МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»
Институт Информационных Технологий
Кафедра «Информационные системы и кибербезопасность»

«ДОПУЩЕН К ЗАЩИТЕ»
Зав.кафедрой__PhD Мукашева А.К._____
(ученая степень, звание, Ф.И.О.)
_____ «_____» _____ 2021 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Исследование нарушений работы веб-приложений»

Специальность: Системы Информационной Безопасности_____

Выполнил(а): Мурзабеков Данияр Ерболович_ Группа СИБ-17-2_____
(Ф.И.О.)

Научный руководитель к.т.н., профессор Тынымбаев С.Т._____
(ученая степень, звание, Ф.И.О.)

Консультанты:

по экономической части:
__к.э.н., доцент Габелашвили Кахабер Ревазович_____
_____ «_____» _____2021 г.
(подпись)

по безопасности жизнедеятельности:
__к.т.н., доцент Санатова Тоты Сабиржановна_____
_____ «_____» _____2021 г.
(подпись)

Нормоконтролер:
__старший преподаватель Дмитриева Маргарита Валерьевна_____
_____ «_____» _____2021 г.
(подпись)

Рецензент:
__тех.директор ТОО «Доктор Веб - Центральная Азия» Бугаев В.Н._____
_____ «_____» _____2021 г.
(подпись)

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»

Институт Информационных Технологий_____
Кафедра «Информационные системы и кибербезопасность» _____
Специальность «Системы Информационной Безопасности» _____

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Мурзабекову Данияру Ерболовичу_____
(Ф.И.О.)

Тема проекта «Исследование нарушений работы веб-приложений»
Утверждена приказом по университету № _217_ от «27» _ноября_2020 г.
Срок сдачи законченного проекта «__1__» __июня_____ 2021 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Сайт kazgp.kz компания ТОО «КазГеоПозиция» , утилита для сканирования IP-сетей “nmap”, программа перехватывающая прокси “Burp Suit”

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: основные атаки на веб-приложения; исследование посредством “nmap” на наличие открытых портов, версии служб и даже версии операционной системы. А также произведение фаззинг параметров на наш веб-сайт.

Перечень графического материала (с точным указанием обязательных чертежей): программное средство перехвата прокси “Burp Suit”, контроль доступа на примере сайта “КазГеоПозиция”, смета затрат на разработку веб-приложения.

Основная рекомендуемая литература:

1. Shabtai A., Mimran D., Elovici Y. Evaluation of Security Solutions for Android Systems // arXiv preprint arXiv:1502.04870.
2. Sun M., Tan G. NativeGuard: Protecting android applications from third-party native libraries.

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	доцент, к.э.н. Габелашвили К.Р.	10.04.2021 – 26.05.2021	
Безопасность жизнедеятельности	доцент, к.т.н. Санатова Т.С.	07.04.2021 – 25.05.2021	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Обзор действующих мер сетевой форензики	13.02.2021 – 19.03.2021	
Установка среды для исследования нарушения работы	21.03.2021 – 05.04.2021	
Реализация разведки и атак веб-приложения	07.04.2021 – 04.05.2021	
Экономическое обоснование	10.04.2021 – 26.05.2021	
Безопасность жизнедеятельности	07.04.2021 – 25.05.2021	
Согласование от дипломного руководителя	29.05.2021 – 01.06.2021	

Дата выдачи задания «11» октября 2020 года.

Заведующий кафедрой _____ Мукашева Асель Коптлеуовна
(подпись) (ФИО)

Научный руководитель проекта _____ Маргаров Геворг Иванович
(подпись) (ФИО)

Задание принял к исполнению студент _____ Мурзабеков Данияр Ерболович
(подпись) (ФИО)

Аннотация

В данной дипломной работе мы узнаем как производить разведку веб-приложений Узнаем сколько портов под что были заняты и кем прослушивались. На сколько наш веб-сайт устойчив к перебору паролей ftp доступа. А также как посредством main-in-the-midle выявить внутреннюю структуру веб-сайта, открывающая потенциальную уязвимость для внешнего воздействия. Основной идеей данной работы было выявлений уязвимостей и нарушений работы и последующее описание методологии

Annotation

In this thesis, we will learn how to perform web application intelligence, find out how many ports were used for what and by whom they were tapped. How much our website is resistant to brute-force ftp access passwords. And also how to use man-in-the-middle to identify the internal structure of the website, which opens up a potential vulnerability to external influence. The main idea of this work was to identify vulnerabilities and violations of the work and then describe the methodology

Аннотация

Бұл дипломдық жобада біз веб-қосымшаларды қалай іздеу керектігін білеміз, қанша порт жұмыс істегенін және кім тыңдағанын білеміз. Біздің веб-сайт ftp кіру паролін таңдауға қанша төзімді. Сондай-ақ, man-in-the-middle арқылы веб-сайттың ішкі құрылымын қалай анықтауға болады, ол сыртқы әсердің ықтимал осалдығын ашады. Бұл жұмыстың негізгі идеясы жұмыстың осалдығы мен бұзылуын анықтау және әдіснаманы кейіннен сипаттау болды

Содержание

Введение 6
1. Теоретический аспект исследования нарушения работы веб-приложений 8
1.1 Веб-приложение, определение, виды, отличия 8
1.2 Основные виды кибер-угроз и способы атаки на веб-приложений 10
1.3 OWASP теория, история, классификация 11
1.4 Методология исследования нарушений работы веб-приложений 16
1.5Программное обеспечение необходимое для осуществления процесса исследования нарушений работы веб-приложения.....18
2. Практический аспект исследования нарушений работы веб-приложений на примере веб-сайта компании тоо «казгеопозиция» 19
2.1 Установка среды ОС Linux в Windows и дополнительного программного обеспечения 19
2.2 Исследование веб-приложения ТОО «КазГеоПозиция».....31
2.3 Рекомендации к обеспечению безопасности для веб-приложения компании ТОО «КазГеоПозиция».....44
3. Технико-экономическое обоснование проекта.....48
3.1 Цели и задачи, решаемые в экономической части 48
3.2 Определение объёма и трудоёмкости разработки ПО 48
3.3 Расчёт затрат на разработку программного продукта 49
3.4 Смета затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ.....56
3.5 Расчёт возможной (ориентировочной) цены ВЕБ-ПРИЛОЖЕНИЯ.....57
3.6 Расчёт эксплуатационных затрат при использовании ВЕБ-ПРИЛОЖЕНИЯ.....57
3.7 Расчёт результатов без использования ВЕБ-ПРИЛОЖЕНИЯ 59
3.8 Расчёт основных показателей экономической эффективности 61
3.9 Выводы Веб-приложения технико-экономическому обоснованию 62
4. Безопасность жизнедеятельности.....63
4.1 Эскиз и схема помещения.....63
4.2 Разработка системы место освещения рабочего места 64
4.3 Система и виды производственного освещения 68
4.4 Разработка рабочего места с учётом эргономических требований 70
4.5 Вывод по БЖД.....74
Заключение.....75
Перечень сокращений.....77
Список литературы 79

Введение

В современном обществе онлайн технологии стали обыденной и органичной частью нашей жизни. Обширное место онлайн раздела занимают веб-приложения, являющиеся основой взаимодействия юзеров с серверами обрабатывающими их запросы. Но обычный не подготовленный пользователь подвергается жесткому риску утечки личных данных, кражи частных данных, мошенничества, по статистике в 2019 г. 50% веб-приложений располагают высокий, 39% средний и только 11% имеют невысокий показатель уязвимости.

Согласно статистике веб-приложения располагают следующие риски:

– в 9 из 10 веб-приложений преступники могут проводить атаки на пользователей, в том числе — перенаправлять клиентов на подконтрольный им ресурс, похищать учетные данные с помощью фишинговых атак, заражать компьютер вредоносным ПО;

– несанкционированный доступ к приложению возможен на 39% сайтов, кроме того, в 2019 году полный контроль над системой был получен в 16% веб-приложений, а в 8% систем полный контроль над сервером веб-приложения позволял проводить атаки на локальную сеть организации;

– угроза утечки важных данных присутствует в 68% веб-приложений, среди «утекших» данных на первом месте персональные (47% утечек), а на втором — учетные (31%).

Состав уязвимостей был следующим:

– 82% уязвимостей содержались в коде приложения;

– число уязвимостей, которое в среднем приходится на одно веб-приложение, снизилось по сравнению с 2018 годом в полтора раза, в среднем на одну систему приходятся 22 уязвимости, четыре из которых имеют высокий уровень риска;

– каждая пятая уязвимость — высокого уровня риска.

