законный пользователь на сервере и может получить несанкционированный доступ. Перехват сеанса возможен, когда канал связи между пользователем и веб-сервером небезопасен. Другими словами, когда веб-служба не настроена на использование соединения HTTPS, все данные, передаваемые между клиентом и сервером, передаются в виде открытого текста. Без шифрования передача может быть перехвачена любым компьютером в сети, через которую проходит пакет. Например, элементы формы POST на странице входа содержат имя пользователя и пароль, и эти данные можно увидеть при перехвате передачи открытого текста. После этого ответ HTTP от сервера может содержать заголовок Set-Cookie, включая SID.

1.2.3 Проблемы аутентификации и управления сессиями

Недавно механизмы безопасности были классифицированы по как ориентированный на сеть или ориентированный на хост, в зависимости от об их модели развертывания и типе деятельности они наблюдают и проверяют. Сетецентрические подходы будет включать системы контроля доступа к сети или обнаружения вторжений, а также антивирусное программное обеспечение и хост-системы. Системы обнаружения вторжений, установленные на хостах, будут классифицируется как хост-ориентированный. Насколько они важны в на предприятии сетецентрическим механизмам не хватает необходимой глубины или контекста, поскольку они проверяют биты или данные, которые передачи по сети, особенно в тех случаях, когда данные передаются по SSL. Далее осмотр компонентов сети по отдельности не комплексно проанализировать общую безопасность сети [12]. Исследование методов получения более широкой картины что происходит в сети и предложен ряд инструментов анализа.

Атака с перехватом сеанса относится к любому виду атаки, при которой происходит утечка или компрометация конфиденциального токена или идентификатора сеанса. Злоумышленник получает действительный идентификатор сеанса после того, как он связан с законным сеансом пользователя. Используя украденный токен сеанса, злоумышленник может получить права пользователя и доступ к веб-сервису. Токен сеанса может быть скомпрометирован разными способами. В зависимости от метода эксплуатации перехват сеанса может называться по-разному. Некоторыми распространенными методами перехвата сеанса являются прогнозирование сеанса, перехват сеанса, межсайтовый скриптинг и захват сеанса.

Сначала законный клиент передает свои учетные данные веб-серверу, и сервер отвечает подтверждением после проверки учетных данных. Как только клиент аутентифицируется, между ними устанавливается сеанс, и они начинают общаться. Злоумышленник может перехватить сеанс между клиентом и сервером, используя один из методов, описанных далее в этом разделе, и начать выдавать себя за клиента с информацией о своем сеансе. Выполнив эти шаги, злоумышленник может получить разрешение на доступ клиента.

В общем, существует много способов защиты от перехвата сеанса. Одним из способов является передача идентификатора сеанса для защищенного содержимого через защищенное соединение. Следовательно, перехват сеанса можно предотвратить с помощью различных идентификаторов сеанса при переключении между безопасным и незащищенным содержимым, т. е. сеансами с проверкой подлинности и сеансами без проверки подлинности или незащищенными соединениями http и https. Еще одна мера предосторожности при перехвате сеанса — повторная аутентификация пользователя перед любыми конфиденциальными действиями. Таким образом, любой злоумышленник, выдающий себя за действительного пользователя, не может двигаться дальше и не может выполнять ограничительные действия. Кроме того, каждый сеанс может иметь максимальное время жизни. После активности

в течение определенного времени пользователю может быть предложено повторно пройти аутентификацию.

Следовательно, злоумышленник не может продолжать использовать захваченный сеанс в течение очень длительного периода времени. Для систем, использующих метод распространения сеанса GET, SID следует часто регенерировать, чтобы избежать раскрытия SID в течение более длительного времени. Правильно определить перехват сеанса практически невозможно, так как от клиента передается очень мало атрибутов. Даже в этом случае можно принять некоторые меры предосторожности, чтобы затруднить перехват сеанса. Некоторыми сведениями о клиенте, которые можно использовать для обнаружения попытки перехвата сеанса, являются пользовательский агент клиента и IP-адрес. Пользовательский агент — это заголовок ответа HTTP, отправляемый из браузера, который включает в себя имя браузера, его версию и операционную систему. Обычно пользовательский агент не меняется через короткие промежутки времени.

Для создания SID можно использовать надежные криптографические функции с надежным случайным ключом. Идентификатор сеанса, длина которого превышает 128 бит, можно считать безопасным. Более того, шифрование идентификатора сеанса может в некоторой степени защитить веб-систему. Если данные, передаваемые между сервером и клиентом, зашифрованы, данные, полученные при передаче, становятся нечитаемыми. Следовательно, любой вид перехвата не плодотворен. Однако любое шифрование не может предотвратить атаки перехвата сеанса. Эффективно только использование сильного криптографического алгоритма. Использование связи TLS для зашифрованной передачи может предотвратить атаки такого типа. Сообщения об ошибках и трассировки стека, представляемые веб-системой при возникновении непредвиденных событий, также должны быть надлежащим образом очищены, чтобы они не раскрывали никакой информации или SID. Часто злоумышленники могут собирать важные данные о системе, вызывая ошибки.

Однако пользовательский агент легко предсказуем, и значение может не быть постоянным в системе с несколькими прокси-серверами. В системе такого типа изменение пользовательского агента может вызвать ложную тревогу. Другая используемая информация, IP-адрес клиента, в настоящее время может часто меняться, и поэтому нецелесообразно проверять точный IP-адрес. Лучшим решением является проверка только части IP-адреса, относящейся к подсети. Несмотря на то, что проверка IP-адреса ненадежна, она может немного усложнить работу злоумышленника. Любой пользователь, не прошедший эту проверку, должен автоматически выйти из сеанса с существующим SID, чтобы предотвратить любую попытку перехвата сеанса.

Завершение сеанса — важный аспект безопасного управления сеансом. Сеансы, которые остаются открытыми с течением времени, потребляют ресурсы сервера и представляют потенциальный риск сессионных атак. Незавершенные сеансы уязвимы для атак с захватом сеанса и потенциального

олицетворения пользователя сеанса. Особое внимание следует уделить сокращению продолжительности сеанса. Сеанс может быть завершен во многих различных ситуациях.

Хорошей практикой является предоставление пользователям возможности надежного выхода из системы. Когда пользователь решает выйти из системы, он предполагает, что никто не сможет получить доступ к этому сеансу в будущем. По сути, выбирая вариант выхода из системы, пользователь запрашивает сценарий выхода из системы, который должен быть выполнен сервером, который явно завершает сеанс. С точки зрения безопасности важно удалить данные сеанса пользователя с сервера при выполнении функции выхода из системы, а не полагаться на браузер клиента для удаления файла cookie или идентификатора сеанса. Кроме того, любое возникновение ошибки безопасности в системе должно приводить к завершению сеанса.

1.2.4 Недостаточная защита транспортного уровня

Эта уязвимость связана с тем, что веб-приложения не защищают или плохо защищают веб-приложения. Если, например, использование SSL/TLS выполняется только на этапе проверки подлинности,данные и идентификаторы сеансов могут быть представлены в сетевых потоках приложений.

Основным условием защиты сеанса является защита всех его компонентов. Самым основным правилом является защита учетных данных пользователя и передачи любой другой информации, достаточной для доступа к сеансу. После процесса аутентификации инициируется сеанс. Этот сеанс связан с пользователем через идентификатор сеанса. Идентификатор сеанса действует как временный пароль, пока сеанс действителен. Поэтому идентификатор сеанса должен быть криптографически стойким, безопасным и защищенным от перехвата или подделки. Безопасность сетевого соединения между пользователем и сервером является еще одним важным вопросом.

С точки зрения безопасности также важно обеспечить достаточную защиту информации о состоянии сеанса от несанкционированного доступа на стороне клиента и сервера. Чтобы обеспечить защиту сетевого соединения между клиентом и сервером, информация об аутентификации и данные сеанса могут передаваться по защищенному соединению. Безопасность транспортного уровня может обеспечить такое безопасное, аутентифицированное и зашифрованное соединение.

Transport Layer Security (TLS) [12] — это стандартный протокол безопасности, который используется для создания безопасного соединения на транспортном уровне между устройствами. В этом протоколе используются криптографические методы для шифрования сетевых соединений над транспортным уровнем, чтобы обеспечить безопасный канал связи по незащищенной сети. TLS — это протокол, который обычно используется для защиты передачи конфиденциальных учетных данных пользователя и маркеров сеанса. Он основан на предыдущем проприетарном протоколе уровня защищенных сокетов (SSL), разработанном Netscape.

TLS использует как асимметричную, так и симметричную криптографию для обеспечения безопасности. TLS может использовать набор алгоритмов криптографической подписи и шифрования, называемых наборами шифров, для транзакций. В начале соединения TLS согласовывает, какой пакет будет использоваться. Когда пользователь подключается к серверу с поддержкой TLS, сервер отвечает своим цифровым сертификатом. Цифровой сертификат — это сообщение, в котором указывается личность и открытый ключ объекта, подписанный доверенным органом. Открытые ключи официальных доверенных органов обычно встраиваются в клиентские браузеры для проверки подлинности цифровых сертификатов веб-серверов. Браузер клиента проверяет цифровой сертификат, отправленный сервером, проверяя, подписан ли он доверенным органом, и проверяет подлинность сервера. Затем клиентский браузер выполняет зашифрованный обмен с сервером для создания общего ключа. Асимметричная криптография с открытым ключом в основном защищает обмен общим секретом, который используется для симметричной криптографии. Общий секрет используется для шифрования передачи данных между клиентом и сервером. После обмена общим секретом сеанс между клиентом и сервером продолжается в зашифрованном виде.

TLS предоставляет функцию возобновления сеанса. Через возобновление сеанса устройства могут возобновить ранее созданный сеанс TLS, сохранив состояние сеанса, например набор шифров и главный секрет, в билете на стороне клиента. Билет шифруется ключом, известным только серверу. Клиент сохраняет этот билет и главный секрет. Когда клиент хочет возобновить сеанс, он включает билет в запрос к серверу. Злоумышленник с украденным билетом не может возобновить соединение, так как билет зашифрован и секретный ключ ему неизвестен. Возобновление сеанса TLS может только возобновить соединение TLS, но не может возобновить транзакцию, которую защищало соединение TLS. Это удобно, потому что создание TLS-соединений может быть затруднено для легковесных клиентов. Еще одна особенность TLS — возможность предоставлять клиентские сертификаты. Клиент может подтвердить свою личность, предъявив свой сертификат и ответив на зашифрованное сообщение. Таким образом, клиент может подтвердить свою личность, даже не раскрывая свой секретный ключ, то есть закрытый ключ. Использование клиентского сертификата может смягчить проблему захвата учетных данных злоумышленником. Однако закрытый ключ представляет собой длинную серию случайных двоичных данных, которые невозможно запомнить. Закрытый ключ необходимо где-то хранить, обычно на компьютере пользователя. Это ограничивает мобильность клиента при использовании сертификата в качестве учетных данных для аутентификации.[14] Кроме того, обработка миллионов клиентов в центрах сертификации непроста. TLS обеспечивает безопасную передачу данных с использованием зашифрованного канала связи. Он также аутентифицирует серверы и, при необходимости, клиентов и позволяет подтвердить личность участвующих сторон в защищенной связи. TLS также обеспечивает целостность передаваемых данных

с помощью проверки целостности. Помимо защиты данных, TLS может защищать от атак типа «человек посередине» или повторного воспроизведения.

Сеанс привязывается к клиентскому устройству путем сохранения идентификатора сеанса на устройстве. Когда пользователь решает выйти из системы, уничтожается только сеанс, привязанный к этому устройству. Часто приложения не предоставляют возможность аннулировать все сеансы, связанные с конкретным пользователем. Один из способов реализовать эту опцию глобального выхода из системы — поддерживать случайное число, известное как соль, для каждого пользователя на стороне сервера и использовать эту соль при создании идентификатора сеанса пользователя. Если пользователь выбирает локальный выход из системы, только SID, привязанный к этому устройству, может стать недействительным. С другой стороны, если пользователь решит выйти из системы глобально, сервер может просто изменить соль пользователя на новую. Таким образом, сервер не распознает SID этого пользователя, созданного с использованием предыдущей соли, и все предыдущие сеансы этого пользователя будут автоматически аннулированы. Более того, сервер может поддерживать общую соль и использовать ее при создании всех идентификаторов сеанса. Изменение общей соли приведет к прекращению всех сеансов всех пользователей, которые обслуживал сервер.

1.2.5 SQL-инъекция

Это атака, при которой хакер использует кусок кода SQL («Язык структурированных запросов») для манипулирования базой данных и доступа к потенциально важной информации. Это один из наиболее распространенных и угрожающих типов атак, поскольку он потенциально может быть использован для нанесения вреда любому веб-сайту, использующему базу данных SQL.

В стандартных процессах программного обеспечения запрос SQL по сути представляет собой запрос, отправляемый в базу данных (компьютеризированное хранилище информации) для определенного типа действия или функции, например, запроса или выполнения данных, когда информация о соединении отправляется через веб-форму, чтобы пользователь мог получить доступ к сайту.

Как правило, этот тип веб-формы предназначен для приема типов данных, таких как имя или пароль. Когда эта информация добавлена, она сверяется с базой данных, и, если она совпадает, пользователь может войти. В противном случае ему будет отказано в доступе.

Потенциальные проблемы могут возникнуть, потому что большинство веб-форм не имеют возможности останавливать ввод дополнительной информации в формы. Хакеры могут использовать этот недостаток и использовать поля ввода формы для отправки собственных запросов в базу данных. Это может потенциально позволить им выполнять несколько видов злонамеренных действий, от кражи конфиденциальных данных до манипулирования информацией из базы данных для своих собственных целей.

Веб-скрипт может быть разработан со слабостью или ошибкой что может остаться незамеченным. Например, страница, которая только предназначенный только для операций чтения, может не применить необходимый контроль, что позволит злоумышленнику выполнить привилегированные операции. SQL Injection — одна из таких атак. это приложение уровневая атака и использует неправильные стандарты кодирования [2] в веб-приложениях, которые по своей конструкции могут поддерживать SQL команды, которые будут вводиться через веб-формы. Командная инъекция похожа на SQL-инъекцию. использует веб-программу для выполнения команд, таких как вызовы операционной системы, от ее имени. Среди различных существующих типов уязвимостей программного обеспечения, инъекции команд особенно распространены [13].

Хакер бы использовать набор команд, доступный веб-программе как только они узнают его платформу, а затем выполнят такие команды, как команды оболочки или другие системные приложения или системные команды. Межсайтовый скриптинг в основном предназначен для атаки на пользователем веб-сайта и используя собственность веб-сайт, который позволяет пользователям вводить данные [11]. Здесь хакер вводит часть синтаксиса HTML или Java/VB Script синтаксис как данные в форме ввода данных, такой как страница блога.

Большинство сеансов PHP основаны на файлах cookie. Как правило, при инициации сеанса в браузер клиента отправляется файл cookie вместе с уникальным идентификатором сеанса (SID). Чтобы персонализированные

пользовательские данные хранились на сервере, информация о пользователе должна где-то храниться. В зависимости от количества пользователей и размера пользовательских данных данные могут храниться в файле или на сервере базы данных. По умолчанию переменные сеанса хранятся в локальном файле на веб-сервере, соответствующем уникальному SID. Файл сеанса содержит информацию о сеансе в виде массива. Всякий раз, когда требуется переменная сеанса, сервер извлекает переменные из файла с именем SID клиента.

Изменить значение переменной сеанса легко. Присвоение нового значения обновляет переменную сеанса. PHP позволяет хранить любые переменные в качестве данных сеанса, т. е. простую переменную, массив, объект и т. д. Содержимое данных сеанса сериализуется и сохраняется где-то снаружи в виде двоичной строки после каждого запроса. Строка извлекается из хранилища и десериализуется, когда данные сеанса используются в следующем запросе. Данные сеанса в PHP не могут хранить какие-либо ресурсы, например дескрипторы соединения. В PHP хранилище сеанса для пользователя блокируется исключительно, когда скрипт вызывает функцию session_start(), и данные сеанса хранятся исключительно этим процессом до тех пор, пока скрипт не закроет сеанс, вызвав session_close() или session_write_close(), когда запрос закончен. Когда одновременные запросы выполняются для одного и того же сеанса, они обрабатываются последовательно при доступе к данным сеанса.

По умолчанию PHP хранит данные сеанса в файловой системе. Это хранилище файлов по умолчанию для данных сеанса можно изменить, изменив session.save_handler в файле конфигурации php.ini. Если опция установлена как mm, то данные сеанса будут храниться в памяти. Хранение данных сеанса в памяти увеличивает производительность. Сессии также могут храниться в базе данных. Хранилище базы данных обеспечивает большую масштабируемость и лучшую управляемость сеансами. Распределенную базу данных можно использовать для синхронизации данных сеанса между распределенными серверами. Другой вариант — хранить данные сеанса в собственной базе данных PHP с именем SQLite.