Начиная с 2018 года в Республике Казахстан была инициирована специальная программа системы защиты для борьбы с кибер-преступностью под названием «Киберщит Казахстана» согласно поручению Первого Президента Республики Казахстан Нурсултана Назарбаева, под контролем Правительства РК и реализуемой Комитетом национальной безопасности. В настоящее время проходит второй этап проекта, в ходе которого будет осуществлен запуск всей защитной инфраструктуры. Для решения вопросов кибер-безопасности и защиты веб-приложений в дипломной работе рассмотрен теоретический аспект исследования нарушений работы веб-приложений и практическая работа с выявлением нарушений работы веб-приложения на примере веб-сайта компании ТОО «КазГеоПозиция». Веб-приложение действующей компании было выбрано не случайно, так как это самый быстрорастущий и уязвимый сектор сети Интернет. В дипломной работе исследуются методы поиска и проверки уязвимостей веб-приложений на выявление распространенных уязвимостей и методы их решения.

Основные задачи дипломной работы:

– рассмотреть и проанализировать теоретический материал;

– определить необходимое программное обеспечение для исследования нарушения работы веб приложений;

– разработать методологию исследования для выявления нарушений безопасности веб-приложения;

– провести практическое исследование веб-приложения компании «КазГеоПозиция» для выявления нарушений безопасности;

– проанализировать экономический аспект проведения исследования веб-приложения компании «КазГеоПозиция».

Научная новизна заключается в использовании современного программного обеспечения в исследовании нарушений безопасности, анализ теоретической информации и исследований как отечественных, так и зарубежных коллег, внедрение современных международных стандартов анализа безопасности веб-приложений на территории Республики Казахстан.

1. Теоретический аспект исследования нарушения работы веб-приложений

1.1 Веб-приложение, определение, виды, отличия

Веб-приложения — это так называемые приложения, которые работают на платформе Web, которые используются для соединения юзера с веб-сервером, работающий по протоколу HTTP и браузер, интерпретирующий страницы HTML. Основное отличие веб-сайта от веб-приложения тем что веб-сайт это информационная, статичная страница.

Статичный веб-сайт это набор подготовленных заранее HTML-файлов, которые лежат на удаленном сервере и отдаются браузеру по запросу. А веб-приложение — это что-то технически более сложное. Тут HTML-страницы генерируются на лету в зависимости от запроса пользователя. Почтовые клиенты, соцсети, поисковики, интернет-магазины, онлайн-программы для бизнеса, это все веб-приложения, наиболее известными примерами веб-приложений являются:

- веб-почта(http://mail.ru);
- интернет-магазины (http://amazon.com);
- онлайновые аукционы (http://ebay.com);
- и другие формируемые на ходу веб-страницы.

Впрочем сфера употребления веб-приложений намного шире, чем электронный бизнес, они употребляются во многих академических и коммерческих областях. впрочем вероятны всевозможные вариации, чаще всего применяется трехуровневая структура ради концепции веб-приложений: веб-браузер, какой нибудь метод предоставления динамического веб-контента и база данных. Веб-браузер высылает запросы среднему уровню, который обслуживает их, вырабатывая запросы к базе данных, и доставляя итоги в пользовательском интерфейсе. преимущественно известным на сегодняшний период платформой для разработки веб-приложений, представляется CMS системы, одной из самых известной из них является WordPress. тут значимость среднего уровня играет веб-сервер с установленным на нем модулем содействия скриптового языка программирования PHP, а в качестве базы данных обозначивает MySQL [1].Достоинства:

- автономность от клиентской и серверной платформы;
- отсутствие требовательности к ресурсам юзера;
- не просит установки на клиентский компьютер какого-нибудь программного предоставления опричь браузера;
- легкость обновления версий веб-приложения, т.к. оно обновляется только один раз — на сервере;
- «встроенные» сетевые способности (возможность действовать с несколькими клиентами одновременно);
- сохраняются все сведения при переходе клиента с машины на машину.

Недостатки и методы их обхода:

– низкое время реакции — установка на клиентские технологии, типа JavaScript, DOM, Flash, XUL;
– протокол HTTP не хранит состояния — механизм cookies, сессии;
– малая безопасность — защищенные соединения HTTPS, авторизация интегрированная в HTTP или на базе сессий;
– неразвитость языка HTML в значении конфигураций — разработка стандарта XForms, Flash, Java applets.

Ныне веб-приложения набирают популярность [1]. Самыми навещаемыми сайтами стают не чисто информационные, гипертекстовые сайты, а те, которые дают какой-нибудь сервис, как-нибудь взаимодействуют с пользователем. Но даже и обычные информационные сайты зачастую применяют системы управления контентом ради удобства управления информацией, так что и их тоже возможно прибавить к веб-приложениям.

В зависимости от стоящих проблем веб-приложения деляться на последующие типы:

– корпоративный портал;
– CRM;
– ERP;
– системы электронной коммерции.

Рассмотрим каждый из этих видов веб-приложений и решаемыми ими задачами.

Корпоративный портал — универсальный веб-сервис, позволяющий удобно и эффективно оптимизировать бизнес процессы. Решаемые задачи:

– улучшение качества работы с клиентами;
– повышение результативности работы сотрудников;
– упрочнение и улучшение связей между подразделениями компании;
– удобное и результативное общение с контрагентами;
– повышение мобильности сотрудников;
– удаленная работа с документами;
– проведение PR-мероприятий различной степени сложности.

CRM — мощный инструмент автоматизации отношений с покупателями, эффективно решающий задачу успешного контроля, планирования и развития любого клиентоориентированного бизнеса.

Решаемые задачи:

– целостность и сохранность клиентской базы;
– получение аналитики по продажам;
– повышение объёма продаж;
– эффективная оптимизация работы персонала;
– сокращение бумажного документооборота.

ERP — организационная стратегия интеграции производства и операций, управления трудовыми ресурсами, финансового менеджмента и управления активами, ориентированная на непрерывную балансировку и оптимизацию

ресурсов предприятия посредством специализированного веб-приложения, обеспечивающего общую модель данных и процессов для всех сфер деятельности. Решаемые задачи:

- стандартизация форм отчетности и информационных систем;
- улучшение взаимодействия между отделами;
- контроль и синхронизация процессов;
- интеграция с контрагентами.

Так называемые системы электронной коммерции, это такая система веб-приложения, при помощи которой производители и поставщики услуг/товаров имеют возможность предлагать свой продукт в сети потенциальным покупателям, осуществлять прием и обработку заказов, управлять статусом заявок. Решаемые задачи:

- получение подробной информации о запросах каждого индивидуального потребителя;
- стремительный вывод нового продукта на рынок;
- уменьшение затрат на совершение сделки;
- сокращение пути товара к потребителю.

Веб-сайт компании ТОО «КазГеоПозиция» является многоцелевым веб-приложением, сочетающим элементы корпоративного портала, CRM и e-commerce.

1.2 Основные виды кибер-угроз и способы атаки на веб-приложений

Вследствии информации об атаках, организованным за 2017 и 1 квартал 2018 года, посчастливилось подробно изучить нынешние кибератаки. Были учреждены первопричины атак: в 70% случаев они были ориентированы на кражу информации о кредитных картах, в 26% — на взламывание вебсайтов, а еще в 4% — на хакинг баз данных приложений. Сопоставив данную информацию с отчетом WhiteHat Security о нынешних уязвимостях, итогах наблюдений по атакам от компании Loryka, информацией по разработке вредоносного ПО от Exploits-DB, базой данных ресурса CVE с вредоносным ПО и перечнем уязвимых приложений и программного обеспечения, которые могут являться им подвержены, преимущественно внушительные новые риски, сопряженные с веб-приложениями.

Инъекционные атаки на службы веб-приложений. Максимальный процент (70%) всех утечек информации в первом квартале 2018 года был сопряжен с веб-инъекциями, которые приводили к краже информации кредитных карт. Инъекционные атаки дозволяют взломщикам подключать свежие команды либо новоизобретенный код конкретно в работающее веб-приложение (вмешиваться в работу приложения) для запуска вредоносной программы [2]. За последнее десятилетие 23% утечек информации были спровоцированы инъекционными атаками SQL, что является, пожалуй, самым худшим видом этих атак. При этом инъекционных уязвимостей (слабые места, которые еще не установлены) остается достаточно много. К примеру, WhiteHat

Security утверждает о том, что инъекционные уязвимости собрали 17% от числа всех свежих замеченных ними уязвимостей в 2017 году.

Эта проблема настолько велика, что OWASP предлагает инъекционные недостатки риском номер один в своем ассортименте из 10 самых ужасных рисков ИБ. По этой причине, поиску, корректированию и блокированию таких уязвимостей надлежит давать первоочередное внимание.

Кража прав доступа. изучение записей об утере информации говорит нам о том , что 13% всех хакингов веб-приложений в 2017 и 1 квартале 2018 года были связаны с правами доступа. Тут положено обозначить 5 главных категорий: материал учетных записей, полученные путем искусвественно созданной электронной почты (34,29%), ложная конфигурация управления правами допуска (22,86%), атаки которые используют перебор всеозможных потенциальных вариантов паролей (5,71%), использование краденных учетных данных (8. 57%) и кражи личной информации с помощью социальной инженерии (2,76%). кроме того имелось обнаружение, которое показывало примерно 25% скриптов ExploitDB для веб-приложений соединены с правами доступа. Хотелось бы добавить, что отчет F5 и института Ponemon изобразил —75% пользователей используют только логин и пароль для аутентификации в критически важных приложениях [2]. В то время как должно извлекать усиленные решения аутентификации, такие как, федеративная или многофакторная аутентификация.

А для наружных приложений, над которыми у нас нет полного контроля, следует применять брокеры сохранности облачного допуска (Cloud Access Security Broker, CASB), которые в свою очередь могут хорошо править процессом аутентификации.

Десериализационные атаки на службы приложений. В 2017 г. десериализационные атаки имелись немногочисленными, впрочем при всём этом нашли нешуточный эффект. беззащитность к десериализации инъекций Apache Struts, к примеру, имелась той дырой, через который хакеры обнаружили американское бюро кредитных историй и забрали учетные записи 148 миллионов американцев и 15,2 миллионов жителей Великобритании. дело сериализации – это преобразование информации в формат, подходящий для их транспортировки; десериализация — дело их возвратной конвертации. Эти атаки становятся все более распространенными, поэтому веб-приложения сейчас презентуют собой сетевые кластеры подсистем, которым понадобятся потоки данных, связанные с сериализацией данных.

Взломщики включают свои установки в сериализованный течение информации и передают их нефильтруемыми напрямую вовнутрь веб-приложений. Так, к примеру, в Exploit-DB вероятно заметить 30 подобных сценариев атак. чтобы остеречься от них, приложения должны анализировать и выбирать все пользовательские входные данные, включительно потоки предоставленных сериализации.

Атаки защиты транспортного уровня. В то время, как 63% респондентов заявили, что они всегда используют SSL/TLS кодирование для своих веб-

приложений, только 46% анкетированных сказали, что пользуются шифрованием SSL/TLS для всех своих веб-приложений (в данном же количестве и оффлайн), порция которых доносится в отдельных организациях 76-100% всей IT-инфраструктуры. также стоит учитывать, что в результате наличия наибольшего числа стандартов зашифровки транспортного значения (SSL и TLS 1. 0), однако они и были скомпрометированы ранее, присутствует непрерывный риск утечки информации или атак типа «man in the middle» («человек посредине»), когда злоумышленники перехватывают материал пользователей. Кроме того, 47% организаций заявили, что используют самозаверяемые сертификаты, что уменьшает авторитетность их приложений, вместо того, для обеспечения применения всеми веб-приложениями благородного уровня зашифровки и наличие у них сообразных сторонних подписанных сертификатов.

Атаки на отказ в обслуживании против определенной части веб-приложения. Атаки типа «отказ в обслуживании» опасны вариативностью возможных целей и могут иногда атаковать зафиксированные бреши в программном обеспечении. Exploit-DB располагает в своей базе данных 5665 уязвимостей, объединенных со схожими атаками. Хочется отметить, что зачастую появляются распределенные атаки типа «отказ в обслуживании» (DDoS) от целой армии устройств под контролем взломщиков и атаки thingbots с прямым или усиленным трафиком, что в свою очередь перегружает приложения. Еще более сильной приходится гибридная атака, что связывает потоки трафика с заранее наставленными атаками на уязвимости в службах веб-приложений.

Данные атаки настроены ради разгрома и манипуляцию инфраструктурой веб-приложений, что должно повергнуть к перегрузке сайта. Например, были обнаружены подобные виды атак с тысячами раздельных IP-адресов и с 2000 запросов страницы в минуту. Атаки DDoS разносятся на всех ступеньках степени веб-приложения, значит очень важно, дабы каждая организация имела стратегию реагирования для атак данного типа.

Создание возможных атак против пользователей ради совершения кражи прав доступа. риск от атак для пользователей веб-приложений нередко недооценивается, поэтому они наставлены на индивидуальные лица, что вряд ли будут упомянуты в публичных докладах о нарушениях, потому что нет никаких регуляций в данной нормативной отчетности подобных атак на веб-приложения. Однако, самым популярным способом завладения информацией пользователя имеется межсайтовый скриптинг (XSS), что приходится одной из наиболее знаменитых уязвимостей (30% уязвимостей найденных WhiteHat Security в 2017 г. и 9,24% сценариев атак на Exploit-DB) [2]. Атаки XSS частенько могут привести к краже учетных информации пользователя или к краже прав доступа. Межсайтовая замена через подделывание запросов (Cross Site Request Forgery, CSRF) представляется еще определенным способом, что сведения клиента могут быть перехвачены при неосознанном запуске несанкционированных команд в веб-сайте.

Оба этих типа атак связанны с данными веб-приложениями, что подвергается вводу вредоносного кода, установленного взломщиком непосредственно для веб-сайта. Но даже при таких критериях специалисты, обслуживающие данные сайты, могут сами снизить число сценариев атак, используя интегрированные опции веб-сервера, таковые как cookie HTTP и доменов, и установку опций X-frame в режиме «отказ».

Атаки пользователей веб-приложений с помощью вредоносного ПО. Не менее значительным способом, через которого пользователи подвергаются атаке, есть вредоносное программное обеспечение, что перехватывает веб-браузер для перехвата учетных данных, используемых для аутентификации в веб-приложение.

А еще, заражённые программы, использующиеся ради "финансовых" логинов, могут использоваться как в браузерах, так и в мобильных приложениях. И если раньше щит устройств юзера в большей степени игнорировалось изза невыполнимости первоклассного контроля, то сейчас, с выходом GDPR, что обещает компаниям большие штрафы за утечку информации юзеров, уровень защиты юзеров приложений увеличится. Еще учитываем, что некоторые брандмауэры веб-приложений могут отслеживать подозрительные подключения, выявляя скомпрометированных клиентов, и фильтровать их доступ.

1.3 OWASP теория, история, классификация

Open Web Application Security Project (OWASP) — такой публичный проект для защиты веб-приложений. Содружество OWASP содержит в себе корпорации, образовательные компании и частных лиц всего мира. Сообщество функционирует над созданием статей, учебных пособий, документации, инструментов и технологий, находящихся в открытом доступе. Фонд OWASP — такая филантропическая организация, которая проявляет помощь и реализовывает администрирование проектами и инфраструктурой OWASP. Более того, OWASP отмечен как некоммерческая организация в Европе с июня 2011 г. OWASP не связан ни с одной компанией, занимающейся разработкой технологий, но он поддерживает высокотехнологичное применение технологий безопасности. Проект не хочет связываться, потому что полагает, что свобода от влияния со стороны иных компаний возможно упростит распространение беспристрастной, нужной и недорогой информации о безопасности приложений. члены общества OWASP создают приложения безопаснее, учитывая человеческий фактор и технологичный уровень. Преимущественно нужные документы, опубликованные OWASP, включают в себя наставлние OWASP, Обзорное наставление по Коду OWASP и обширно потребляемый Проект Топ-10 OWASP. Самыми известными инструментами OWASP являются учебная среда, прокси-анализатор WebScarabи . NET инструменты. OWASP состоит приблизительно из 190 здешних отделений, располагающихся по всему миру и тысяча соучастников в листах рассылки проекта. OWASP создал серию конференций AppSec ради будующей

концепции сообщества, отданного безопасности приложений. OWASP создаёт стандарты, один из которых недавно был опубликован как OWASP Application Security Verification Standard (ASVS)). ключевая цель OWASP ASVS — это стандартизация спектра охвата и степени строгости общедоступных на рынке приложений, обеспечивающих безопасность [3].