1.2.6 Распределенная атака типа «отказ в обслуживании» (DDoS)

Мы обсудили риски, влияющие на веб-приложения и опирается на две классификации механизмов безопасности ввести новое, третье измерение безопасности. Два классификация: сетецентрическая и хост-ориентированная. Безопасность и включают инструменты и методы, такие как контроль доступа к сети и антивирусное программное обеспечение.

У нас есть установлено, что, насколько эти методы имеют решающее значение в предприятия им не хватает необходимой глубины для всестороннего анализа общей безопасности приложений. Главный причина в том, что корпоративное приложение обычно охватывает на нескольких серверах в зависимости от количества уровней. Поэтому мы классифицировали механизмы безопасности как либо ориентированный на инфраструктуру, либо ориентированный на приложения о том, какой актив находится под залогом. У первого будет сетецентрические и хост-ориентированные механизмы безопасности в то время как первый будет состоять из методов, которые пытаются обеспечить целостную защиту приложения независимо от количества местоположений или серверов, которые оно охватывает физически или виртуально.

Чтобы описать безопасность, ориентированную на приложения, мы выделили ряд усилий и исследований в этом направлении. и, объяснил их недостатки. По сути, мы описал, что ориентированная на приложения безопасность веб-приложения требует учета проектных соображений. в обеспечении полной проверяемости операций бизнес-процесс в приложении. Сделав это, зарегистрируйтесь анализ и аудит безопасности улучшены и, в конце концов, должны влиять на безопасность системы в целом. Установив потребность в ориентированном на приложения механизмов безопасности, должна быть формальная спецификация требований к ним.

Определенно целостное ведение журнала и мониторинг являются необходимой отправной точкой. за которыми следуют более целенаправленные целостные меры безопасности например контроль доступа.

Сеанс может быть завершен пользователем, когда он выбирает выход из системы. Однако сервер никогда не может быть уверен, что пользователь всегда выйдет из системы после завершения использования приложения. По этой причине серверу необходимо удалить сеансы, которые не использовались в течение определенного периода времени. Этот тип завершения сеанса, также известный как относительный тайм-аут, может быть достигнут путем установления определенного ограничения по времени бездействия сеанса. Любой сеанс, который не был активен в течение разумного времени, удаляется из хранилища сеансов. Относительный тайм-аут — полезный способ очистки устаревших сеансов. Это также предотвращает атаки с раскрытием сеанса, когда злоумышленник может завладеть сеансом, пользователь которого не вышел из системы явным образом. Чтобы реализовать неактивный тайм-аут, приложение может отслеживать последний запрос, полученный от пользователя, и вычислять прошедшее время через равные промежутки

времени. Когда прошедшее время достигает времени относительного тайм-аута, приложение перенаправляет пользователя на страницу, которая разрушает сеанс. Время последнего запроса пользователя сбрасывается при любой активности в веб-браузере.

1.2.7 Неправильная настройка безопасности

Инфраструктура, поддерживающая веб-приложение, включает в себя множество устройств и программного обеспечения: серверы, брандмауэры, базы данных, операционные системы и приложения. Все эти элементы должны быть настроены и защищены безопасным образом. Одной из основных причин плохого системного администрирования является отсутствие надлежащей подготовки тех, кто отвечает за управление веб-приложениями и базовой инфраструктуры.

Часто сеанс уничтожается, когда пользователь выходит из приложения. Хотя данные сеанса являются временными и их не всегда нужно очищать явно, в некоторых случаях необходимо удалить данные сеанса. Функция session_destroy() используется для уничтожения всех данных сеанса, связанных с текущим сеансом на сервере. Однако эта функция не сбрасывает какие-либо глобальные переменные, привязанные к сеансу или соответствующему файлу cookie в клиенте. Для очистки всех зарегистрированных переменных сеанса используется функция session_unset().

С точки зрения безопасности рекомендуется аннулировать сеанс на стороне сервера, а также на стороне клиента, когда пользователь выходит из системы. Для этого запись данных сеанса пользователя удаляется из хранилища сеансов сервера, а также аннулируется файл cookie, находящийся на клиенте.

Более того, несмотря на то, что сеанс является активным, сервер не может определить, используется ли сеанс законным пользователем или он был украден злоумышленником. Поэтому приложение должно реализовать абсолютный тайм-аут, который ограничивает общую продолжительность сеанса. Абсолютный тайм-аут полезен в случае, когда злоумышленник, похитивший идентификатор сеанса, пытается сохранить его действительным, делая периодические запросы.

Ограничение времени относительного и абсолютного времени ожидания может варьироваться в зависимости от среды приложений и требований безопасности. Как правило, от 15 до 30 минут бездействия сеанса достаточно, чтобы завершить сеанс. Для абсолютного тайм-аута ограничение по времени может варьироваться от 6 до 12 часов, в зависимости от требуемого уровня безопасности системы.

Для большей безопасности рекомендуется использовать оба вида тайм-аута в системе. Кроме того, для удобства использования можно использовать двухуровневый механизм тайм-аута. На первом уровне приложение может временно заблокировать сеанс пользователя по истечении определенного периода времени и запросить у пользователя его учетные данные. Приложение будет ожидать учетных данных пользователя в течение другого периода времени. Если клиент не предоставит учетные данные вовремя, приложение

безвозвратно удалит информацию о сеансе с сервера. Таким образом, сеанс пользователя блокируется через определенный период времени, чтобы предотвратить его использование злоумышленником, но все же дает пользователю возможность продолжить сеанс, предоставив свои учетные данные.

1.3. Анализ возможностей подходов для обеспечения безопасности Веб-сайта

Разработчики и администраторы, отвечающие за ИТ-безопасность, могут применять различные методы для борьбы с угрозами, направленными на веб-приложения. Далее будут приведены несколько основных способов обеспечения безопасности, учитывая следующие цели:

- Предотвращение уязвимостей: предотвращение внедрения уязвимостей путем применения строгих методов разработки.
- Устранение уязвимостей: выявление уязвимостей и их устранение с помощью методов проверки и тестирования.
- Предотвращение, обнаружение и устойчивость к вторжениям: Защита системы во время работы от атак и вторжений путем внедрения защитных барьеров (брандмауэров и систем обнаружения), позволяющих приложению обеспечить правильную службу, несмотря на атаки.
- Оценка: оценка воздействия уязвимостей и атак, а также эффективности механизмов защиты [5].

1.3.1 Предотвращение уязвимостей

Предотвращение уязвимостей обычно основывается на использовании строгих процессов, правил разработки и эксплуатации для снижения риска внедрения уязвимостей на различных этапах жизненного цикла приложения.

Независимо от того, является ли разработчик новичком или уже опытным в области безопасности веб-приложений, создание нового веб-сайта или обеспечение безопасности существующего может быть затруднено, контролируя все риски без применения соответствующих процессов и инструментов. По этой причине важно учитывать аспекты безопасности на всем этапе разработки приложения, особенно на ранних этапах.

Во-первых, необходимо соблюдать требования безопасности приложений, т. е. указать, что означает безопасность, потому что гораздо выгоднее разрабатывать приложение, защищая его с самого начала, а не устраняя его недостатки после.

Во-вторых, при написании кода приложения необходимо соблюдать стандартные проверки безопасности. Например, набор стандартных элементов управления может значительно облегчить разработку приложений на Java, PHP и Python.

Применение этих правил может внести существенный вклад в повышение безопасности веб-приложений. Однако, учитывая сложность прикладных программ и используемых технологий, эти меры недостаточны для обеспечения разработки без ошибок и должны дополняться другими средствами.

1.3.2 Устранение уязвимостей: проверка и тестирование

Устранение ошибок направлено на выявление и исправление остаточных уязвимостей, возникших при разработке приложения. Оно основано на различных дополнительных методах проверки, которые могут использоваться как с активацией приложения, так и без него. Проверка без фактической активации - это статическая проверка, которая может касаться различных этапов разработки (спецификация, проектирование, кодирование). В другом случае это динамическая проверка, обычно основанная на методах тестирования, в частности, на тестировании на проникновение. Необходимо применять оба метода проверки так часто, как это необходимо, поскольку они дополняют друг друга. Кроме того, из-за растущей сложности веб-приложений разработка средств поддержки проверки становится необходимой.

Несмотря на то, что данные сеанса удаляются с сервера с помощью функции session_destroy(), когда пользователь выходит из системы, сервер никогда не может быть уверен, что пользователь всегда выйдет из системы. Следовательно, необходимо использовать встроенный в PHP механизм сборки мусора, который очищает сервер от незавершенных, неиспользуемых файлов сеанса. Эта функция помогает удалить избыточные файлы сеансов с сервера и снижает риск раскрытия сеанса. Механизм сборки мусора PHP имеет два

параметра: session.gc_maxlifetime и session.gc_probability. Оба эти параметра можно определить в конфигурационном файле php.ini. Параметр gc_maxlifetime определяет период бездействия в секундах. Механизм PHP запускает процесс сборки мусора при инициализации сеанса и исследует каждый сеанс. Сеансы, к которым не было доступа в течение заданного периода времени, удаляются из хранилища сеансов. В файловом механизме управления сеансом время обновления файла сеанса используется как индикатор последнего доступа. Следовательно, PHP необходимо изменить время обновления файла сеанса при записи или чтении переменных сеанса. Другой параметр gc_probability определяет процентную вероятность того, что процесс сборки мусора будет активирован. Поскольку процесс сборки мусора может увеличить нагрузку на сервер при большом количестве пользователей, для этого свойства устанавливается сбалансированное процентное значение в зависимости от требований приложения. Кроме того, настройка PHP по умолчанию будет поддерживать активность файла cookie сеанса неопределенно долгое время, пока клиентский браузер не будет закрыт. Это поведение можно изменить, изменив значение session.cookie_lifetime в файле php.ini. Значение этого параметра определяет время жизни файла cookie, отправляемого клиенту.

1.3.3 Предотвращение, обнаружение и устойчивость к вторжениям

С развитием Интернета и веб-технологий многие услуги теперь предлагаются пользователям по всему миру. Был запущен широкий ряд приложений, начиная от информационные сайты, социальные сети, электронная коммерция и Программное обеспечение как услуга (SAAS). Как упоминалось в [1], это рост может быть связан с программным обеспечением веб-приложений архитектура, отвечающая потребностям распределенной гипермедиа-системы масштаба Интернета. можно интегрировать и создавать больше приложений и компонентов в Интернете гораздо проще по сравнению с традиционным рабочим столом программные подходы. Наряду с этой успешной волной интернет-приложений, было обнаружено и действительно использовано множество уязвимостей, что вынуждает поставщиков услуг продолжать инвестировать много усилий и ресурсов для мониторинга и содержащие нарушения. Примеры атак включают хакеров использование неподходящих стандартов кодирования веб-приложения, работающего на веб-сервере, или встроенных уязвимостей самого веб-сервера [2]. Важно быть на шаг вперед в обеспечении безопасности таких систем.

Несколько способов были разработаны, включая методологии аудита безопасности и архитектуры безопасности для улучшения обнаружения и предотвращения рисков. Такие методы, как установка интеллектуальных брандмауэров, безопасная сквозная связь через виртуальный Уровни частных сетей и безопасных сокетов были используются для повышения безопасности и действительно были приняты для нескольких типов веб-приложений. Также в соответствии с тот факт, что электронные записи, такие как компьютер и сетевые журналы считаются наиболее важными данными в цифровая

криминалистика [3-5], инструменты и методы мониторинга широко используются, включая регистрацию транзакций, таких как системные журналы, журналы базы данных и веб-сервер. журналы, а также использование интеллектуального мониторинга журналов и программное обеспечение для анализа. Однако все эти методы и усилия не гарантировали безопасность систем или обнаружение покушения на них. В соответствии с тремя принципами информационной безопасности; Конфиденциальность, целостность и доступность, одиннадцать параметры защиты информационных активов рисуются [6]. Идея этих измерений заключается в том, что необходим целостный подход, при котором все измерения используются для повысить безопасность. В частности, два из этих измерений осуществляют мониторинг и оценку, где наблюдения за система и выполняются соответствующие действия.

Первый измерение мониторинга также было отмечено в [7] как столп информационной безопасности, особенно важный для веб-приложений, где мониторинг веб-серверов, базы данных серверы и серверы аутентификации посредством анализа журнала файлы. Мы объясняем риски, влияющие на веб-приложения и значимость ведения журналов и анализа журналов, насколько речь идет о мониторинге. Затем мы подводим итоги классификаций механизмов безопасности, на которые были направлены предыдущие усилия по реализации безопасности. был сосредоточен.

Большинство веб-приложений поддерживают данные сеанса, специфичные для пользователя, и обеспечивают контролируемый доступ к конфиденциальным ресурсам. По этой причине приложения требуют, чтобы их пользователь идентифицировал и аутентифицировал при доступе к ограниченным частям приложения. Безопасный метод аутентификации — это точка входа для безопасного управления сеансом в приложении. Безопасная система аутентификации вместе с механизмом защиты учетных данных пользователя может предотвратить вторжение в сеанс. В некотором смысле безопасная аутентификация также может повысить безопасность сеанса. Веб-приложения могут использовать соединение TLS при передаче учетных данных пользователя от клиента к серверу. Поскольку соединение будет зашифровано в TLS, учетные данные будут защищены от подделки или прослушивания злоумышленником. Когда пользователь аутентифицирован, сервер обычно создает маркер сеанса или идентификатор сеанса для отслеживания пользователя и поддержания его аутентифицированного статуса. Затем этот токен сеанса действует как временный пароль для сеанса пользователя. Когда веб-приложение использует подключение TLS для конфиденциальных ресурсов с проверкой подлинности, веб-приложение должно быть правильно спроектировано для обработки маркеров безопасности. При перемещении между защищенными и незащищенными областями приложения важно обрабатывать сеанс таким образом, чтобы гарантировать, что по незашифрованным соединениям HTTP не будет передан ни один маркер безопасности.

1.3.4 Оценка методов защиты

Различные методы, описанные в предыдущих разделах, играют очень важную роль в обеспечении безопасности веб-приложений. Безопасность этих приложений зависит от уровня угрозы, предназначенной для этих приложений, и эффективности контрмер, применяемых для борьбы с этими угрозами. Уровень угрозы зависит от количества остаточных уязвимостей в этих приложениях и частоты попыток атак с целью использования этих уязвимостей. Успех или нет атаки будет зависеть от эффективности защитных механизмов.

Далее будет рассмотрена методология, которая обычно используется для проведения этих оценок.

Методология

Оценка системы обнаружения вторжений обычно требует фазы экспериментов, чтобы убедиться, что она правильно адаптируется к окружающей среде, которую она защищает. Чтобы получить надежные и легко достижимые оценки на практике, необходимо разработать строгие и автоматизированные экспериментальные протоколы. Исторически первые подходы были основаны на неофициальных экспериментах групп экспертов по безопасности, называемых “redteams”, целью которых было попытаться обойти защитные механизмы, чтобы поставить под угрозу безопасность рассматриваемых систем. Однако возникла необходимость в разработке более структурированных подходов, основанных на систематическом экспериментальном протоколе, чтобы строго характеризовать эффективность IDS.

При управлении рисками используются несколько средств контроля. обычно определяется с целью снижения рисков из-за возможных уязвимостей в системе. Давайте рассмотрим пример следующего риска; «несанкционированный доступ к панели управления пользователями». Такой риск может могут быть использованы из-за таких уязвимостей, как слабый пароль или передача данных между администраторами и сервером в виде открытого текста. Чтобы смягчить это, контроль будет заключаться в том, чтобы весь доступ к удаленному администрированию был разрешено только через шифрование с помощью SSL. Другой был бы что все пароли администратора должны быть надежными и менялись регулярно. Впоследствии аудит безопасности для оценить, реализованы ли средства контроля и работают ли они в системе будет заключаться в формировании контрольной деятельности для каждый проводит несколько тестов, чтобы посоветовать одно и то же. В нашем Например, контрольных действий может быть два: 1) Проверка все журналы веб-сервера страниц администратора и 2) проверить надежность пароля и политику истечения срока действия. Для каждого из контрольных действий одним тестом будет открытие всех веб-серверов. журналы и наблюдайте за IP-адресом клиента и уровнем шифрования SSL для разделов администратора и выполнить проверку надежности пароля и срок годности соответственно. Давайте рассмотрим веб-среду с вышеуказанным средства контроля и обсудить возможную лазейку. А хакер получает доступ к общедоступной части веб-сайта и, эксплуатирует уязвимую страницу на сайте, использует SQL инъекции для изучения пользовательских таблиц в базе данных и вносит некоторые изменения в базу данных.