Целью OWASP ASVS вдобавок представляло создание комплекта коммерчески эффективных выявленных стандартов, адаптированных для специализированных веб-технологий. Сборник про Веб-приложения ранее был опубликован. Книга про Веб-Сервис в процессе написания. OWASP был создан 9 сентября 2001 года Марком Керфи и Дэннисом Грувзом. И начиная с конца 2003 года, Джефф Вильямс трудился добровольным руководителем OWASP до сентября 2011 года. Нынешний руководитель — Майкл Коатс, а вице-председатель Эойн Кири. Фонд OWASP учредили в 2004 г. и занимается помощью проектов и инфраструктуры OWASP. OWASP служит не личным целям её руководителей, а популизации знаний. Лидеры OWASP в ответе за принятие выводов о техническом руководстве, приоритеты проекта, расписание и издание продукции. Говоря иначе лидеры OWASP могут оцениваться как менеджмент Фонда OWASP. В OWASP официально функционирует 8 человек, из-за чего у проекта очень низкие расходы, которые покрывают конференциями, корпоративными спонсорами и рекламой. OWASP ежегодно удостаивает грантами корпоративных и персональных членов за разработку перспективных приложений, обеспечивающих безопасность. С 2011 года OWASP зарегистрирована как некоммерческая организация в Бельгии под именем OWASP Европа VZW [3].

OWASP классификация использует уровни от A1 до A10. Рассмотрим каждый из них:

– A1 Инъекции — Уязвимости, связанные с внедрением SQL, NoSQL, OS и LDAP. Возникают, когда непроверенные данные отправляются интерпретатору в составе команды или запроса. Вредоносные данные могут заставить интерпретатор выполнить непредусмотренные команды или обратиться к данным без прохождения соответствующей авторизации;

– A2 Недостатки аутентификации — Функции приложений, связанные с аутентификацией и управлением сессиями, часто некорректно реализуются, позволяя злоумышленникам скомпрометировать пароли, ключи или сессионные токены, а также эксплуатировать другие ошибки реализации для временного или постоянного перехвата учетных записей пользователей;

– A3 Разглашение конфиденциальных данных — Многие веб-приложения и API имеют плохую защиту критичных финансовых, медицинских или персональных данных. Злоумышленники могут похитить или изменить эти данные, а затем осуществить мошеннические действия с кредитными картами или персональными данными. Конфиденциальные данные требуют дополнительных мер защиты, например их шифрования при хранении или передаче, а также специальных мер предосторожности при работе с браузером;

– A4 Внедрение внешних сущностей XML — Старые или плохо настроенные XML-процессоры обрабатывают ссылки на внешние сущности внутри документов. Эти сущности могут быть использованы для доступа к внутренним файлам через обработчики URI файлов, общие папки, сканирование портов, удаленное выполнения кода и отказ в обслуживании;

– A5 Недостатки контроля доступа — Действия, разрешенные аутентифицированным пользователям, зачастую некорректно контролируются. Злоумышленники могут воспользоваться этими недостатками и получить несанкционированный доступ к учетным записям других пользователей или конфиденциальной информации, а также изменить пользовательские данные или права доступа;

– A6 Некорректная настройка параметров безопасности — Некорректная настройка безопасности является распространенной ошибкой. Это происходит из-за использования стандартных параметров безопасности, неполной или специфичной настройки, открытого облачного хранения, некорректных HTTP-заголовков и подробных сообщений об ошибках, содержащих критичные данные. Все ОС, фреймворки, библиотеки и приложения должны быть не только настроены должным образом, но и своевременно корректироваться и обновляться;

– A7 Межсайтовое выполнение сценариев — XSS имеет место, когда приложение добавляет непроверенные данные на новую вебстраницу без их соответствующей проверки или преобразования, или когда обновляет открытую страницу через API браузера, используя предоставленные пользователем данные, содержащие HTML- или JavaScript-код. С помощью XSS злоумышленники могут выполнять сценарии в браузере жертвы, позволяющие им перехватывать пользовательские сессии, подменять страницы сайта или перенаправлять пользователей на вредоносные сайты;

– A8 Небезопасная десериализация — Небезопасная десериализация часто приводит к удаленному выполнению кода. Ошибки десериализации, не приводящие к удаленному выполнению кода, могут быть использованы для атак с повторным воспроизведением, внедрением и повышением привилегий;

– A9 Использование компонентов с известными уязвимостями — Компоненты, такие как библиотеки, фреймворки и программные модули, запускаются с привилегиями приложения. Эксплуатация уязвимого компонента может привести к потере данных или перехвату контроля над сервером. Использование приложениями и API компонентов с известными уязвимостями может нарушить защиту приложения и привести к серьезным последствиям;

– A10 Недостатки журналирования и мониторинга — Недостатки журналирования и мониторинга, а также отсутствие или неэффективное использование системы реагирования на инциденты, позволяет злоумышленникам развить атаку, скрыть свое присутствие и проникнуть в другие системы, а также изменить, извлечь или уничтожить данные.

Проникновение в систему обычно обнаруживают только через 200 дней и, как правило, сторонние исследователи, а не в рамках внутренних проверок или мониторинга [3].

1.4 Методология исследования нарушений работы веб-приложений

Для успешного тестирования веб-приложений необходимо применять систематизированный подход или методологию. Наиболее известные это OWASP и WASC. Они являются наиболее полными и формализованными методологиями на сегодняшний день. Есть несколько принципов тестирования, которые считаются наиболее современными и полными:

– DAST – динамический (т.е. требующий выполнения) анализ приложения без доступа к исходному коду и серверной части, по сути BlackBox;

– SAST – статический (т.е. не требующий выполнения) анализ приложения с доступом к исходному коду веб-приложения и к веб-серверу, по сути это анализ исходного кода по формальным признакам наличия уязвимостей и аудит безопасности сервера;

– IAST – динамический анализ безопасности веб-приложения, с полным доступом к исходному коду, веб-серверу — по своей сути является WhiteBox тестированием;

– анализ исходного кода – статический или динамический анализ с доступом к исходному коду без доступа к серверному окружению.

Эти методы подойдут для полного исследования нарушений работы веб-приложения, либо частичному, при исследовании стороннего веб-приложения, к примеру, при участии в программе BugBounty. Основные этапы исследования включают следующие этапы, сформированные исходя из анализа международных стандартов тестирования веб-приложений [3]:

– разведка;
– контроль доступа;
– фаззинг параметров;
– проверка логики работы веб-приложения;
– проверка серверного окружения.

Каждый из этих этапов содержит список определенных действий и использует свои методы исследования, программные или аналитические. Подробный список действий каждого из этапов представлен ниже.

Разведка, подразумевает под собой сбор данных и статистики веб-приложения и включает следующие действия:

– сканирование портов;
– сканирование поддоменов;
– исследование видимого контента;
– поиск скрытого контента (директорий, файлов);
– определение платформы и веб-окружения;
– определение форм ввода.

Контроль доступа подразумевает проверку и тестирования пользовательских и административных функций входа или сессии, включающих следующие действия:

- проверка средств аутентификации и авторизации;
- определение требований парольной политики;
- тестирование подбора учетных данных;
- тестирование восстановления учетной записи;
- тестирование функций сохранения сессии;
- тестирование функций идентификации учетной записи;
- проверка полномочий и прав доступа;
- исследования сессии (время жизни, сессионный токены, признаки, попытки одновременной работы и т.д.);
- проверка CSRF.

Фаззинг параметров подразумевает фиксацию веб-приложения в различных фазах его действительной функциональности, включает в себя следующие действия:

- тестирование приложения к различному виду инъекций (SQL, SOAP, LDAP, XPATH и т.д.);
- тестирование приложения к XSS-уязвимостям;
- проверка HTTP заголовков;
- проверка редиректов и переадресаций;
- проверка выполнения команд ОС;
- проверка локального и удаленного инклуда;
- проверка к внедрению XML-сущностей;
- проверка тимплейт-инъекций;
- проверка взаимодействия веб-сокетов.

Проверки логики работы веб-приложения подразумевает комплексную проверку логики и способов работы, включающий в себя следующие действия:

- тестирование логики работы приложения на стороне клиента;
- тестирования на т.н. «состояние гонки» — race condition;
- тестирование канала передачи данных;
- тестирование доступности информации исходя из прав доступа или его отсутствия;
- проверка возможности дублирования или разделения данных.

Проверка серверного окружения подразумевает тестирование бэкенд структуры, серверной части веб-приложения, включающий в себя следующие действия

- проверка архитектуры сервера;
- поиск и выявление публичных уязвимостей;
- проверка серверных учетных записей (службы и сервисы);
- определение настроек сервера или компонентов (SSL и т.д.);
- проверка прав доступа.