Хакер создает нового пользователя X с действительно надежным паролем и назначает привилегии уровня администратора, а затем пытается войти в систему в раздел администрирования пользователей с этими новыми данными с использованием зашифрованного SSL. Попытка входа работает! Более важно, когда аудитор проводит свои тесты, они выявляют что, действительно, в администрировании пользователей доступа не было разделы в соединениях без SSL, а также что пароли достаточно надежны, чтобы предотвратить взлом паролей. Даже в том случае, если какие-то действия хакера поднимать флаги в процессе аудита, нет возможности целостно определить точную степень нанесенного ущерба или влияние. Немедленное смягчение проблемы заключаться в разработке обширных элементов управления во всем веб-решении включая надлежащее планирование доступа пользователей к базе данных, стандарты веб-разработки и так далее. В частности, в нашем примере выше дополнительный элемент управления для ограничения административного доступа к определенному IP-адресу уменьшит риск.

Тем не менее, это все еще не останавливает другие риски, связанные с с SQL-инъекцией. Безопасность, ориентированная на приложения, будет иметь большое значение в решение этой задачи в три раза. Во-первых, действия будут быть зарегистрированным, показывающим, какие страницы посетил определенный пользователь и какие объекты базы данных были доступны и

любые другие информацию о размерах, которая облегчит построение Контекст. Эта контекстная информация, собранная за период Затем время будет проанализировано для определения контекстных карт. которые представляют модели использования системы. Было бы легко распознавать нормальные или разрешенные контекстные карты и отмечать те, которые отличаются от нормы, что позволяет обнаружить потенциальных атак. Во-вторых, это позволило бы быстро определить степень ущерба после того, как он произошел. Это происходит из-за присущей ему способности записывать пользовательские поведение или активность в нескольких веб-модулях.

Такой информация, использованная в нашем предыдущем примере, покажет действия этого хакера и меры контроля могут быть добавлены соответственно. В-третьих, его можно использовать как средство предотвращения атаки. Инструмент, способный считывать и извлекать информацию о состоянии из контекстных журналов, может распознавать странные привычки просмотра и остановить их. например, публичный обычно ожидается, что пользователь будет следовать определенному шаблону и контекст.

Вывод по первому разделу:

1.Информационная среда – это среда, в которой реализуется совокупность информационных процессов и средств для обслуживания объектов инфраструктуры территориальных образований в интересах обеспечения баланса спроса и предложения на рынке информационных услуг

Существующие подходы позволяют обеспечивать ИБ только в случае монозапросов [6]. Поэтому самостоятельный научный и практический интерес представляет разработка моделей и методов, гарантирующих обеспечения ИБ применения веб-технологий в условиях применения сложных запросов.

Для этого целесообразно использовать естественно-научный подход, разработанный научной школой «Системная интеграция процессов государственного управления». Эта научная школа зарегистрирована в Реестре ведущих научно-образовательных школ Правительства СПБ. Процесс обеспечения ИБ строится на основе системной интеграции

- целевого информационного процесс;
- процесса образования угрозы;
- процесса идентификации угрозы;
- процесса нейтрализации угрозы;
- показателя информационной безопасности.

В результате системной интеграции формируется условие существования процесса обеспечения ИБ. Это условие в итоге позволяет сформировать одно уравнение с двумя неизвестными:

- обобщённая характеристики процесса идентификации угрозы;
- обобщённая характеристика нейтрализации угрозы.

Формирование обобщённых характеристик этих двух процессов исходя из требуемого уровня ИБ позволяет удовлетворить условие существования процесса обеспечения ИБ и тем самым обеспечить требуемую ИБ. В целом, этот результат позволил перейти к разработке аналитической математической модели обеспечения ИБ веб-сайта в условиях деструктивных воздействий среды.

Злоумышленники могут воспользоваться различными классами веб-уязвимостей. Однако, сегодня никто не может утверждать, что существуют надежные механизмы компьютерной безопасности, с одной стороны, потому, что не все уязвимости известны, с другой стороны, потому, что системы и Технологии быстро развиваются с каждым разом с большим количеством новых уязвимостей, вследствии этого необходимо оценить эффективность новых средств безопасности или совершенствования существующих технологий. Поэтому для того, чтобы обеспечить адекватную безопасность веб-сайта, необходимо иметь адекватную математическую модель решения человека.

2. Разработка аналитической, математической модели в условиях глобальной сети интернет

2.1 Общий подход к разработке метода

В основе процесса обеспечения информационной безопасности всегда лежит решение человека. Решение принимается человеком на основе модели. Под моделью объекта понимается описание или представление соответствующего объекта, позволяющее получить основные свойства и характеристики о нем. Поэтому решение – модель процесса, с которым работает человек. Процесс - это объект в действии при фиксированном предназначении [7].

Для гарантированного достижения цели деятельности при обеспечении безопасности необходимо уметь формировать процессы с наперёд заданными свойствами. Не имея условия существования этих процессов, мы не можем их формировать. А это не позволяет формировать процессы с наперёд заданными свойствами. [8-9] Не имея такой возможности, мы не можем гарантированно достигать цель деятельности. А это возможно только на основе методологии обеспечения безопасности [10].

Не имея методологических основ решения проблем функционирования потенциально опасного объекта в виде условий существования процесса, мы не можем гарантировать достижение цели деятельности. Эта ситуация породила фундаментальную проблему: «Результаты деятельности по решению задач функционирования систем обеспечения безопасности не соответствуют ожиданиям лица, принимающего решения». Лицо, принимающее решение, действует на основе трех категорий. Это Система. Модель. Предназначение.

Для разработки системы известны два подхода. Разработка системы на основе анализа. Для решения задачи 1 предлагается использовать для синтеза - закон сохранения целостности объекта (ЗСЦО), который обеспечивает достижение цели функционирования систем обеспечения безопасности. (ЗСЦО – это устойчивая, объективная, повторяющаяся связь свойств объекта и свойств его действий при фиксированном предназначении) [11].

Задача 2. Лицо, принимающее решения, осуществляет функционирование систем обеспечения безопасности на основе модели. Для этого нужно уметь синтезировать адекватные модели. Достижение цели функционирования систем искусственного интеллекта возможно только на основе правильно построенной системы (ППС) и адекватной модели. В известных публикациях такой подход к функционированию систем обеспечения безопасности отсутствует. Поэтому вся конструкция ППС реализована на основе ЗСЦО, что подтверждает целесообразность рассмотрения ЗСЦО как условия существования функционирования систем обеспечения безопасности. Модель функционирования систем обеспечения безопасности основана на системной интеграции четырех процессов. Целевой процесс. Формирование проблемы. Процесс распознания проблемы. Процесс устранения проблемы.

В данной ВКР управленческое решения преобразовано в виде математической модели, имеющей вид:

Где $\Delta t_{пп}$– среднее время обнаружения проблемы перед человеком.$\boldsymbol{\Delta t_{нп}}$ – средние нейтрализация проблемычеловеком.$\boldsymbol{\Delta t_{ип}}$- средним временем идентификация проблемы**.**

Это есть условие существование процесса управления безопасностью веб-сайта**.** Этой характеристикой является время, требуемое на прогнозирование характеристик веб-сайта($T_э$). Графическая иллюстрация процесса представлена рисунке5.

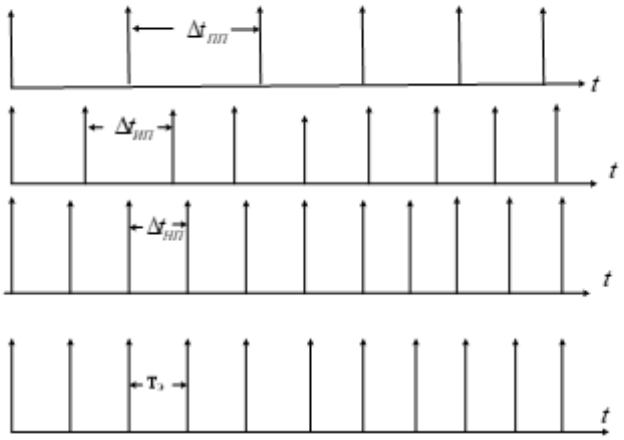

Рисунок 5 – Диаграмма проявления базовых элементов формирования модели решения

Рассмотренные четыре процесса, представленные на диаграммах, характеризуют процесс функционирования веб-сайта.

Модель функционирования сайта можно представить виде графа на рисунке 6 с двумя состояниями: начальной – «1» и конечной – «2».

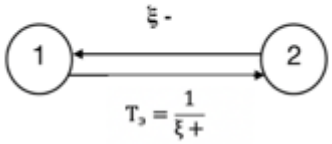

Рисунок 6 – Граф состояний

Где среднее выполнение запроса обозначено, как «$T_э = \frac{1}{\xi+}$». При функционировании возможно неудовлетворение запроса. Этот факт можно описать как частота срыва выполнения запроса «ξ -». А деятельность сайта можно описать как частота выполнения запросов, это есть «$T_э = \frac{1}{\xi+}$». Из-за деятельности злоумышленников возникает угроза разрушения функционирования сайта. Отсюда возникает проблема: «Как увязать процесс функционирования сайта с деятельностью системы обеспечения безопасности». В основу у нас будет положен естественно-научный подход (ЕНП). ЕНП определяется интеграцией свойств мышления человека, окружающего мира и познания [12].

Необходимо учесть логику функционирования системы управления веб-сайта и сократить время, необходимое специалисту для принятия решения о найденных уязвимостей веб-сайта на основе автоматического управления. Для осуществления автоматического управления необходимо получить условия существования процесса и обеспечить обратную связь,структурная схема которого представлена на рисунке 7 следующим образом:

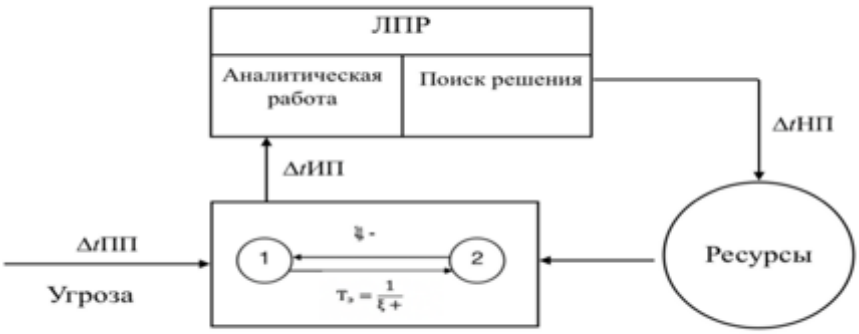

Рисунок 7– Структурная схема функционирования веб-сайта, на основе обратной связи

В результате синтез модели управления безопасностью веб-сайта преобразованв математическую модель управленческого решения следующего вида.

Когда объект управления находится в состоянии 3 под воздействием интенсивности λ, необходимо выявить угрозу. Естественно, специалист тратит некоторое время Δt_{PI} на выявление проблемы. На данном этапе идет подготовка к привлечению дополнительных ресурсов для решения проблемы. Таким образом, при анализе решения управляемая система переходит в состояние 4, где специалист распознает угрозу и понимает, какие ресурсы следует использовать для нейтрализации угрозы. Далее идет устранение угрозы, система переходит из состояния 4 в состояние 2, т. е. проблема решена. Далее на вход поступает очередная угроза, и ее надо распознавать. Процесс повторяется.

Переход системы обратно в исходное состояние характеризует ее работоспособность на множестве задач. Частота перехода системы из состояния 1 в состояние 2(ξ+) эквивалентна значению, обратному среднему времени выполнения целевой задачи, характеризующему уровень способности реагирования запроса веб-сайта к решению целевой задачи, в то время как частота (ξ -) характеризует среднюю частоту невыполнения целевой задачи, но обычно это значение должно колебаться в пределах 0,1%.

Частота перехода из состояния 4 в состояние 2 равна значению $v_2 = \frac{1}{\Delta t_{PN}}$ среднему времени нейтрализации задачи. Уровень компетентности в решении неизвестных задач зависит от соотношения v_2.

Эта логика рассуждений позволяет построить следующий график, изображенный на рисунке 8:

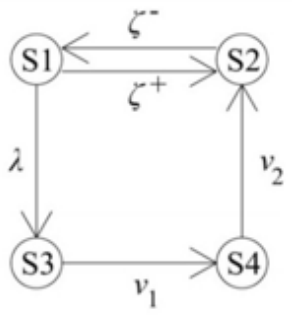

Рисунок 8– Граф состояний, процесса формирования управленческого решения

На фоне этого может быть использована система дифференциальных уравнений Колмогорова:

$$\frac{dP_i(t)}{dt} = \sum_{j=1}^{n} \lambda_{ji}(t) * P_j(t) - P_i(t) * \sum_{j=1}^{n} \lambda_{ji}(t) \quad (1)$$

где i=0,1, 2,…,n.

Конечные вероятности состояний могут быть вычислены путем решения системы линейных алгебраических уравнений, полученных из дифференциальных уравнений Колмогорова, если производные равны нулю, а вероятностные функции состояний $P_1(t)$, …, $P_n(t)$ в правой части уравнений переходят в неизвестные конечные вероятности P_1 , …, P_n. Чтобы найти точное значение P_1 , …, P_n , к уравнениям добавляется нормализующее условие $P_0 + P_1 + \cdots + P_n = 1$.

Система уравнений Колмогорова для графа состояний изображенном на рисунке 3 имеет вид:

$$\begin{cases} \frac{dP_1(t)}{dt} = -(\xi^+ + \lambda) * P_1(t) + \xi^- * P_2(t) \\ \frac{dP_2(t)}{dt} = \xi^+ * P_1(t) - \xi^- * P_2(t) + v_2 * P_4(t) \\ \frac{dP_3(t)}{dt} = \lambda * P_1(t) - v_1 * P_3(t) \\ \frac{dP_4(t)}{dt} = v_1 * P_3(t) - v_2 * P_4(t) \end{cases} \quad (2)$$

Тогда конечные вероятности могут быть вычислены путем решения системы линейных алгебраических уравнений.

$$\begin{cases} 0 = -(\xi^+ + \lambda) * P_1 + \xi^- * P_2 \\ 0 = \xi^+ * P_1 - \xi^- * P_2 + v_2 * P_4 \\ 0 = \lambda * P_1 - v_1 * P_3 \\ 1 = P_1 + P_2 + P_3 + P_4 \end{cases} \quad (3)$$

Системное решение выглядит следующим образом:

$$P_1 = \frac{v_1 * v_2 * \xi^-}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

$$P_2 = \frac{\lambda * v_1 * v_2 + v_1 * v_2 * \xi^+}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

$$P_3 = \frac{\lambda * v_2 * \xi^-}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

$$P_4 = \frac{\lambda * v_1 * \xi^-}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-} \quad (4)$$

Вероятность выявления и нейтрализации проблемы емкостью определяется следующей корреляцией:

$$P_2 = \frac{\lambda * v_1 * v_2 + v_1 * v_2 * \xi^+}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-} \quad (5)$$

В этом соотношении связаны три параметра. Таким образом установлена аналитическая зависимость обобщённых характеристик появления проблемы ($\Delta t_{\text{пп}}$), идентификации проблемы ($\Delta t_{\text{ип}}$) и нейтрализации проблемы ($\Delta t_{\text{нп}}$), возникшей при обеспечении безопасности веб-сайта.

Для анализа модели используются сетевые модели, которые представляют разновидность ориентированных графов.Вершины графа - этособытия обнаружения проблем, определяющие начало и окончание отдельных работ, а дуги в этом случае будут соответствовать работам.В данной работе используется сетевая модель так, как наглядно видно,как взаимодействует узлы друг с другом.

2.2 Проектирование сетевой модели образования угроз Веб-сайта

Задачей проектирования является то, что необходимо связать программно-аппаратную часть с безопасностью системы. Появление угрозы зачастую может сопровождаться изменением параметров функционирования информационной системы.

2.2.1 Сетевая модель угроз

В данном графе, изображенном на рисунке 9, работа веб-сайтаразделяется на 3 ветви:

1. Программное обеспечение;
2. Аппаратное обеспечение;
3. Ресурсы.

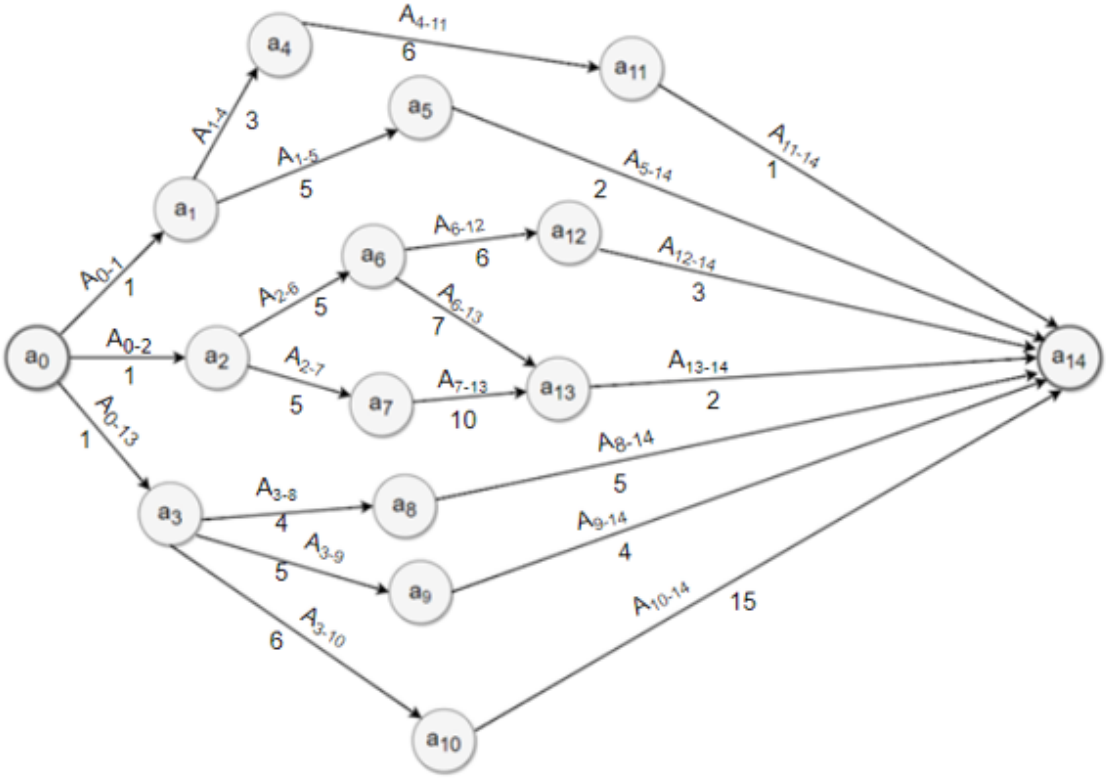

Рисунок 9 – Сетевой граф угроз

Обозначение	Наименование событий
a_0	Угроза
a_1	Угроза аппаратного обеспечения
a_2	Угроза программного обеспечения
a_3	Угроза информационным ресурсам
a_4	Разрыв кабеля
a_5	Сбой в работе оборудования
a_6	Вредоносная программа
a_7	Сбой в работе программного обеспечения
a_8	Изменение контента
a_9	Отказ в доступе
a_{10}	Несанкционированный доступ к персональным данным
a_{11}	Сбой в передачи трафика
a_{12}	Нарушение целостности информационных ресурсов
a_{13}	Прекращение работы ПО
a_{14}	Нарушение работы сайта

Таблица 1 – Перечень событий веб-сайта

Обозначение работ	Наименование работ	Время выполнения работы, нс	Предшествующие работы	Последующие работы
A_{1-2}	Начало угрозы работы еб-сайта	1	-	A_{2-3}, A_{2-4}
A_{2-3}	Изменение состояния еб-сайта	1	A_{1-2}	A_{3-12}
A_{3-12}	Анализ информации и арушение работы в истеме	5	A_{2-3}	-
A_{2-4}	Изменение состояния абеля	3	A_{1-2}	A_{4-12}
A_{4-12}	Анализ информации и арушение работы в истеме	11	A_{2-4}	-
A_{1-5}	Нарушение работы рограммного беспечения	3	-	A_{5-6}, A_{5-7}, A_{5-8}
A_{5-6}	Внедрение редоносной программы	2	A_{1-5}	A_{6-12}
A_{6-12}	Анализ информации и арушение работы в истеме	6	A_{5-6}	-
A_{5-8}	Отказ в доступе к истеме	5	A_{1-5}	A_{8-12}
A_{8-12}	Анализ информации и арушение работы в истеме	2	A_{5-8}	-
A_{5-7}	Несанкционированный ступ к системе	4	A_{1-5}	A_{7-12}
A_{7-12}	Анализ информации и арушение работы в истеме	5	A_{5-7}	-
A_{1-9}	Нарушение работы нформационных есурсов	5	-	A_{9-10}, A_{9-11}
A_{9-10}	Изменение передачи нформации к базе анных	3	A_{1-9}	A_{10-12}
A_{10-12}	Анализ информации и арушение работы в истеме	4	A_{9-10}	-
A_{9-11}	Изменение состояния лектропитания	3	A_{1-9}	A_{11-12}
A_{11-12}	Анализ информации и арушение работы в истеме	10	A_{9-11}	-

Таблица 2 – Перечень угроз веб-сайта.

2.2.2. Сетевая модель мониторингавеб-сервера

Стадия анализа, входящие в который процедуры обрабатывают полученную на стадии мониторинга информацию, сравнения с ранее полученными результатами и, на основании результатов, предоставление информации о вероятных причинах сбоев или ненадежной работе в сети. Для составления сетевой модели мониторинга системы нужно определить список работ, которые следует сделать для полного мониторинга и время, потраченное на эти работы (Таблица 3).

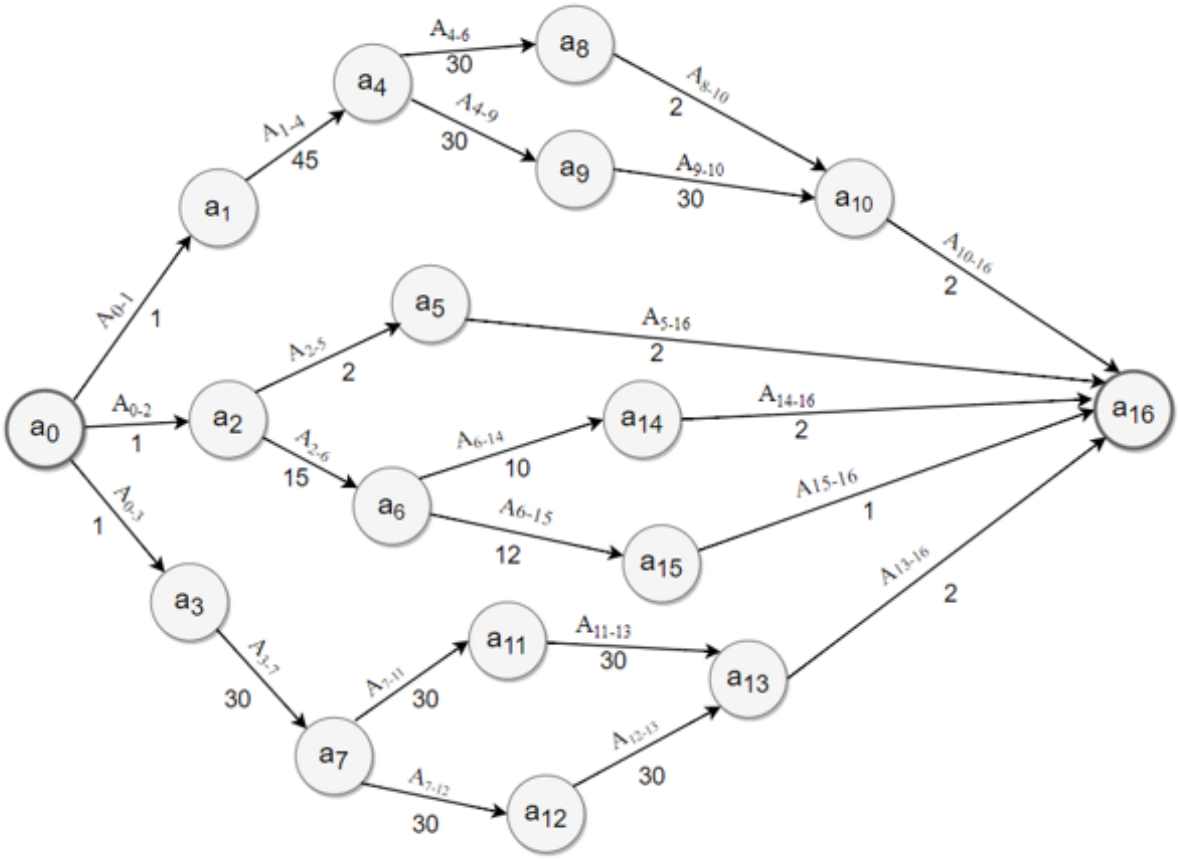

Рисунок 10 – Сетевой график мониторинга веб-сайта

Обозначение	Наименование событий
a_0	Мониторинг веб-сервера
a_1	Мониторинг дефектов аппаратной части веб-сервера
a_2	Мониторинг дефектов физического уровня сети и сетевого борудования
a_3	Мониторинг дефектов прикладного ПО
a_4	Мониторинг сбоев аппаратного обеспечения веб-сервера
a_5	Мониторинг наличия отклика сервера
a_6	Произведена диагностика состояния сетевого кабеля
a_7	Мониторинг сбоев программного обеспечения веб-сервера
a_8	Сбоев в работе аппаратного обеспечения не обнаружено
a_9	Обнаружены сбои в работе аппаратного обеспечения
a_{10}	Произведена диагностика АО
a_{11}	Сбоев в работе программного обеспечения не обнаружено
a_{12}	Обнаружены сбои в работе программного обеспечения
a_{13}	Произведена диагностика ПО
a_{14}	Измерена текущая загруженность канала связи сети
a_{15}	Измерено число ошибок передачи данных на уровне канала вязи и выяснены причины их возникновения
a_{16}	Запись в журнал тестирования

Таблица 3 – Перечень событий мониторинга веб-сайта

Обозначение работ	Наименование работ	Время выполнения работы, нс	Предшествующие работы	Последующие работы
A_{0-1}	Начался мониторинг дефектов внутри веб-сервера	1	-	$A_{1-4,}$
A_{0-2}	Начался мониторинг дефектов прикладного ПО	1	-	A_{2-5} , A_{2-6}
A_{0-3}	Начался мониторинг дефектов физического уровня сети и сетевого оборудования	1	-	$A_{3-7,}$
A_{1-4}	Диагностика АО при помощи Zabbix	45	A_{0-1}	A_{4-8} A_{4-9}
A_{2-5}	Проверка отклика сервера с использованием утилиты ping	2	A_{0-2}	A_{5-16}
A_{2-6}	Анализ и выявление причины коллизии в сети из-за дефектов кабельной системы и активного оборудования с помощью анализаторов протоколов и агентов mini-RMON	15	A_{0-2}	$A_{6-14,}$ A_{6-15}
A_{3-7}	Диагностика ПО при помощи Zabbix	30	A_{0-3}	A_{7-11},A_{7-12}
A_{4-8}	Диагностика АО при помощи Zabbix	30	A_{1-4}	A8-10
A_{4-9}	Диагностика АО при помощи Zabbix	30	A_{1-4}	A9-10
A_{6-14}	Измерение текущей нагруженности канала связи с помощью бесплатной программы «iperf»	10	A_{2-6}	A_{14-16}
A_{6-15}	Анализ и выявление причины коллизии в сети из-за высокого уровня утилизации канала связи с помощью сетевых анализаторов	12	A_{2-6}	A_{15-16}
A_{7-11}	Диагностика веб-сервера при Zabbix	30	A_{3-7}	A_{11-13}
A_{7-12}	Диагностика АО при помощи AIDA64ExtremeEdition	30	A_{3-7}	A12-13
A_{8-10}	Анализирование информации и запись ее в журнал тестирования	2	A_{4-8}	A_{10-16}
A_{9-10}	Диагностика АО при помощи Zabbix	30	A_{4-9}	A_{10-16}
A_{11-13}	Диагностика АО при помощи Zabbix	30	A_{7-11}	A_{13-16}
A_{12-13}	Диагностика АО при помощи AIDA64ExtremeEdition	30	A_{7-12}	A_{13-16}
A_{5-16}	Анализирование информации и запись ее в журнал тестирования	2	A_{2-5}	-
A_{10-16}	Анализирование информации и запись ее в журнал тестирования	2	A_{8-10}	-
A_{13-16}	Анализирование информации и запись ее в журнал тестирования	2	A_{11-13},A_{12-13}	-
A_{14-16}	Анализирование информации и запись ее в журнал тестирования	2	A_{6-14}	-
A_{15-16}	Анализирование информации и запись ее в журнал тестирования	1	A_{6-15}	-

Таблица 4– Перечень работ мониторинга веб-сайта

Анализ сетевого графика мониторинга сети

Основными параметрами сетевого графика являются:

1. Наиболее раннее возможное время наступления j-го события T_p (j), вычисляемое по формуле:

$$T_P(j) = \frac{\max}{i \subset j}(T_P(i) - t_{ij}),$$ где

- i и j обозначаются номера предшествующего и последующего событий соответственно;
- t_{ij} — продолжительность (i, j) -й работы.

Из обозначения $i \subset j$ следует, что событие i предшествует событию j.

Расчёты:

$T_p(0)=0$	$T_p(6)=16$	$T_p(12)=61$
$T_p(1)=1$	$T_p(7)=31$	$T_p(13)=91$
$T_p(2)=1$	$T_p(8)=76$	$T_p(14)=26$
$T_p(3)=1$	$T_p(9)=76$	$T_p(15)=28$
$T_p(4)=46$	$T_p(10)=106$	$T_p(16)=108$
$T_p(5)=3$	$T_p(11)=61$	

2. Самое позднее допустимое время наступления i-го события Тп (i), вычисляемое по формуле

$$T_П(j) = \frac{\min}{i \supset j}(T_П(i) - t_{ij})$$

где из обозначения $i \supset j$ следует, что событие j предшествует событию i

Расчёты:

$T_p(0)=0$	$T_p(6)=96$	$T_p(12)=76$
$T_p(1)=29$	$T_p(7)=46$	$T_p(13)=106$
$T_p(2)=104$	$T_p(8)=104$	$T_p(14)=106$
$T_p(3)=16$	$T_p(9)=76$	$T_p(15)=107$
$T_p(4)=74$	$T_p(10)=106$	$T_p(16)=108$
$T_p(5)=106$	$T_p(11)=76$	

3. Резерв времени данного события Ri вычисляемый по формуле

$$R_i = (T_П(i) - T_P(i))$$

Расчёты:

$R_0=0$	$R_6=80$	$R_{12}=15$
$R_1=28$	$R_7=15$	$R_{13}=15$
$R_2=103$	$R_8=28$	$R_{14}=80$
$R_3=15$	$R_9=0$	$R_{15}=79$
$R_4=28$	$R_{10}=0$	$R_{16}=0$
$R_5=103$	$R_{11}=15$	

4. Полный резерв времени работы rп (i,j), вычисляемый по формуле

$$r_{П}(\mathrm{i,j}) = (T_{П}(j) - T_{Р}(i) - t_{ij})$$

Расчёты:

$r_п(0,1)=28$	$r_п(4,9)=0$	$r_п(12,13)=15$
$r_п(0,2)=102$	$r_п(6,14)=80$	$r_п(5,16)=103$
$r_п(0,3)=14$	$r_п(6,15)=79$	$r_п(10,16)=0$
$r_п(1,4)=30$	$r_п(7,11)=15$	$r_п(13,16)=15$
$r_п(2,5)=103$	$r_п(7,12)=15$	$r_п(14,16)=80$
$r_п(2,6)=80$	$r_п(8,10)=28$	$r_п(15,16)=79$
$r_п(3,7)=15$	$r_п(9,10)=0$	
$r_п(4,8)=28$	$r_п(11,13)=15$	

2.2.3 Сетевая модель устранения проблемы Веб-сайта

Определяем работы, которые следует сделать для полного устранения выявленных проблем:

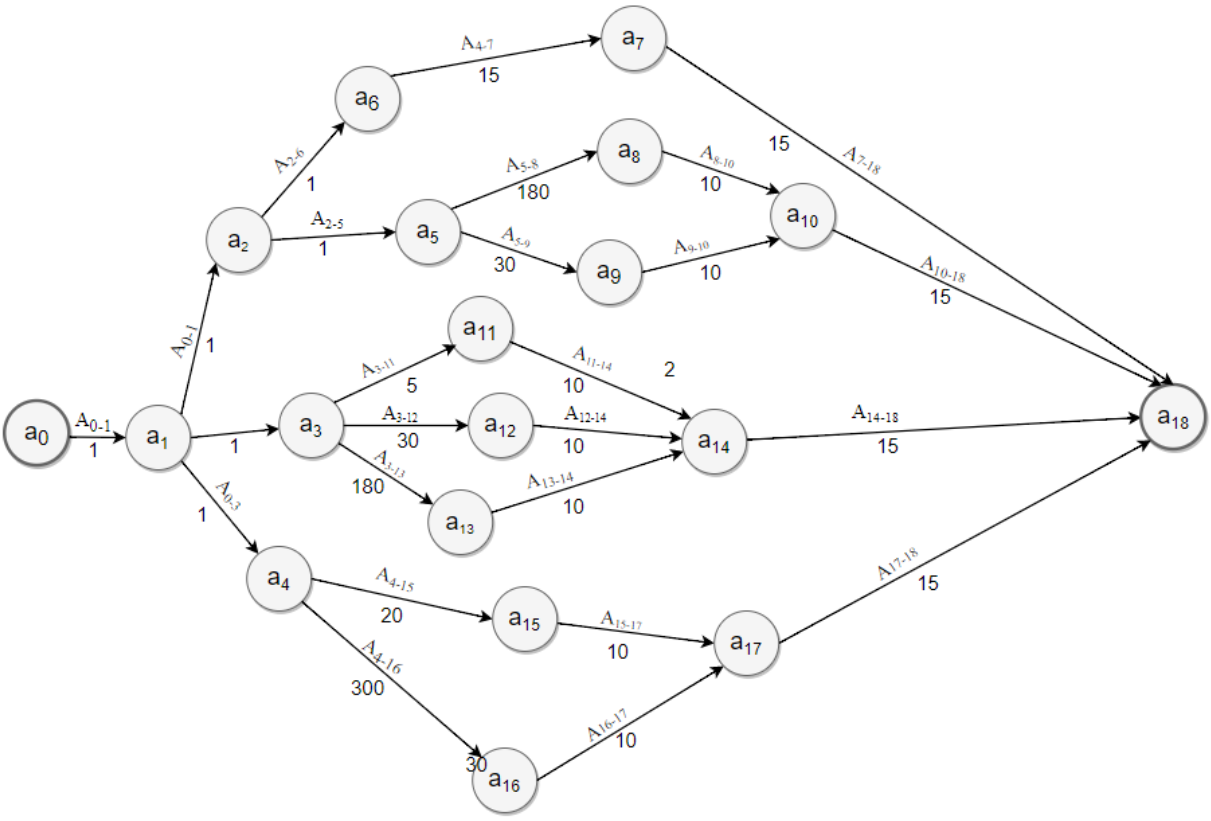

Рисунок 11 – Сетевой график устранения проблемы

Обозначение	Наименование событий
a_0	Обнаружена проблема
a_1	Обращение к журналу тестирования
a_2	Проблемы с физическим уровнем сети и сетевым борудованием
a_3	Проблемы с АО
a_4	Проблемы с прикладным ПО
a_5	Проблемы с сетевым кабелем
a_6	Проблемы с активным сетевым оборудованием
a_7	Оборудование перезагружено
a_8	Оборудование самостоятельно починено
a_9	Оборудование заменено на новое
a_{10}	Устранены неисправности в сети
a_{11}	Оборудование перезагружено
a_{12}	Оборудование заменено на новое
a_{13}	Оборудование самостоятельно починено
a_{14}	Устранение неисправностей в АО
a_{15}	Перезагрузка сервера
a_{16}	Переустановка ПО
a_{17}	Устранены неисправности ПО
a_{18}	Проблема устранена

Таблица 5 – Перечень событий устранения проблем

Обозначение работ	Наименование работ	Время ыполнения работы, нс	Предшествующие работы	Последующие работы
A_{0-1}	Просмотр журнала тестирования	1	-	A_{1-2}, A_{1-3}, A_{1-4}
A_{1-2}	Переход к проблемам с физическим ровнем сети и сетевым борудованием	1	A_{0-1}	A_{2-5}, A_{2-6}
A_{1-3}	Переход к проблемам АО	1	A_{0-1}	A_{3-11},
A_{1-4}	Переход к проблемам с рикладным ПО	1	A_{0-1}	A_{4-15},A_{4-16},
A_{2-5}	Переход к проблемам с сетевым абелем	1	A_{1-2}	A_{5-8}, A_{5-9},
A_{2-6}	Переход к проблемам с активным етевым оборудованием	1	A_{1-2}	A_{6-7}
A_{6-7}	Перезагрузка активного борудования	15	A_{2-6}	A_{7-18}
A_{5-8}	Починка сетевого кабеля амостоятельно	180	A_{2-5}	A_{8-10}
A_{5-9}	Замена неисправного кабеля на овое	30	A_{2-5}	A_{9-10}
A_{8-10}	Устранение неисправностей в сети	10	A_{1-3}	A_{10-18}
A_{9-10}	Устранение неисправностей в сети	10	A_{1-3}	A_{10-18}
A_{3-11}	Перезагрузка активного борудования	5	A_{1-3}	A_{11-14},
A_{3-12}	Замена сломанного оборудование а новое	30	A_{1-3}	A_{12-14}
A_{3-13}	Починка оборудования амостоятельно	180	A_{1-3}	A_{13-14}
A_{11-14}	Устранение неисправностей с АО	10	A_{3-11}	A_{14-18}
A_{12-14}	Устранение неисправностей с АО	10	A_{3-12}	A_{14-18}
A_{13-14}	Устранение неисправностей с АО	10	A_{3-13}	A14-18
A_{4-15}	Перезагрузка сервера	20	A_{1-4}	A_{15-17}
A_{4-16}	Переустановка ПО	300	A_{1-4}	A_{16-17}
A_{15-17}	Устранение неисправностей с ПО	10	A_{4-15}	A_{17-18}
A_{16-17}	Устранение неисправностей с ПО	10	A_{4-16}	A17-18
A_{7-18}	Составление доклада о устранении роблемы	15	A_{6-7}	-
A_{10-18}	Составление доклада о устранении роблемы	15	A_{11-14}	-
A_{14-18}	Составление доклада о устранении роблемы	15	A12-14	-

Таблица 6 – Перечень работ устранения проблем

Анализ сетевого графика устранения проблемы

Основными параметрами сетевого графика являются:

1. Наиболее раннее возможное время наступления j-го события T_p (j), вычисляемое по формуле:

$$T_P(j) = \max_{i \subset j} (T_P(i) - t_{ij}), \text{ где}$$

- i и j обозначаются номера предшествующего и последующего событий соответственно;

- t_{ij} — продолжительность (i, j) -й работы.

Из обозначения $i \subset j$ следует, что событие i предшествует событию j.

Расчёты:

$T_p(0)=0$	$T_p(7)=18$	$T_p(14)=192$
$T_p(1)=1$	$T_p(8)=183$	$T_p(15)=22$
$T_p(2)=2$	$T_p(9)=33$	$T_p(16)=302$
$T_p(3)=2$	$T_p(10)=193$	$T_p(17)=312$
$T_p(4)=2$	$T_p(11)=7$	$T_p(18)=327$
$T_p(5)=3$	$T_p(12)=32$	
$T_p(6)=3$	$T_p(13)=182$	

2. Самое позднее допустимое время наступления i-го события Тп (i), вычисляемое по формуле

$$T_П(j) = \min_{i \supset j} (T_П(i) - t_{ij})$$

где из обозначения $i \supset j$ следует, что событие j предшествует событию i

Расчёты:

$T_П(0)=0$	$T_П(7)=312$	$T_П(14)=312$
$T_П(1)=296$	$T_П(8)=302$	$T_П(15)=302$
$T_П(2)=296$	$T_П(9)=302$	$T_П(16)=302$
$T_П(3)=297$	$T_П(10)=312$	$T_П(17)=312$
$T_П(4)=282$	$T_П(11)=312$	$T_П(18)=327$
$T_П(5)=272$	$T_П(12)=312$	
$T_П(6)=297$	$T_П(13)=312$	

3. Резерв времени данного события Ri вычисляемый по формуле

$$R_i = (T_П(i) - T_P(i))$$

Расчёты:

$R_0=0$	$R_6=294$	$R_{12}=280$
$R_1=295$	$R_7=294$	$R_{13}=130$
$R_2=294$	$R_8=119$	$R_{14}=120$
$R_3=295$	$R_9=269$	$R_{15}=280$
$R_4=280$	$R_{10}=119$	$R_{16}=0$
$R_5=269$	$R_{11}=305$	$R_{17}=0$

4. Полный резерв времени работы гп (i,j), вычисляемый по формуле

$$r_{П}(\mathrm{i,j}) = (T_{П}(j) - T_{Р}(i) - t_{ij})$$

Расчёты:

$r_п(0,1)=295$
$r_п(1,2)=294$
$r_п(1,3)=295$
$r_п(1,4)=280$
$r_п(2,5)=268$
$r_п(2,6)=294$
$r_п(6,7)=294$
$r_п(5,8)=119$
$r_п(5,9)=269$
$r_п(8,10)=119$
$r_п(9,10)=269$
$r_п(3,11)=305$
$r_п(3,12)=280$
$r_п(3,13)=130$
$r_п(11,14)=295$
$r_п(12,14)=270$
$r_п(13,14)=110$
$r_п(4,15)=280$
$r_п(4,16)=0$
$r_п(15,17)=280$
$r_п(16,17)=0$
$r_п(7,18)=294$
$r_п(10,18)=119$
$r_п(14,18)=120$
$r_п(17,18)=0$

2.2.4 Сетевая модель функционирования Веб-сайта

На рисунке 12 представлен граф функционирования веб-сайта.

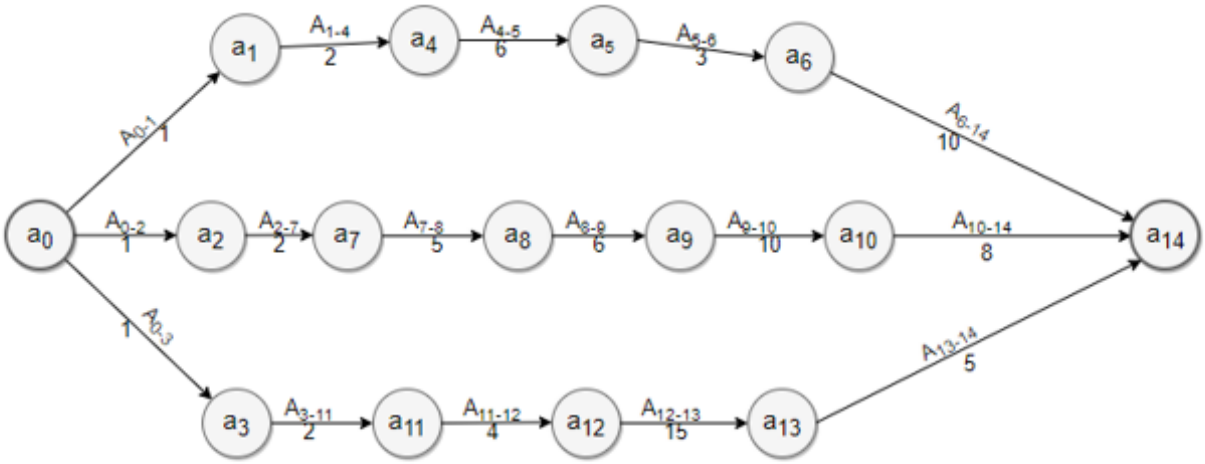

Рисунок 12– Сетевой график функционирования веб-сайта

Обозначение	Наименование событий
a_0	Начало функционирования веб-сайта
a_1	Запуск аппаратной составляющей веб-сайта
a_2	Запуск программной составляющей веб-сайта
a_3	Доступ к информационным ресурсам
a_4	Проверка кабеля электропитания
a_5	Обработка введенного адреса с клавиатуры системой
a_6	Проверка доступности вычислительных ресурсов
a_7	Проверка системы управления веб-сайта и серверов
a_8	Данные переведены в нужный формат и структурированы
a_9	Данные переданы на сервер
a_{10}	Пройдена проверка соответствия адресов
a_{11}	Обработка трафика системой
a_{12}	Сформирован запрос к базе данных
a_{13}	Проверка целостности контента
a_{14}	Пройдена проверка обращения к памяти системы
a_{15}	Успешное функционирование веб-сайта

Таблица 7 – Перечень событий функционирования сайта

Обозначение работ	Наименование работ	Время выполнения работы, нс	Предшествующие работы	Последующие работы
A_{0-1}	Начало функционирования аппаратного обеспечения	1	-	A_{1-2}
A_{1-4}	Подключение электропитания	2	A_{0-1}	A_{4-5}
A_{4-5}	Ввод с клавиатура адреса	6	A_{1-2}	A_{5-6}
A_{5-6}	Создание сессии с сервером	3	A_{4-5}	A_{6-15}
A_{6-15}	Успешное функционирование системы	10	A_{5-6}	-
A_{0-2}	Начало функционирования программного обеспечения	1	-	A_{2-7}
A_{2-7}	Формирование запроса	2	A_{0-2}	A_{7-8}
A_{7-8}	Структурирование данных и перевод данных в нужный формат	5	A_{2-7}	A_{8-9}
A_{8-9}	Определение кратчайшего пути передачи данных и формирования связи с сервером	6	A_{7-8}	A_{9-10}
A_{9-10}	Сравнение отправленных данных с установленными данными	10	A_{8-9}	A_{10-15}
A_{10-15}	Успешное функционирование системы	8	A_{9-10}	-
A_{0-3}	Начало работы с информационными ресурсами	1	-	A_{3-11},
A_{3-11}	Передача трафика	2	A_{0-3}	A_{11-12}
A_{11-12}	Формирование запроса к базе данных	4	A_{3-11}	A_{12-13}
A_{12-13}	Сканирование контента	10	A_{11-12}	A_{13-14}
A_{13-14}	Обращение к памяти системы	15	A_{12-13}	A_{14-15}
A_{14-15}	Успешное функционирование системы	5	A_{13-14}	-

Таблица 8 – Перечень работ функционирования сайта

1) Расчет наиболее ра

T_p (0)=0
T_p (1)=8
T_p (2)=9
T_p (3)=11
T_p (4)=14

2) Расчет самого позд

$T_п$ (14)=33
$T_п$ (13)=32
$T_п$ (12)=30
$T_п$ (11)=27
$T_п$ (10)=20

3) Резерв времени каж

R_0=0-0=0
R_1=8-8=0
R_2=9-9=0
R_3=11-11=0
R_4=14-14=0

4) Полный резерв вре

$r_п$ (0,1)=8-0-8=0
$r_п$ (1,2)=9-8-4=0
$r_п$ (2,3)=11-12-2=0
$r_п$ (3,8)= 14-14-1=0
$r_п$ (1,4)=18-8-3=0
$r_п$ (4,5)=15-11-7=0

5) Свободный резерв в

r_c (0,1)=8-0-8=0
r_c (1,2)=9-8-4=0
r_c(2,3)=11-12-2=0
r_c(3,8)=14-14-1
r_c(1,4)=11-8-3=0
r_c (4,5)=18-11-7=0

Основные показатели сетевого графика, по которым выполняется его анализ:

Критический путь системы функционирования аппаратного обеспечения:

$a_0+a_1+a_4+a_5+a_6$= 1+2+6+3+10=26(нс)

Критический путь системы функционирования программного обеспечения:

$a_0+a_2+a_8+a_7+a_9+a_{10}$=1+2+5+6+10+8=32(нс)

Критический путь системы функционирования информационных ресурсов:

$a_0+a_3+a_{11}+a_{12}+a_{13}$=1+2+4+15+5=27 (нс)

Оценивание характеристик систем функционирования:

Математическая модель управленческого решения:

$P = F(T_Э, \Delta t_{пп}, \Delta t_{ип}, \Delta t_{нп})$.

Производим расчеты для системы, согласно результатам, полученным при сетевом планировании:

Δtпп= 1день; λ = 0,0007;

Δtип= 27 нс = 0,018сут; ν2= 1/0,018=55,5

Δtнп=32 нс = 0,022сут; ν2= 1/0,022=45,4

ξ^+=0,06

ξ^-=0,03

Используя эти данные и произведя расчеты по формуле

$P_2 = \frac{\lambda*v_1*v_2+v_1*v_2*\xi^+}{\lambda*v_1*v_2+\lambda*v_1*\xi^-+\lambda*v_2*\xi^-+v_1*v_2*\xi^++v_1*v_2*\xi^-}$, получим вероятность, равную 0.864.

Требуется рассчитать среднее время проявления проблемы в системе (λ) для заданных вероятностей: 0.8,0.85, 0.9.

При разных вероятностях $\mathbf{P_{инп}}$получаем следующие результаты:

P = 0,8: λ= 2,11; $\Delta t_{пп}$=0,35 сут.;

P = 0,9: λ= 1,45; $\Delta t_{пп}$= 0,5 сут.;

P = 0,95: λ= 1,39; $\Delta t_{пп}$= 0,54 сут.

Далее требуется рассчитать среднее время обеспечение уровня безопасности в системе (λ) для заданных вероятностей: 0.8, 0.9, 0,95. Для того чтобы рассчитать среднее время обеспечение уровня безопасности в системе нужно уменьшать критический путь для вероятности 0,8:

Критический путь	Исходное значение	Измененное значение
v_1	55,5	73,23
v_2	45,4	56,63

Таблица 9 – Критический путь вероятности 0,8

При P=0,8, λ=6,43⇨ раз в 0,33 ($\Delta t_{пп}$) суток

Для того чтобы рассчитать среднее время обеспечение уровня безопасности в системе нужно увеличить критический путь для вероятности 0,95:

Критический путь	Исходное значение	Измененное значение
v_1	55,5	42,2
v_2	45,4	37,73

Таблица 10 – Критический путь вероятности 0,95

При P=0,95, λ= 1,25⇨раз в 0,75 ($\Delta t_{нп}$) суток.