Данная методология, позволяет комплексно и структурировано подойти к процессу исследования нарушений работы веб-приложений. Исходя из веб-приложения, те или иные пункты могут быть дополнены специфичными для данного приложения проверками.

1.5 Программное обеспечение необходимое для осуществления процесса исследования нарушений работы веб-приложения

Для понимания разработанной методологии процесса исследования нарушений работы веб-приложений, на каждом этапе будет использовано следующее программное обеспечение.

Разведка.

Сканирование портов. Лучшая программа для данного вида тестирования — nmap. Необходимо учесть, что по умолчанию nmap сканирует ~1000 портов (первые и популярные выше), а также не сканирует UDP. На этапе сканирования поддоменов необходима работа с утилитой dig и понимание AXFR запросов. Также пригодится утилита subbrute. Исследование видимого контента, используя визуальный анализ, для того чтобы исследовать веб-приложение, понять его логику работы. Для сохранения анонимности, используйте кэш поисковых систем и системы типа google.tranlsate. Для поиска скрытого контента (директорий, файлов, информации), используются утилиты dirb, dirsearch, также возможно использование инструментов Foca, но эта программа считается устаревшей или maltego, но для использования необходима регистрация, бесплатная версия урезана в функционале. Для определение платформы и веб-окружения, необходимо воспользоваться аддоном к браузеру wappalyzer или утилитой whatweb. Определение форм ввода. На этом этапе можно ограничиться визуальным осмотром форм на страницах, выявленных в результате поиска скрытого контента. Отдельно упоминания заслуживают «комбайны» для сбора информации: theharvester и recon-ng — данные инструменты, предназначены для получения большого количества информации, от выявления учетных записей и поддоменов до поиска критичной информации на сайте.

Контроль доступа.

На данном этапе требуется как инструментальная, так и ручная проверка требований парольной политики. Для проверки необходимо провести атаку по словарю, например с помощью hydra или patator, используя заведомо известные учетные данные: таким образом можно выявить защиту от такого рода атак (или ее отсутствие). Определение требований парольной политики. Здесь ручная проверка логики требований политики. Использование только цифр (например как пин-кода) без защиты от брута — очень плохая идея. Тестирование восстановления учетной записи. На данном этапе приходится наличие нескольких ссылок или триггеров для сброса пароля (желательно от разных учетных записей). Здесь необходимо будет выявить и определить хеш (частое явление), например с помощью hashID. Далее необходимо произвести сравнение тригеров сброса (например ссылок) с помощью утилит сравнения

(например comparer в burp suite). Тестирование функций сохранения сессии. Тестирование функций идентификации учетной записи. Проверка полномочий и прав доступа. Исследования сессии (время жизни, сессионный токены, признаки, попытки одновременной работы и т.д.) Проверка CSRF. Для этих задач хорошо подойдет mantra — есть версия как в виде firefox, так и chrome сборки.

Фаззинг параметров.

Тестирование веб-приложения может быть выполнено как в инструментальном режиме (w3af, vega, arachni, sqlmap, Acunetix, Netsparker и.д.), так и полу-инструментальном — Burp Suite, OWASP ZAP и д.р. С помощью этих инструментов, как автоматическом, так и в ручном (наиболее точном) режиме можно выявлять следующие уязвимости: инъекции (SQL, SOAP, LDAP, XPATH и т.д.), XSS-уязвимости, редиректы и переадресации — весь спектр уязвимостей веба (OWASP TOP 10) [4].

Проверки логики работы веб-приложения.

Тестирование логики работы приложения на стороне клиента. Тестирование на т.н. «состояние гонки» — race condition. Тестирование доступности информации, исходя из прав доступа или его отсутствия. Проверка возможности дублирования или разделения данных. На этом этапе нам понадобится хорошо изучить логику работы приложения и эксплуатация с помощью Burp Suite, OWASP ZAP или все той же mantra. Выявление таких уязвимостей в автоматическом режиме практически невозможно (кроме утилит работы с кодом для выявления формальных признаков такого рода уязвимостей и изучения исходного кода).

2. Практический аспект исследования нарушений работы веб-приложений на примере веб-сайта компании тоо «казгеопозиция»

2.1 Установка среды ОС Linux в Windows и дополнительного программного обеспечения

Так как исследование нарушений работы веб-приложений предполагает работу посредством операционной системы Linux или среды операционной системы Linux в Windows, если брать по в Linux, то необходимо произвести установку среды ос Linux в Windows. В Windows 10 присутствует новоиспеченная право установки оболочки Linux аналогичных систем поверху основной. предоставленная функция даёт возможность использовать скрипты bash непостредственно в Windows 10. Для этого необходимо совершить последующие действия:

– необходимо влить элемент «Подсистема Windows для Linux» в «Панель управления» — «Программы и компоненты» — «Включение и отключение частей Windows»;

– после непосредственной установки перезагрузки компьютера, заходим в Магазин приложений Windows 10 и загружаем Kali-Linux;

– запускаем наш дистрибутив как обычное приложение Windows 10 и начинаем совершать первоначальную настройку.

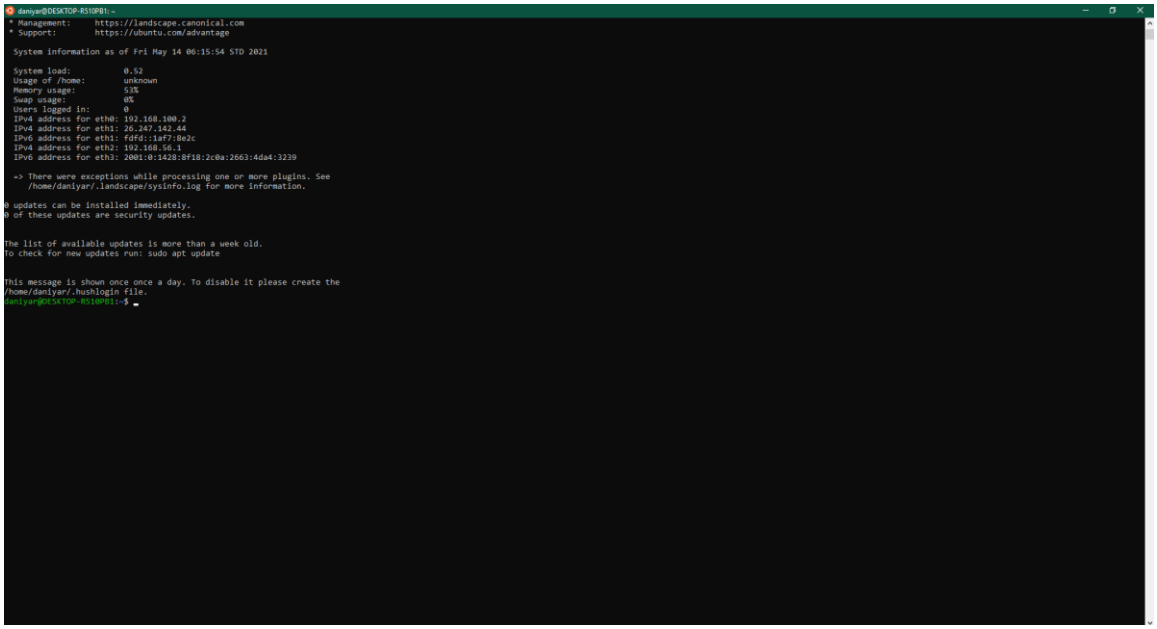


Рисунок 2.1 – Установленная среда kali-linux

После установки Linux среды в виде Kali-Linux, необходимо установить программное обеспечение для тестирования.

Разведка:

– nmap;
– dig;
– subbrute;
– dirb;
– whatweb;
– theharvester.

Контроль доступа:

– hydra;
– hashID;
– burp suite (comparer);
– mantra.

Фаззинг параметров(возможен выбор лишь одного из них):

– w3af;
– vega;
– arachni;
– sqlmap;
– Acunetix;
– Netsparker.

Проверки логики работы веб-приложения:

– Burp Suite;
– OWASP ZAP.

Также, из-за того ,что веб-приложение компании ТОО «КазГеоПозиция разработано на CMS WordPress используем wpscan, который уже имеется в kali-linux.

«WPScan» – blackbox сканер уязвимостей WordPress, написанный на Ruby.

Позволяет выявлять уязвимости в:

– в версии движка;
– темах оформления;
– плагинах [4].