После анализа результатов получаем P – показатель эффективности, который характеризует вероятность идентификации ситуации и принятия адекватного решения для процесса обеспечения безопасной работы модели БДЗ:

0.864= F (1; 0,018; 0,022)

Для нормальной работоспособности системы значение вероятности должно быть не меньше 0,8. В ходе работы было вычислено, что вероятность появления проблемы в БДЗ равна 0,864, что говорит о хорошем функционировании системы.

2.3 Оценивание показателя эффективности реализации управленческих решений Веб-сайта

Условие, при котором работа объекта всегда будет достоверным событием ($P_2 = 1$), то есть показатель эффективности должен быть равным 1. Эффективность – свойство, которое характеризует степень достижения цели (или свойство, которое характеризует степень реализации возможностей, заложенной в системе разработчиком). Показатель эффективности позволяет понимать предназначение той системы, которую разрабатываем.

Итак, Т – среднее время обращения пользователя к биометрическому замку. В процессе обращения возникает определенное количество угроз. Показатель пригодности биометрического замка – это среднее количество срывов решения задач. Показатель пригодности системы мониторинга определяется количеством срывов.

Системное решение:

$$P_1 = \frac{v_1 * v_2 * \xi^-}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

$$P_2 = \frac{\lambda * v_1 * v_2 + v_1 * v_2 * \xi^+}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

$$P_3 = \frac{\lambda * v_2 * \xi^-}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

$$P_4 = \frac{\lambda * v_1 * \xi^-}{\lambda * v_1 * v_2 + \lambda * v_1 * \xi^- + \lambda * v_2 * \xi^- + v_1 * v_2 * \xi^+ + v_1 * v_2 * \xi^-}$$

События не достоверны, так как в системе есть возмущения (случайные факторы). В работе под возмущением имеется ввиду срывов (показатель пригодности). Если показатель срывов устремить к нулю, то события будут достоверны. То есть $P_1 = 0$, $P_3 = 0$, $P_4 = 0$ при $\xi^- \to 0$.

Тогда для вероятности выявления и нейтрализации проблемы: $P_2 = \frac{\lambda * v_1 * v_2}{\lambda * v_1 * v_2} = 1$.

Показателем эффективности модели управления процессом прохождения сайта, в зависимости от различной обстановки, будет служить аналитическая зависимость:

$$P_2 = f\left(\lambda, v_1, v_2, v_3, \zeta^+, \zeta^-\right).$$

где λ - есть величина $\left(\lambda = \frac{1}{\Delta t_{ПП}}\right)$, где $\Delta t_{ПП}$ - среднее время проявление проблемы;

v_1 - есть величина $\left(v_1 = \frac{1}{\Delta t_{ИП}}\right)$, где $\Delta t_{ИП}$ - среднее время идентификации проблемы;

v_2 - есть величина $\left(v_2 = \frac{1}{\Delta t_{НП}}\right)$, где $\Delta t_{НП}$ - среднее время нейтрализации проблемы;

v_3 - частота срыва нейтрализации проблемы ЛПР, по причине невозможности распознать ситуацию (показатель квалификации ЛПР);

ζ^+ - есть величина $\left(\zeta^+ = \frac{1}{T_Э}\right)$, где $T_Э$ - длительность решения задачи;

ζ^- - частота срыва плана;

P_2 - показатель эффективности реализации управленческих решений при прохождении судном судоходного шлюза.

По условию задачи имеем:

$T_Э = 3ч.$ - время решения задачи;

$N_{СРЫВ} = 1шт.$ - количество срывов за 1 сутки;

$N_П = 50шт.$ - количество проходов пользователей за 1 сутки;

$$\Delta t_{ПП} = 1 \Rightarrow \lambda = \left(\frac{1}{\Delta t_{ПП}}\right) = \left(\frac{1}{1}\right) = 1$$

$$\Delta t_{ИП} = 0{,}014 \Rightarrow \nu_1 = \left(\frac{1}{\Delta t_{ИП}}\right) = \left(\frac{1}{0{,}014}\right) = 71{,}43$$

$$\Delta t_{НП} = 0{,}02 \Rightarrow \nu_2 = \left(\frac{1}{\Delta t_{НП}}\right) = \left(\frac{1}{0{,}02}\right) = 50.$$

$$\zeta^- = \frac{N_{СРЫВ}}{N_П} = \frac{1}{50} = 0.02.$$

$$T_Э = 3 \Rightarrow \zeta^+ = \left(\frac{1}{T_Э}\right) = \left(\frac{1}{3}\right) = 0.33$$

Задаем условие:

$$\frac{\Delta t_{ИП} + \Delta t_{НП}}{\Delta t_{ПП}} < 1.$$

$$\frac{0{,}014 + 0{,}02}{1} < 1$$

Найдем вероятность выполнения задачи P_2 с учетом:

– частоты (ν_3) - которая характеризует частоту срыва нейтрализации проблемы ЛПР, по причине невозможности распознать ситуацию, что является основой показателя квалификации ЛПР;

– частоты (ζ^-) - которая характеризует среднее количество срыва выполнения плана прохождения пользователей, что показывает успешность выполнения функционирования сайта.

1. Если $\nu_3 = \frac{\nu_1}{1000}$, $\zeta^- = \frac{N_{СРЫВ}}{N_П} = \frac{1}{50} = 0.02.$ то $P2 = 0.8195.$

$$P_1 = \frac{\nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0743.$$

$$P_2 = \frac{\lambda \cdot \nu_1 \cdot \nu_2 + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \zeta^+ \cdot \nu_3}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.8195.$$

$$P_3 = \frac{\lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0531.$$

$$P_4 = \frac{\lambda \cdot \nu_1 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0531.$$

Проверяем условие $P_1 + P_2 + P_3 + P_4 = 0.0743 + 0.8195 + 0.0531 + 0.0531 = 1$.

1. Если $\nu_3 = \frac{\nu_1}{100}$, $\zeta^- = \frac{N_{СРЫВ}}{N_П} = \frac{1}{50} = 0.02$. то $P2 = 0.8387$.

$$P_1 = \frac{\nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0761.$$

$$P_2 = \frac{\lambda \cdot \nu_1 \cdot \nu_2 + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \zeta^+ \cdot \nu_3}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.8387.$$

$$P_3 = \frac{\lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0429$$

$$P_4 = \frac{\lambda \cdot \nu_1 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0423.$$

Проверяем условие $P_1 + P_2 + P_3 + P_4 = 0.0761 + 0.8387 + 0.0429 + 0.0423 = 1$.

2. Если $\nu_3 = \frac{\nu_1}{1000}$, $\zeta^- = \frac{N_{СРЫВ}}{N_П} = \frac{2}{50} = 0.04$. то $P2 = 0.7938$.

$$P_1 = \frac{\nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0409.$$

$$P_2 = \frac{\lambda \cdot \nu_1 \cdot \nu_2 + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \zeta^+ \cdot \nu_3}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.7938.$$

$$P_3 = \frac{\lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0827.$$

$$P_4 = \frac{\lambda \cdot \nu_1 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0826.$$

Проверяем условие $P_1 + P_2 + P_3 + P_4 = 0.0409 + 0.7938 + 0.0827 + 0.0826 = 1$.

3. Если $\nu_3 = \frac{\nu_1}{100}$, $\zeta^- = \frac{N_{СРЫВ}}{N_П} = \frac{2}{50} = 0.04$. то $P2 = 0.8001$.

$$P_1 = \frac{\nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0361.$$

$$P_2 = \frac{\lambda \cdot \nu_1 \cdot \nu_2 + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \zeta^+ \cdot \nu_3}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.8001.$$

$$P_3 = \frac{\lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0824.$$

$$P_4 = \frac{\lambda \cdot \nu_1 \cdot \zeta^-}{\lambda \cdot \nu_1 \cdot \nu_2 + \lambda \cdot \nu_1 \cdot \zeta^- + \lambda \cdot \nu_3 \cdot \zeta^- + \nu_1 \cdot \nu_2 \cdot \zeta^+ + \nu_1 \cdot \nu_2 \cdot \zeta^- + \nu_1 \cdot \zeta^+ \cdot \nu_3 + \nu_1 \cdot \nu_2 \cdot \zeta^-} = 0.0814.$$

Проверяем условие $P_1 + P_2 + P_3 + P_4 = 0.0361 + 0.8001 + 0.0824 + 0.0814 = 1$.

Разработка модели обеспечения безопасности на основе решения задачи при заданной вероятности реализации управленческих решений при прохождении пользователями сайта $P_2 = 0.8$, при общем времени решения задачи $\zeta^+ = 0.057.$, при количестве срывов $\zeta^- = 0.02.$, частоте срыва нейтрализации ЛПР $\nu_3 = 0.001$, и заданной угрозе $\lambda = 0.125$:

$P_2 = 0.8$

$$T_Э = 3 \Rightarrow \zeta^+ = \left(\frac{1}{T_Э}\right) = \left(\frac{1}{3}\right) = 0.33$$

$$\zeta^- = \frac{N_{СРЫВ}}{N_П} = \frac{1}{50} = 0.02.$$

$$\Delta t_{ПП} = 1 \Rightarrow \lambda = \left(\frac{1}{\Delta t_{ПП}}\right) = \left(\frac{1}{1}\right) = 1$$

$\nu_3 = 0.071$

$$\frac{1 \cdot \nu_1 \cdot \nu_2 + \nu_1 \cdot \nu_2 \cdot 0.33 + \nu_1 \cdot 0.33 \cdot 0.071}{1 \cdot \nu_1 \cdot \nu_2 + 1 \cdot \nu_1 \cdot 0.02 + 1 \cdot 0.071 \cdot 0.02 + \nu_1 \cdot \nu_2 \cdot 0.33 + \nu_1 \cdot \nu_2 \cdot 0.02 + \nu_1 \cdot 0.33 \cdot 0.071 + \nu_1 \cdot \nu_2 \cdot 0.02} = 0.8$$

$$\nu_1 = 2{,}5 \Rightarrow \Delta t_{ИПП} = \left(\frac{1}{\nu_1}\right) = \left(\frac{1}{2{,}5}\right) = 0.4.$$

$$\nu_2 = 0.8 \Rightarrow \Delta t_{НП} = \left(\frac{1}{\nu_2}\right) = \left(\frac{1}{0.8}\right) = 1{,}25$$

$P_2 = f(\lambda, \nu_1, \nu_2, \nu_3, \zeta^+, \zeta^-)$; $P_2 = f(1; 2.5; 0.8; 0.071; 0.33; 0.02)$

Чтобы обеспечить вероятность $P_2 = 0.8$, расчеты для идентификации и нейтрализации на каждом переходе события к событию должны быть при расчете графиков идентификации все данные надо уменьшить в 2 раза.

Обозначение работ	Время выполнения работы,нс	Измененное время выполнения работы, нс
A_{0-1}	1	0.5
A_{0-2}	1	0.5
A_{1-3}	3	1.5
A_{3-5}	2	1
A_{3-4}	2	1
A_{4-7}	2	1
A_{4-6}	6	3
A_{6-12}	5	2.5
A_{7-12}	4	2
A_{5-8}	3	1.5
A_{8-12}	3	1.5
A_{2-9}	8	4
A_{9-11}	5	2.5
A_{11-12}	4	2
A_{2-10}	3	1.5
A_{10-11}	10	5

Таблица 11 – Изменение времени в перечне работ мониторинга

1. При расчете графиков нейтрализации все данные надо уменьшить в 3 раза.

Обозначение работ	Время выполнения работы, нс	Измененное время выполнения работы, нс
A_{0-1}	1	0.3
A_{1-2}	2	0.6
A_{2-4}	2	06
A_{4-12}	6	2
A_{12-13}	15	5
A_{2-6}	2	0.6
A_{2-7}	2	0.6
A_{6-11}	5	1.6
A_{7-11}	15	5
A_{11-13}	1	0.3
A_{1-3}	3	1
A_{3-5}	5	1.6
A_{5-9}	6	2
A_{3-8}	8	2.6
A_{8-9}	4	1.3
A_{9-10}	10	3.3
A_{10-13}	1	0.3

Таблица 12 – Изменение времени в перечне работ устранения угрозы.

Вывод по второму разделу:

Для обеспечения информационной безопасности сайта в условиях деструктивных воздействий внешней среды был использован естественно-научный подход. Для реализации его возможностей необходимо разработать аналитическую динамическую модель процесса обеспечения информационной безопасности сайта. В результате аналитическая, динамическая модель позволила получить условие существования процесса обеспечения информационной безопасности сайта.

Для обеспечения необходимого уровня информационной безопасности сайта необходимо провести системную интеграцию.

Процессы:
- проявления проблемы;
- осознание проблемы;
- предотвращение проблем;
- Индикатор IB сайта.

Это позволяет анализировать уровень безопасности, в основе которого лежит своевременное выявление и устранение проблемы.

Необходимо обеспечить требуемый уровень информационной безопасности следующим образом:

- сократить время на нейтрализацию проблемы или ускорить процесс мониторинга, есть возможность улучшить эти два показателя равномерно;
- внедрить современные технические средства, такие как специализированное программное обеспечение, которое можно получить в открытом доступе;
- проводить разъяснительные беседы с рабочим персоналом по правилам работы с сайтом, а также мерам защиты информации.

Показано, что условие существования процесса сопровождения формируется из временных характеристик деятельности по обеспечению информационной безопасности сайта.

Установлено, что эти временные характеристики зависят от состояний 4-х процессов. Для этого решена задача увязки временных характеристик процессов с состояниями этих процессов, работами, которые необходимо выполнить для достижения этих состояний и ресурсами времени при выполнении этих работ. Для этого использовался математический аппарат сетевых моделей, который при расчете критического пути обеспечил количественное формирование элементов условия существования процесса обеспечения информационной безопасности сайта. И в результате обеспечил необходимый уровень информационной безопасности сайта.

Разработанная аналитическая математическая модель обеспечения информационной безопасности сайта позволила перейти к проектированию и разработке технологии обеспечения информационной безопасности сайта.

3. Проектирование и разработка технологии обеспечения Информационной безопасности функционирования Веб-сайта

3.1. Проектирование Веб-сайта

В результате разработанной аналитической, математической модели в условиях глобальной сети Интернет и полученных расчетов следующим этапом ВКР является проектирование условно защищенного веб-сайта.

Чтобы что-то спроектировать, нужно понимать, что мы проектируем. Поэтому очень важно перед проектированием сайта проводить тесты производительности и безопасности, так называемое многопользовательское тестирование, так как каждый день большое количество пользователей ищут нужные им рецепты, заходят на сайт, сортируют товары, создавать аккаунты и т.д., что постоянно заставляет сервер работать. Работа сервера в первую очередь зависит от нагрузки. Часто серверы не выдерживают большой нагрузки и, соответственно, на этапе разработки важно провести тестирование, которое проверяет, как сайт будет вести себя в разные интервалы нагрузки. Это часто вызывает проблемы:

- Медленная загрузка страниц.
- Снижение активности
- Случайные ошибки
- Прерывание соединения с веб-сервером [13]

Эффективная платформа для оценки систем обнаружения вторжений должна включать в себя инструменты, которые могут генерировать атакующий трафик, полезный трафик, инструменты, которые могут собирать предупреждения, генерируемые системами обнаружения вторжений, и инструменты, которые могут создавать отчеты, включающие ряд измерений (включая индикаторы ложных срабатываний). и ложноотрицательные).

Поэтому нужна хорошая система мониторинга. Одним из самых популярных сервисов является Zabbix, система мониторинга, позволяющая быстро узнавать о нехватке ресурсов [14].

При проектировании каждого сайта выделяют три основных направления:

1. Проектирование фронтальной части
2. Проектирование серверной части
3. Дизайн базы данных

Front end и back end разработка — две жизненно важные области кодирования. Фронтенд-разработка описывает программирование, которое выполняется на сайте для воздействия на клиентскую или пользовательскую сторону веб-сайта или приложения. Бэкенд-разработка — это программирование, выполняемое на стороне сервера или на стороне, которую видит только разработчик.

3.1.1. Проектирование Front-end

Когда вы слышите, как разработчики говорят о внешнем интерфейсе веб-сайта или приложения, они говорят о той части веб-сайта, которую вы можете видеть и с которой можете взаимодействовать как пользователь. Внешний интерфейс включает в себя все: от дизайна, структуры и макета веб-сайта до контента. Когда новый посетитель переходит на веб-сайт или в приложение, внешний интерфейс создает первое впечатление о компании или бренде, поэтому крайне важен удобный для пользователя (UX) дизайн внешнего интерфейса. Подробнее о фронтенд-разработке можно прочитать здесь.

Три основных языка внешнего интерфейса, которые разработчики программного обеспечения используют для создания бесшовных, удобных для пользователя веб-сайтов:

- HTML означает язык гипертекстовой разметки, и все это касается контента. Он определяет структуру веб-страниц, которые мы просматриваем в нашем браузере каждый раз, когда заходим на Facebook, Twitter, Amazon или любой другой сайт. Самое главное, это позволяет нам делать внешние ссылки на другие веб-страницы.
- Каскадные таблицы стилей (CSS) — это язык стиля. Это позволяет разработчикам обеспечивать визуальное воздействие на веб-сайты без необходимости написания сложного HTML. CSS позволяет изменять элементы, созданные в HTML, и дает представление о работе с различными типами шрифтов и другими функциями стиля. Он добавляет элемент дизайна к веб-страницам и приложениям.
- JavaScript — еще один важный язык программирования для фронтенд-разработки. Он позволяет улучшить ваш HTML с помощью анимации, интерактивности и динамических визуальных эффектов. Он включает в себя возможность мгновенного предоставления обратной связи пользователям, что делает веб-страницы более информативными. Например, корзина покупок или корзина на веб-сайте, в которой суммируется общая стоимость, — это интерактивная функция JavaScript.

Лучший способ поддерживать сессию в PHP — это использовать встроенный механизм обработки сессий. Сеансы в PHP позволяют сохранять данные сеанса на веб-сервере при последующих запросах. Типичный HTML-сайт не может передавать данные с одной страницы на другую. Любая информация, используемая на странице, забывается на новой странице. В PHP сеанс поддерживается для каждого пользователя путем создания уникального идентификатора сеанса (SID) и сохранения различных переменных сеанса на основе этого SID. PHP использует файлы cookie или кодировку URL в качестве механизма отслеживания сеанса. Уникальный SID для каждого пользователя помогает различать данные двух разных пользователей. Хранение данных на разных страницах в PHP полезно тем, что все взаимодействия скрыты от пользователя.

Более того, эти данные сеанса являются временными и в большинстве случаев уничтожаются после того, как пользователь покинет веб-сайт и сеанс. Переменные сеанса — это информация о состоянии, связанная с пользователем и хранящаяся на веб-сервере. Информацию о состоянии пользователя на сервере можно найти с помощью идентификатора сеанса. Например, простая информация о сеансе в PHP может рассматриваться как установка значения переменной на веб-странице, а затем вызов переменной на следующей странице. PHP использует глобальный массив $_SESSION для хранения всех переменных сеанса. Простая переменная сеанса может иметь вид $_SESSION[’var’]. Ассоциативный массив $_SESSION может хранить любую информацию, например. массивы, объекты, строки и значения.

Сначала страница входа используется для запроса учетных данных пользователя. Пользователь передает учетные данные на сервер с помощью HTML-формы, используя метод POST, и эта информация собирается сценарием установки. После аутентификации сценарий установки устанавливает сеанс пользователя с помощью модуля управления сеансом. После запуска сеанса сервер создает страницу приветствия и перенаправляет пользователя на эту страницу. Страница приветствия и другие страницы приложения могут извлекать переменные сеанса из глобального массива $_SESSION.

Сеанс поддерживается между пользователем и сервером до тех пор, пока пользователь запрашивает любую из страниц приложения. Когда пользователь решает выйти из системы, сервер вызывает сценарий выхода, чтобы уничтожить сеанс пользователя, и перенаправляет пользователя на страницу квитанции, которая не использует сеанс. В следующих разделах мы приведем примеры различных функций управления сеансами PHP.

Преимущества выбора фреймворкReact:

Простота. Преимущество заключается в его гибкости и производительности, в частности благодаря работе над виртуальным DOM * и обновлению рендеринга в браузере только при необходимости. Поскольку браузеры довольно медленно реагируют на изменения в DOM, React имеет преимущество в ограничении взаимодействия с ним. Поэтому он выполняет операции с виртуальным DOM и сравнивает его с реальным DOM, чтобы внести необходимые изменения.

* Виртуальный DOM - это дерево объектов JS, которое быстро идентифицирует узлы, подлежащие обновлению. Поэтому это сокращает количество диалогов с API браузеров для создания DOM и повышения производительности.

Выгода. В отличие от других Фреймворков, ReactJS не привередлив, то есть он также не навязывает определенную библиотеку для данных, это всего лишь часть «View» интерфейса. ReactJS использует свой собственный синтаксис, называемый JSX чтобы смешивать JavaScript и HTML для получения удобочитаемого и обслуживаемого кода

Стабильность. Характерной чертой технологий, тем более цифровых, является то, что они постоянно совершенствуют и разрабатывают новые функционалы. Таким образом, недостаток инфраструктуры внешнего интерфейса заключается в том, что всегда нужно следить за ее развитием.

При проектировании интерфейса было принято решение разбить приложение на две динамически генерируемые страницы.

3.1.2. Проектирование Back-end

Между фронтенд- и бэкенд-разработкой нет реальной конкуренции. Оба типа разработчиков пользуются большим спросом, когда речь идет о возможностях трудоустройства. Тем не менее, быть вооруженным обоими наборами навыков или быть тем, что называется разработчиком полного стека, действительно идеально. Это означает, что вам должны быть открыты все варианты в сочетании и даже больше.

В одиночку React.js будет недостаточно для создания полнофункционального динамического сайта. Однако одна из его характеристик заключается в том, что созданные представления также могут отображаться на сервере через Node.js *. В отличие от ReactNative, нет необходимости использовать другую библиотеку. Один и тот же файл используется на стороне клиента и сервера. Это значительное улучшение производительности. Затем можно сначала создать статическую версию страниц с использованием сервера, которая является более быстрой и удобной для SEO, а затем активировать взаимодействия с пользователем и обновления пользовательского интерфейса с помощью React.js. клиент.

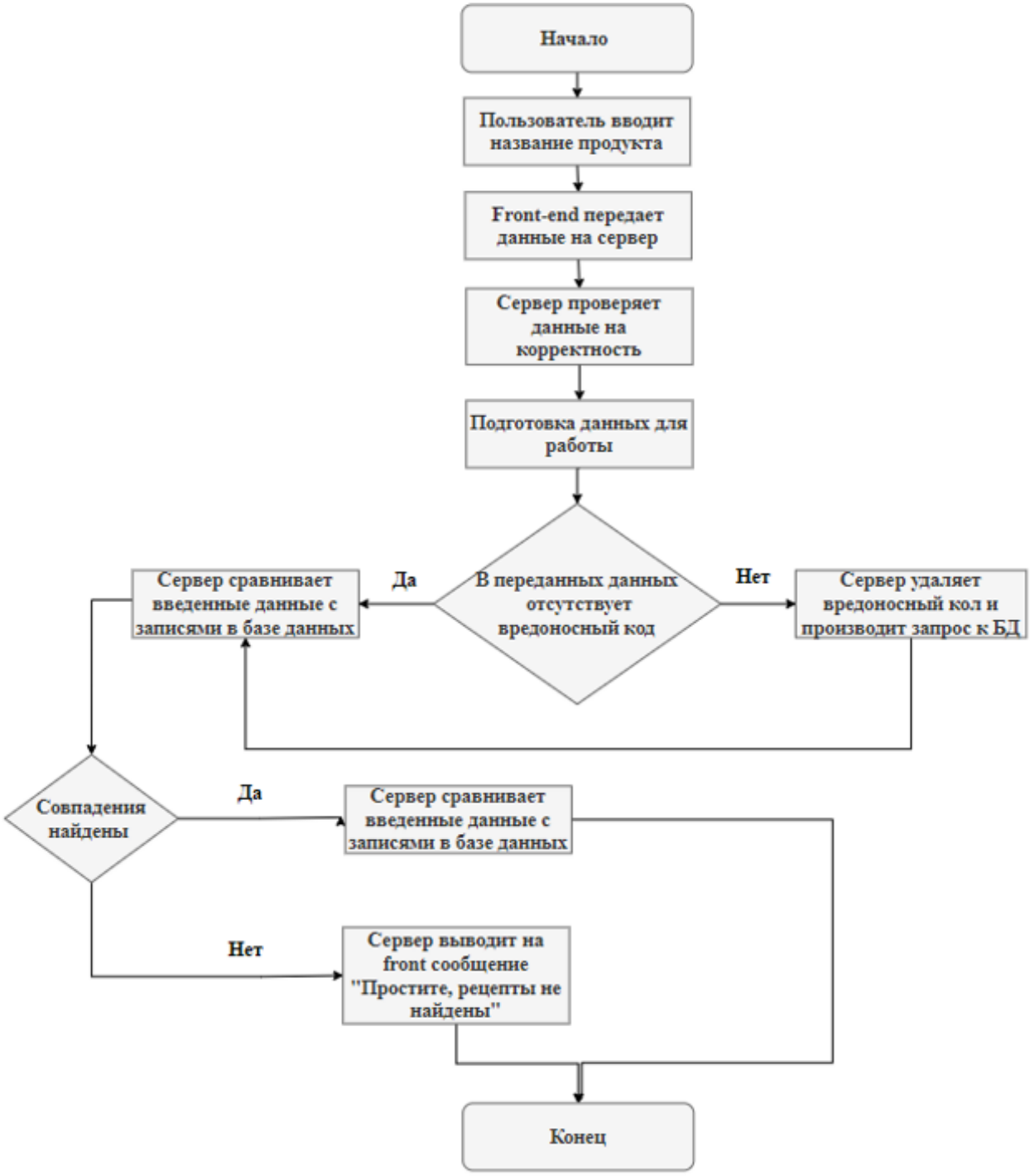

Рисунок 15 – блок-схема прототипа сервера

После того, как спроектирована back-end часть, можно приступить к проектированию базы данных.

3.1.3 Проектирование базы-данных

База данных собирает информацию и организует ее так, чтобы ее можно было легко найти, управлять и обновлять. Благодаря базе данных можно создавать, обновлять или удалять данные. База данных обычно используется для проведения исследований с использованием собранных данных. Таким образом, пользователь может использовать его, вводя запросы. Эти базы данных хранятся на серверах и могут быть перенесены в любой момент времени. На рисунке 16 представлен принцип работы базы данных.

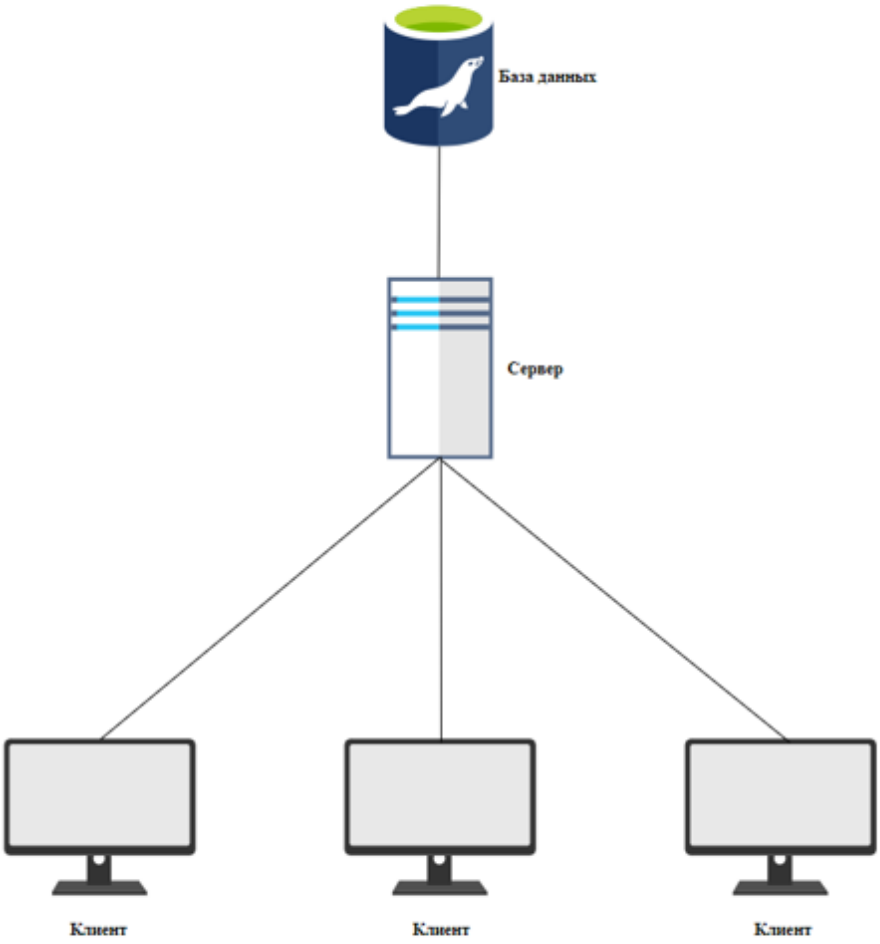

Рисунок 16 – Принцип работы базы данных

База данных может быть локальной, то есть используемой пользователем на машине, или распределенной, то есть информация хранится на удаленных машинах и доступна по сети. Основным преимуществом использования баз данных является возможность одновременного доступа к нескольким пользователям.

Чтобы иметь возможность контролировать данные, а также пользователей, нужна система управления. Управление базой данных осуществляется через систему, называемую СУБД (система управления базами данных).

СУБД может быть разбита на три подсистемы, представленные на рисунке 17:

- Система управления файлами: она позволяет хранить информацию на физическом носителе;
- внутренняя СУБД: она управляет планированием информации;
- внешняя СУБД: представляет собой интерфейс с пользователем.

Рисунок 17 – Принцип функционирования СУБД

В настоящее время большинство СУБД работают в режиме клиент-сервер. Сервер получает запросы от нескольких клиентов одновременно. Сервер анализирует запрос, обрабатывает его и возвращает результат клиенту.

Для проектирования базы данных нужно выбрать надежную СУБД. Для этого поставлены основные задачипри ее выборе[15]:

Для достижения некоторых из этих целей (особенно первых двух) стандартом ANSI / SPARC были определены три уровня описания данных [16].

Внешний уровень. Соответствует восприятию всей или части базы данных данной группой пользователей, независимо от других. Это описание называется внешней диаграммой или представлением. Может быть несколько внешних схем, представляющих различные виды базы данных с возможностью восстановления. Внешний уровень обеспечивает анализ и интерпретацию

запросов в примитивы более низкого уровня, а также отвечает за возможное преобразование необработанных данных, полученных в результате ответа на запрос, в формат, который желает пользователь.

Наиболее важными аспектами механизма безопасного управления сеансом являются возможность привязать входящий запрос к сеансу, которому он принадлежит, определить, где и как может храниться состояние сеанса, а также определить меры по защите механизмов от атак. В данной диссертации основное внимание уделяется анализу этих ключевых аспектов безопасного управления сеансами в среде с одним сервером. Прежде всего, мы изучили основы управления сеансом HTTP и обрисовали в общих чертах различные существующие варианты сохранения состояния приложения и обработки данных сеанса с помощью сеанса HTTP. Затем мы рассмотрели возможные сеансовые атаки и обсудили контрмеры, которые можно принять для предотвращения уязвимостей сеанса. Кроме того, в этой диссертации показаны общие этапы реализации сеанса с PHP и обсуждаются последствия манипулирования некоторыми параметрами конфигурации управления сеансом на уровне безопасности приложения. В конце концов, мы изучили доступные лучшие практики для поддержания безопасного сеанса. Поведение HTTP без сохранения состояния требует от разработчиков веб-приложений использования дополнительных методов для сохранения информации о состоянии и сеансе, специфичной для пользователя. Механизмы с отслеживанием состояния и без сохранения состояния могут использоваться с HTTP для управления сеансами. Чаще всего веб-приложения используют сеансы для наложения ограничений безопасности и инкапсуляции другой информации о состоянии. Таким образом, безопасность модуля управления сеансами веб-приложений является критической проблемой. Получив доступ к сеансу законного пользователя, злоумышленник может получить права доступа пользователя и выполнить любые допустимые операции. По этой причине тенденция запуска атак на процесс управления сессиями растет день ото дня. Часто процессы обработки сеансов веб-приложений слабо реализованы. Уязвимости в процессе управления сеансом могут нанести серьезный ущерб, поскольку они обычно хранят важные и конфиденциальные данные веб-систем. В существующих механизмах требуется дополнительная осторожность и профилактические меры для обеспечения достаточного уровня безопасности сеанса.

- **Концептуальный уровень.** Описывает структуру всех данных в базе данных, их свойства (то есть отношения, которые существуют между ними: их внутреннюю семантику), не беспокоясь о физической реализации или о том, как каждая рабочая группа захочет их использовать. В случае реляционных СУБД - это табличное представление, в котором семантика информации выражается с использованием концепций отношений, атрибутов и ограничений целостности. Это описание называется концептуальной диаграммой.