Рассмотрим программы подробнее.

nmap — это аббревиатура от «Network Mapper», на русский язык наиболее понятно и правильно можно перевести как «сетевой картограф». Скорее всего это не лучший вариант перевода на русский язык, но всё же хорошо отображает суть — инструмент для исследования сети и проверки безопасности. Утилита кроссплатформенна, бесплатна, поддерживаются операционных системы Linux, Windows, FreeBSD, OpenBSD, Solaris, Mac OS X. Nmap умеет сканировать различными методами — например, UDP, TCP connect(), TCP SYN (полуоткрытое), FTP proxy (прорыв через ftp), Reverse-ident, ICMP (ping), FIN, ACK, SYN и NULL-сканирование.

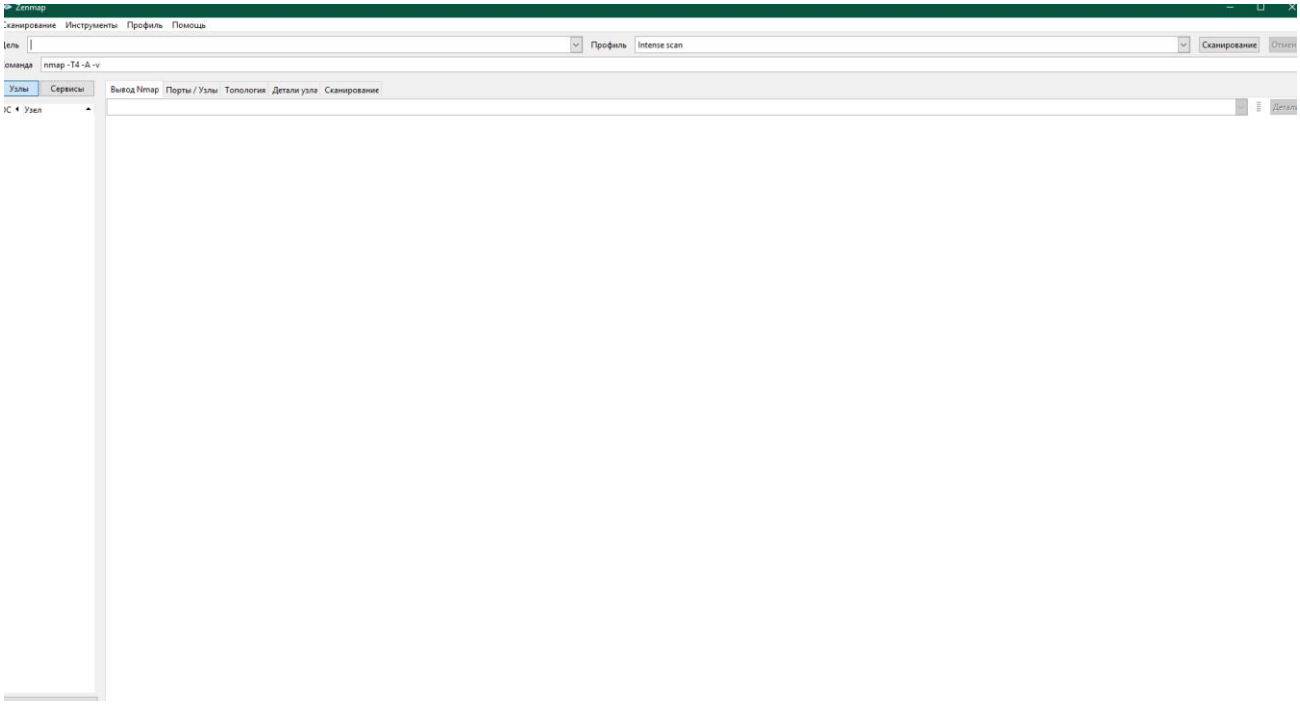


Рисунок 2.2 – Интерфейс программы nmap

dig (Domain Information Groper) – мощный инструмент командной строки для запросов к именам DNS-серверов. С помощью команды dig вы можете

запрашивать информацию о различных записях DNS, включая адреса хостов, почтовые обмены и серверы имен. Это наиболее часто используемый инструмент среди системных администраторов для устранения проблем DNS из-за его гибкости и простоты использования.

```
; <<>> DiG 9.10.3-P4-Ubuntu <<>> MX itsecforu.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6581
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;itsecforu.ru.                  IN      MX

;; ANSWER SECTION:
itsecforu.ru.           1430    IN      MX      10 mx.yandex.net.
itsecforu.ru.           1430    IN      MX      10 mxs.hoster.ru.

;; ADDITIONAL SECTION:
mx.yandex.net.          355     IN      A       213.180.204.89
mx.yandex.net.          355     IN      A       87.250.250.89
mx.yandex.net.          355     IN      A       93.158.134.89
mx.yandex.net.          355     IN      A       213.180.193.89
mx.yandex.net.          355     IN      A       77.88.21.89
mx.yandex.net.          11      IN      AAAA    2a02:6b8::89
mxs.hoster.ru.          670     IN      A       31.28.25.86
mxs.hoster.ru.          670     IN      A       31.28.24.248

;; Query time: 4 msec
;; SERVER: 10.34.240.19#53(10.34.240.19)
;; WHEN: Mon Aug 13 10:53:31 MSK 2018
;; MSG SIZE  rcvd: 237
```

Рисунок 2.3 – Использование команды dig для получения данных о DNS

subbrute — это один из самых популярных и точных инструментов перечисления поддоменов. Проект разработан сообществом и использует открытый определитель имен в качестве прокси, так что SubBrute не отправляет трафик на целевой DNS-сервер. Это не онлайн-инструмент, соответственно его необходимо установить на компьютер. SubBrute можно использовать на Windows или UNIX системах. Программу установить очень легко [5].

```
r00t@r00t-PC:~$ cd Kitploit/subbrute
r00t@r00t-PC:~/Kitploit/subbrute$ python subbrute.py --help
Usage:
subbrute.py [options] target_domain
subbrute.py -p target_domain

Options:
  -h, --help            show this help message and exit
  -s SUBS, --subs=SUBS  (optional) A list of subdomains, accepts a single
                        file, or a directory of files. default = 'names.txt'
  -r RESOLVERS, --resolvers=RESOLVERS
                        (optional) A list of DNS resolvers, if this list is
                        empty it will OS's internal resolver default =
                        'resolvers.txt'
  -t TARGETS, --targets_file=TARGETS
                        (optional) A file containing a newline delimited list
                        of domains to brute force.
  -p, -P                (optional) Print data from found DNS records (default
                        = off).
  -o OUTPUT, --output=OUTPUT
                        (optional) Output to file (Greppable Format)
  -j JSON, --json=JSON  (optional) Output to file (JSON Format)
  --type=TYPE           (optional) Print all reponses for an arbitrary DNS
                        record type (CNAME, AAAA, TXT, SOA, MX...)
  -c PROCESS_COUNT, --process_count=PROCESS_COUNT
                        (optional) Number of lookup theads to run. default =
                        16
  -v, --verbose         (optional) Print debug information.
r00t@r00t-PC:~/Kitploit/subbrute$
```

Рисунок 2.4 – Интерфейс subBrute

dirb — это сканер веб-контента. Он ищет существующие (возможно, скрытые) веб-объекты. В основе его работы лежит поиск по словарю, он формирует запросы к веб-серверу и анализирует ответ. DIRB поставляется с набором настроенных на атаку словарей для простого использования, но вы можете использовать и ваш собственный список слов. Также иногда DIRB можно использовать как классический CGI сканер. Она покрывает некоторые дыры, не охваченные классическими сканерами веб-уязвимостей. DIRB ищет специфические веб-объекты, которые другие сканеры CGI не ищут. Она не ищет уязвимости и не ищет веб-содержимое, которое может быть уязвимым [5].