- **Внутренний или физический уровень.** Использует систему управления файлами для определения политики хранения и размещения данных. Таким образом, физический уровень отвечает за выбор физической организации файлов, а также за использование того или иного метода доступа в соответствии с запросом. Это описание называется внутренней схемой.

На основе вышеперечисленных требований к СУБД, в данной ВКР для проектирования базы данных будет использоваться система управления MariaDBс помощью языка SQL.

MariaDB — ответвление от системы управления базами данных MySQL, разрабатываемое сообществом под лицензией GNU GPL. Разработку и поддержку MariaDB осуществляет компания MariaDBCorporationAb и фонд MariaDBFoundation[17].

MariaDB был разработан, чтобы быть максимально совместимым с MySQL при замене нескольких ключевых компонентов. Он использует механизм хранения Aria, который работает как транзакционный, так и нетранзакционный механизм.

SQL - это язык структурированных запросов, который является компьютерным языком, используемым для хранения, обработки и извлечения данных, хранящихся в реляционной базе данных. Он состоит из четырех подмножеств:

Функциональная схема для обработки запросов представлена на рисунке 18:

Рисунок 18 – схема для обработки запросовSQL

Несмотря на то, что существует множество различных способов реализации механизма управления сеансом, каждый из методов реализации отслеживания сеансов имеет свои преимущества и ограничения. В данной диссертации анализируются преимущества и недостатки каждого метода с точки зрения безопасности. Кроме того, безопасная обработка сеансов может быть деликатной из-за множества уязвимостей сеансов и малоизвестных недостатков. Недостаток знаний о существующих возможностях сеансовых атак может легко сделать веб-приложение уязвимым для нарушения безопасности. По этой причине мы дали подробное описание возможных эксплуатаций сеансов и некоторых методов, которые можно использовать для предотвращения вторжений и минимизации последствий атак. Кроме того, использование инфраструктуры обработки сеансов является важной проблемой проектирования сложных веб-приложений. Разработчики PHP имеют доступ к мощному и надежному модулю управления сессиями благодаря правильному использованию встроенного в PHP механизма обработки сессий.

Мы рассмотрели функции управления сеансами PHP, которые могут помочь создать надежное и безопасное веб-приложение и обеспечить лучший опыт для пользователей приложения. Наконец, разработка защищенного управления сеансом не является сложной задачей, если следовать некоторым простым передовым методам во всех аспектах обслуживания состояния. В этой диссертации мы представили доступные рекомендации по передовому опыту, которые веб-приложения могут реализовать, чтобы обеспечить безопасность обработки сеансов и защиту клиентских данных. Разработчик веб-приложений в среде с одним сервером должен обратить внимание на создание и хранение идентификаторов сеансов, передачу информации между сервером и клиентом и удаление сеанса, когда он больше не нужен из-за выхода пользователя из

системы или слишком длительного простоя. . Не существует идеального решения для безопасного управления сеансами. Уровень безопасности, требуемый веб-приложением, зависит от его предполагаемого использования. Более того, каждое решение для отслеживания сеансов имеет свои преимущества и недостатки, и разработчику веб-приложений важно понимать различные существующие решения для управления сеансами, а также их ограничения. Несмотря на несколько рисков безопасности, связанных с управлением сеансами, существуют простые, но эффективные рекомендации, которые могут значительно повысить безопасность управления сеансами. Мы надеемся, что этот тезис можно рассматривать как общее руководство по поддержанию безопасного сеанса в среде с одним сервером. Однако надежное управление сеансом — это только один элемент безопасного веб-приложения. Чтобы смягчить различные возможные сеансовые атаки, также необходимы хорошие методы проектирования веб-приложений. Для защиты веб-приложений от сессионных атак и нарушений системы необходимы несколько мер глубокоэшелонированной защиты.

3.2.1. Разработка front-end части и обеспечение безопасности передачи информации

При разработке была проведена декомпозиция основной задачи на несколько подзадач, для адекватного программирования веб-сайта:

- управление состоянием приложения, которое состоит из двух экранов, согласно прототипу;
- передача данных на сервер и получение ответа он него;
- построение пользовательского интерфейса;
- обеспечение безопасности информации.

В приложение присутствуют два состояния, созданные с помощью фреймворкаReact. Каждое состояние описывает один из так называемых компонентов, отдельных файлов с расширением JS, содержащий в себе свойства (props)*, переданные из родительского класса и состояния (state), возникающие в коде компонента.

Иерархия файлов, располагаемых на сервере выглядит следующим образом:

- index.html
- js
- js/app.js
- js/components/welcome.js
- js/components/result.js

Компоненты Java, .NET и Flash работают с определенными ограничениями. Они управляются виртуальной машиной, которая является интерфейсом между компонентом и базовой операционной системой. Этот интерпретатор реализует песочницу, и компоненты обычно не получают неограниченный доступ к базовой системе. Основное различие между ActiveX и другими типами толстых клиентов заключается в том, что ActiveX являются собственными

компонентами Windows. Это означает, что после того, как они будут установлены в системе и запущены, будет меньше ограничений на то, что они могут делать, особенно в старых версиях Internet Explorer. Это также открывает двери для традиционных ошибок повреждения памяти, таких как переполнение буфера. Подробное описание того, как безопасно кодировать собственные компоненты, здесь не приводится, но разработчики веб-приложений, рассматривающие возможность использования элементов управления ActiveX в своих приложениях, должны убедиться, что они соответствуют передовым методам безопасного кодирования собственных компонентов. Обзор архитектур безопасности Java, .NET и Flash также выходит за рамки данного руководства, поэтому следует обращаться к специалистам. Оставшаяся часть этого раздела посвящена фундаментальным проблемам безопасности, с которыми сталкиваются компоненты толстого клиента (независимо от их типа), а также общим стратегиям смягчения последствий, которые могут быть реализованы создающими их разработчиками.

3.2.2 Разработка back-end части и обеспечение безопасности передачи информации.

После того, как пользователь запускает приложение происходит динамическая генерация списка на основе, заложенной в БД информации. Для защиты соединения и передаваемых по сети данных используется протокол криптозащиты SSL.

Выполнение функций криптозащиты в Node,js обеспечивает стандартный модуль асимметричного шифрования crypto, в который входят такие параметры защиты как, проверка сертификатов, функции хэш, шифрования и дешифрования данных.

Шифрование по данной функции осуществляется в три действия:

1. Объявление переменной Cipher;
2. Добавление к созданной переменной информации, которую необходимо зашифровать или расшифровать.
3. Окончание процесса и передача зашифрованных данных.

Переменная Cipher создается вызовом метода crypto.createCipheriv(), который принимает два параметра:

- алгоритм;
- ключ.

Переинициализация созданнойпеременной Cipher новой информацией осуществляется с помощью метода [CipherInstance].update(), которому передаются следующие аргументы:

- данные для шифрования;
- шифрование исходных данных;
- шифрование возвращаемого методом значения.

Завершение процесса шифрования осуществляется вызовом метода final(), принимающему кодировку зашифрованных данных.

Выводы по третьему разделу:

1. Для того, чтобы что-то проектировать, нужно понимать, что мы проектируем. Спроектированная модель позволяет разработать технологию обеспечения информационной безопасности веб-сайта, в целях не допустить возможного повреждения и утечки информации.

2. Спроектированная модель обеспечения информационной безопасности веб-сайта в условиях воздействия разнородных факторов позволяет в предлагаемых технологиях разработать защищенный веб-ресурс.

3. Разработанная технология позволяет выстроить защиту трех основных направлений функционирования веб-сайта:

- защита безопасности информации со стороны клиент-серверной части;
- защита безопасности информации со стороны Веб-сервера;
- защита безопасности информационных ресурсов.

4. Разработать предложения по совершенствованию системы обеспечения информационной безопасности Веб-сайта

На базе разработанной модели обеспеченой безопасности веб-сайта были разработаны основные предложения по защите информация, представленные в таблице 13, от таких распространенных уязвимостей и атак, как: аутентификация, авторизация, атаки на стороне клиента, выполнение вредоносных команд, раскрытие конфиденциальной информации,логические ошибки и ошибка программного обеспечения.

Уязвимые		Предложенияпо совершенствованию системы
1	Аутентификация пользователей	Установите уникальный идентификатор (login) для каждого пользователя
		Примите политику паролей пользователей в соответствии с рекомендациями
		Ограничьте количество попыток доступа к учетной записи
2	Авторизация пользователей	Определяйте допуск профиля
		Удаляйте устаревшие права доступа
		Проводите ежедневный мониторинг полномочий пользователей
3	Отслеживать доступы и управлять инцидентами	Планируйте систему ведения журнала
		Информируйте пользователей о внедрении системы ведения журнала
		Защитите средства ведения журнала и информацию, хранящуюся в нем
		Запланируйте процедуры для уведомления о нарушения конфиденциальных анных
3	Защитить сервер	Ограничивайте доступ к инструментам и интерфейсам администрирования олько уполномоченными лицами
		Установить критические обновления
		Обеспечивайте доступность данных
4	Защитить веб-сайт	Убедитесь, что в url-адресах нет пароля или идентификатора
		Убедитесь, что пользовательские записи соответствуют ожидаемым
		Разместите баннер с разрешением на использование cookie-файлов, не ребуемых для предоставления услуги.
		Используйте протокол TLS и проверьте его реализацию
5	Сохранить и обеспечить непрерывность работы сайта	Регулярновыполнятьрезервноекопирование
		Храните резервные носители в надежном месте
		Обеспечьте безопасность при транспортировке резервных копий
		Регулярно планируйте и тестируйте непрерывность работы сайта
6	Использовать криптографические функции	Используйтеалгоритмы, программное обеспечение и библиотеки
		Храните секреты и криптографические ключи в безопасном режиме

Таблица 13 – Предложения по совершенствованию обеспечения информационной безопасности веб-сайта

Заключение

В результате проделанной работы разработана технология управления процессом обеспечения безопасности веб-сайта. В основу были положены результаты и достижения научной школы «Системная интеграция процессов государственного управления», зарегистрированная в реестре ведущих научных школ Правительства СПБ [1].

В основу деятельности школы положен естественно-научный подход, базирующийся на применении закона сохранения целостности объекта [2-3].

В процессе деятельности, особенно при обеспечении безопасности, человек всегда работает с процессами [2-3]. Для гарантированного достижения цели деятельности при обеспечении безопасности необходимо уметь формировать процессы с наперёд заданными свойствами. Не имея условия существования этих процессов, мы не можем их формировать. А это не позволяет формировать процессы с наперёд заданными свойствами [4]. Не имея такой возможности, мы не можем гарантированно достигать цель деятельности. А это возможно только на основе методологии обеспечения безопасности [8].

Не имея методологических основ решения проблем функционирования потенциально опасного объекта в виде условий существования процесса, мы не можем гарантировать достижение цели деятельности. Эта ситуация породила фундаментальную проблему: «Результаты деятельности по решению задач функционирования систем обеспечения безопасности не соответствуют ожиданиям лица, принимающего решения». Лицо, принимающее решение, действует на основе трех категорий. Это Система. Модель. Предназначение. Поэтому необходимо решить две проблемы.

Задача 1. Для разработки системы известны два подхода. Разработка системы на основе анализа. Для решения задачи 1 предлагается использовать для синтеза - закон сохранения целостности объекта (ЗСЦО), который обеспечивает достижение цели функционирования систем обеспечения безопасности. (ЗСЦО – это устойчивая, объективная, повторяющаяся связь свойств объекта и свойств его действий при фиксированном предназначении) [11].

Задача 2. Лицо, принимающее решения, осуществляет функционирование систем обеспечения безопасности на основе модели. Для этого нужно уметь синтезировать адекватные модели. Достижение цели функционирования систем искусственного интеллекта возможно только на основе правильно построенной системы (ППС) и адекватной модели. В известных публикациях такой подход к функционированию систем обеспечения безопасности отсутствует. Поэтому вся конструкция ППС реализована на основе ЗСЦО, что подтверждает целесообразность рассмотрения ЗСЦО как условия существования функционирования систем обеспечения безопасности. Модель функционирования систем обеспечения безопасности основана на системной интеграции четырех процессов. Целевой процесс. Формирование проблемы. Процесс распознания проблемы. Процесс устранения проблемы. Уровень функционирования систем обеспечения безопасности оценивается вероятностью того, что каждая проблема обнаружена и устранена. Результаты

моделирования подтвердили основные тенденции в процессе функционирования систем обеспечения безопасности.

Используемый в ВКР подход позволил получить условие существования процесса обеспечения безопасности в форме аналитической математической модели решения человека, в которой увязываются целевой процесс информационной системы и три базовых процесса обеспечения безопасности. А именно процесс образования угрозы, процесс идентификации угрозы и процесс нейтрализации угрозы.

При реализации подхода в форме технологии управления возникла проблем связи временных характеристик элементов разработанной модели с состояниями процессов обеспечения безопасности. Это стало возможным, благодаря применения сетевых графиков.

Информационная среда – это среда, в которой реализуется совокупность информационных процессов и средств для обслуживания объектов инфраструктуры территориальных образований в интересах обеспечения баланса спроса и предложения на рынке информационных услуг.

В современных условиях информационная среда является одним из основных компонентов системы деятельности территориального образования. Рынку информационных услуг присуща конкурентная борьба и другие процессы, порождающие деструктивные воздействия на информационную среду. Особое место в такой деятельности в настоящее время занимают веб-технологии. Однако в силу деструктивных воздействий результаты деятельности специалистов по информационным технологиям не оправдывают их ожидания. То есть информационные процессы не достигают поставленной цели. Такая ситуация породила проблему обеспечения ИБ.

В данной работе был произведен анализ основных подходов к решению поставленной задачи по трем основным направлениям функционирования ресурса:

- направление «А», защита безопасности информации со стороны клиент-серверной части;
- направление «Б», защита безопасности информации со стороны Веб-сервера;
- направление «В», защита безопасности информационных ресурсов.

Устанавливается, что направления защиты безопасности информации на основе направлений АБВ не в полной мере решает поставленную задачу по свойству уязвимости Веб-сайта.

В связи с чем были разработаны предложения по совершенствованию системы обеспечения информационной безопасности Веб-сайта.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Распределительные информационные системы [Электронный ресурс]. URL:URL:http://www.nsc.ru/win/gis/publ/inf_sys.html/(дата обращения:16.11.2022)
2. Реестр ведущих научных и научно-педагогических школ Санкт-Петербурга[Электронный ресурс]. URL: http://is.ifmo.ru/aboutus/2013/science-schools.pdf/(дата обращения:16.11.2019)
3. Анохин П.К. Системные механизмы высшей нервной деятельности. М.: Наука, 1979. 453 с.
4. Загинайлов Ю. Н.Теория информационной безопасности и методология защиты информации, 2015. 124 с.
5. Статья предотвратить и защитить [Электронный ресурс].URL: https://www.osp.ru/lan/2018/05/13054245/(дата обращения:22.11.2019)
6. Ясенев В.Н., Дорожкин А.В., Сочков А.Л., Ясенев О.В. Учебное пособие Информационная безопасность, 2017. 22с.
7. Бурлов В.Г. Основы моделирования социально-экономических и политических процессов (методология, методы). СПб: изд-во СПбГПУ, 2007. 265 с.
8. Цветков В.Я., Матчин В.Т. Агрегирование информационных моделей //Славянский форум. 2(6). С. 77—81.
9. Гуд Г.Х., Макол Р.Э. Системотехника. М.: Советское радио, 1962. 384 с.
10. В.Г. Бурлов, Н.Н. Попов, Х.А. Гарсия Эскалона. Управление процессом применения космической геоинформационной системы в интересах обеспечения экологической безопасности региона //Ученые записки РГГМУ № 50. Научно-теоретический журнал.—СПб: РГГМУ, 2018.—226 с.
11. Бурлов В.Г. Математические методы моделирования в экономике. Часть 1. СПб: изд-во СПбГПУ, 2007. 330 с.
12. Бурлов В.Г. Методы построения систем поддержки принятия решения, основанные на логико-алгебраической системной концепции математики. (Тезисы доклада) НТК 28-29 10, СПб; ВИКУ им. А.Ф. Можайского, 1999.
13. Оптимизация скорости сайта: как уменьшить время ответа сервера. [Электронный ресурс]. URL: https://netpeak.net/ru/blog/optimizatsiya-skorosti-sayta-kak-umen-shit-vremya-otveta-servera/(дата обращения:15.12.2019)
14. Zabbix [Электронный ресурс]. URL: https://www.zabbix.com/ru/(дата обращения:25.12.2019)
15. Базы данных. Основные положения. [Электронный ресурс]. URL:https://www.intuit.ru/studies/courses/6/6/lecture/160?page=5 /(дата обращения:07.01.2020)
16. Трехуровневая архитектура систем баз данных ANSI/SPARC. [Электронный ресурс]. URL: https://studme.org/93784/informatika/vvedenie_arhitekturu_sistem_dannyh/(дата обращения:08.01.2020)