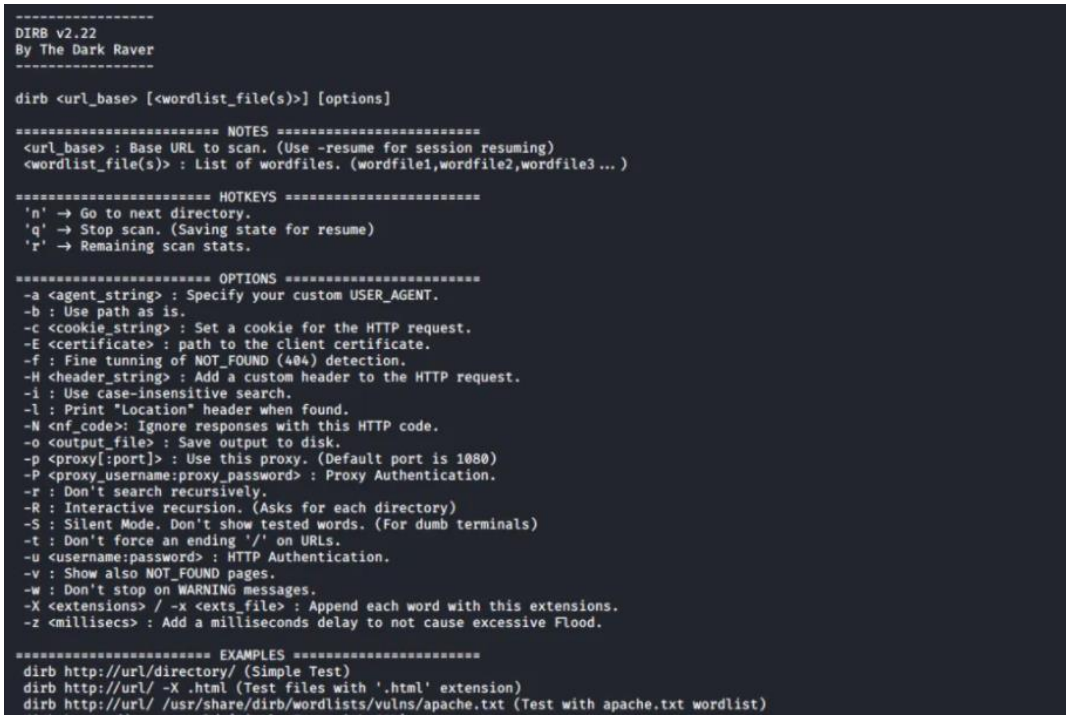


Рисунок 2.5 – Интерфейс dirb

Whatweb применяется для идентификации сайтов и веб-приложений. Цель данной программы ответить на вопрос «Чем представляется данный веб-сайт? WhatWeb распознает веб-технологии, и даже систему управления контентом (CMS), платформы которая служит для ведения блогов, пакеты статистики/аналитики, библиотеки непосредственно JavaScript, веб-сервера и интегрированные устройства. WhatWeb обладает более 1700 плагинов, отдельный из которых для распознавания чего-то одного. WhatWeb вдобавок идентифицирует эти номера версий, email адреса, ID аккаунтов, модули веб-платформ, SQL ошибки и другое [5].

Возможно неприметным и прытким скрупулезным и медленным. Дабы иметь контроль компромисс среди быстроты и достоверности, WhatWeb поддерживает уровни агрессивности. При посещении сайтов в вашем браузере, передаваемые сведения вводят множество подсказок о технологиях, которые лежат в базе работы веб-сайта. Временами одно посещение веб-страницы

охватывает довольно много информации для идентификации веб-сайта, но когда сего недостаточно, WhatWeb сможет продлить опрос веб-сайта.

Уровень агрессивности по умолчанию именуется «незаметный», он приходится самым прытким и требует только один HTTP запрос к веб-сайту. Что подходит для сканирования общественных веб-сайтов. Более враждебные режимы создавались для применения в исследованиях на проникновение.

Рисунок 2.6 – Интерфейс whatWeb

theharvester — это инструмент для сбора e-mail адресов, имён поддоменов, виртуальных хостов, открытых портов/банеров и имён работников из различных открытых источников (поисковые системы, сервера ключей pgp). Это по-настоящему простой инструмент, но эффективный на ранних этапах тестирования на проникновение или чтобы узнать, какую информацию могут собрать о вашей компании через интернет.

Рисунок 2.7 – интерфейс theHarvester

hydra — программа, которая поддерживает огромное количество служб, благодаря своей быстроте и надёжности она завоевала заслуженную признательность среди тестеров на проникновение. Программа прекрасно компилируется и работает на Linux, Windows/Cygwin, Solaris, FreeBSD/OpenBSD, QNX (Blackberry 10) и OSX. В настоящее время поддерживаются следующие протоколы: Asterisk, AFP, Cisco AAA, Cisco auth, Cisco enable, CVS, Firebird, FTP, HTTP-FORM-GET, HTTP-FORM-POST, HTTP-GET, HTTP-HEAD, HTTP-POST, HTTP-PROXY, HTTPS-FORM-GET, HTTPS-FORM-POST, HTTPS-GET, HTTPS-HEAD, HTTPS-POST, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MS-SQL, MYSQL, NCP, NNTP, Oracle Listener, Oracle SID, Oracle, PC-Anywhere, PCNFS, POP3, POSTGRES, RDP, Rexec, Rlogin, Rsh, RTSP, SAP/R3, SIP, SMB, SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, Teamspeak (TS2), Telnet, VMware-Auth, VNC и XMPP [5].

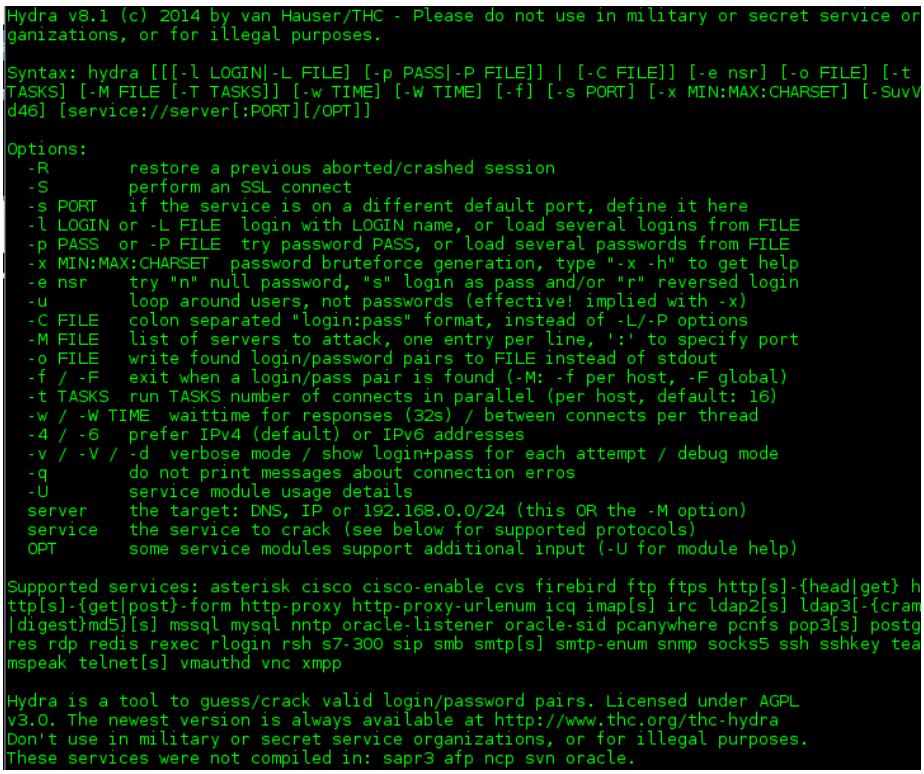


Рисунок 2.8 – Интерфейс thc-hydra

hashID — Идентифицирует различные типы хешей, используемых для шифрования данных, в первую очередь, паролей. Является заменой для программы hash-identifier, которая устрела. hashID — это инструмент, написанный на Python 3, который поддерживает идентификацию более 220 уникальных типов хешей используя регулярные выражения. Подробный список поддерживаемых хешей можно найти здесь. Программа может идентифицировать единичный хеш, разобрать (парсить) файл или прочитать множество файлов в директории и идентифицировать хеши внутри них. hashID также может выводить режим hashcat и/или формат JohnTheRipper, соответствующий идентифицированному хешу [5].

```
[mial@HackWare ~]$ hashid '3af0389f093b181ae26452015f4ae728:user'
Analyzing '3af0389f093b181ae26452015f4ae728:user'
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Domain Cached Credentials
[mial@HackWare ~]$
```

Рисунок 2.9 – Инструмент hashID

burp suite (comparer) — это платформа для проведения аудита безопасности веб-приложений. Содержит инструменты для составления карты веб-приложения, поиска файлов и папок, модификации запросов, фаззинга, подбора паролей и многое другое. Также существует магазин дополнений BApp store, содержащий дополнительные расширения, увеличивающие функционал приложения. Стоит отметить и появление в последнем релизе мобильного помощника для исследования безопасности мобильных приложений — MobileAssistant для платформы iOS. Burp Suite — это интегрированная платформа, предназначенная для проведения аудита веб-приложения, как в ручном, так и в автоматических режимах. Содержит интуитивно понятный интерфейс со специально спроектированными табами, позволяющими улучшить и ускорить процесс атаки. Сам инструмент представляет из себя проксирующий механизм, перехватывающий и обрабатывающий все поступающие от браузера запросы. Имеется возможность установки сертификата burp для анализа https соединений [6].

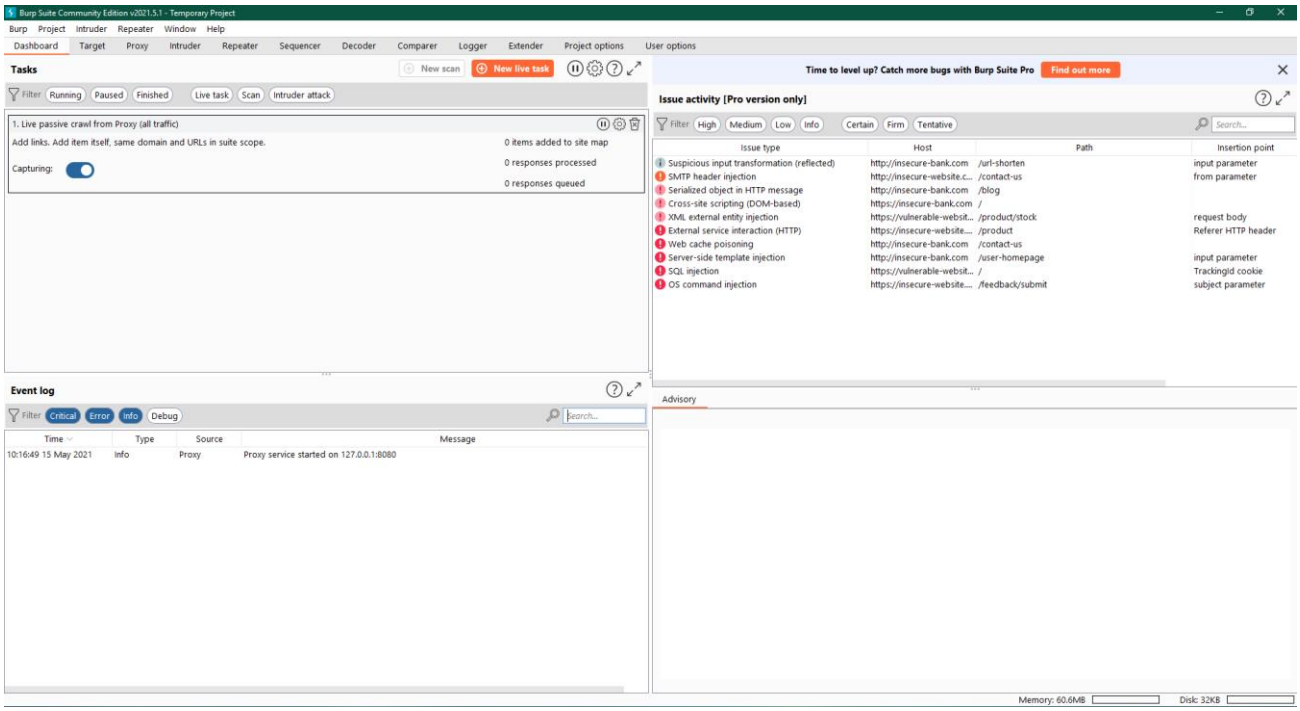


Рисунок 2.10 – Интерфейс burpSuite в операционной системе Windows

Mantra — это браузер, специально разработанный для тестирования безопасности веб-приложений. Имея такой продукт, все больше людей узнают о легкости и гибкости, позволяющих следовать основным процедурам тестирования в браузере. Mantra считает, что такая портативная, простая в использовании и вместе с тем мощная платформа может быть полезна для отрасли безопасности. У Mantra есть много встроенных инструментов для изменения заголовков, управления входными строками, повторного воспроизведения запросов GET / POST, редактирования файлов cookie, быстрого переключения между несколькими прокси-серверами, управления

принудительными перенаправлениями и т. д. Это все делает его хорошим программным обеспечением для выполнения основных проверок безопасности, а иногда и для эксплуатации.

Рисунок 2.11 – Браузер mantra

W3af — это фреймворк атаки и аудита веб-приложений, его целью является идентификация и эксплуатирование всех уязвимостей веб-приложений. Этот пакет снабжён графическим интерфейсом (GUI) для фреймворка. Если вам нужно только приложение командной строки, то установите w3af-console. Этот фреймворк назывался «metasploit для web», но в действительности теперь он намного больше чем это, поскольку он также находит уязвимости веб-приложений, используя техники сканирования по принципу чёрного языка! Ядро w3af и его плагины полностью написаны на Python. Проект имеет более чем 130 плагинов, которые идентифицируют и эксплуатируют SQL-инъекции, межсайтовый скриптинг (XSS), инклуд удалённых файлов и много другое [6].

```
w3af>>> help
|-------------------------------------------------------------------------------------------------------------------------------|
| start                 | Start the scan.                                                                                       |
| plugins               | Enable and configure plugins.                                                                         |
| exploit               | Exploit the vulnerability.                                                                            |
| profiles              | List and use scan profiles.                                                                           |
| cleanup               | Cleanup before starting a new scan.                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------|
| help                  | Display help. Issuing: help [command] , prints more specific help about "command"                    |
| version               | Show w3af version information.                                                                        |
| keys                  | Display key shortcuts.                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------|
| http-settings         | Configure the HTTP settings of the framework.                                                         |
| misc-settings         | Configure w3af misc settings.                                                                         |
| target                | Configure the target URL.                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------|
| back                  | Go to the previous menu.                                                                              |
| exit                  | Exit w3af.                                                                                            |
|-------------------------------------------------------------------------------------------------------------------------------|
| kb                    | Browse the vulnerabilities stored in the Knowledge Base                                               |
|-------------------------------------------------------------------------------------------------------------------------------|
w3af>>>
```

Рисунок 2.12 – Интерфейс w3af

Vega — довольно мощный, универсальный сканер уязвимостей и сниффер пакетов. Интерфейс достаточно простой и не перегружен функционалом. Сканер позволяет проводить автоматический аудит сайта на наличие наиболее распространённых уязвимостей (OWASP Top 10), автоматически ранжирует найденные уязвимости по серьёзности и отображает

исчерпывающую информацию (на какой страничке найдена уязвимость, её полное описание и используемый эксплоит, как исправить, отправленный/полученный TCP пакет). Из минусов — разработчики довольно медленно развивают продукт и на данный момент (07/2016) по-прежнему не реализована выгрузка результатов во внешние форматы (отчёты) [7].

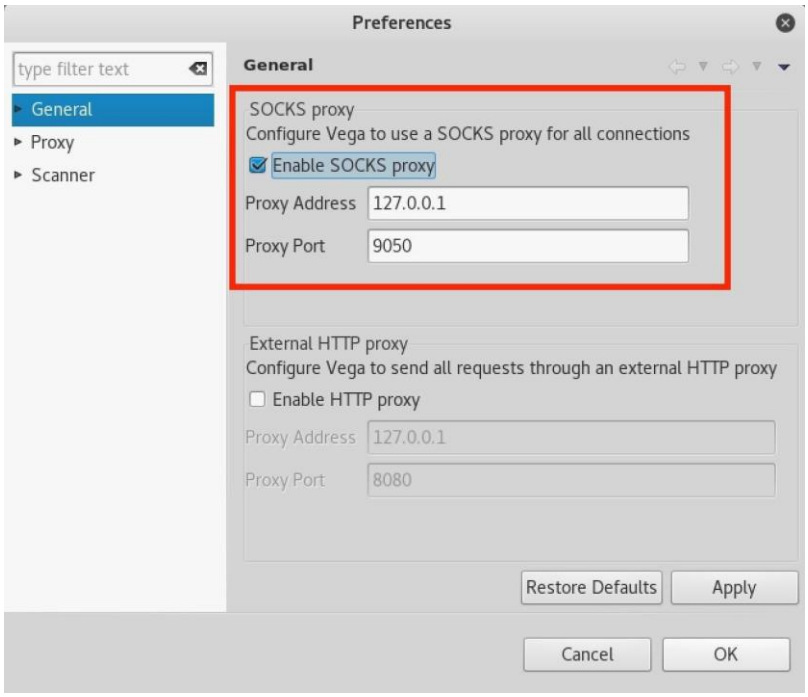


Рисунок 2.13 – Интерфейс vega

Arachni — полностью автоматизированная система, которая в полную силу проверяет веб-сайт, веб-приложения.

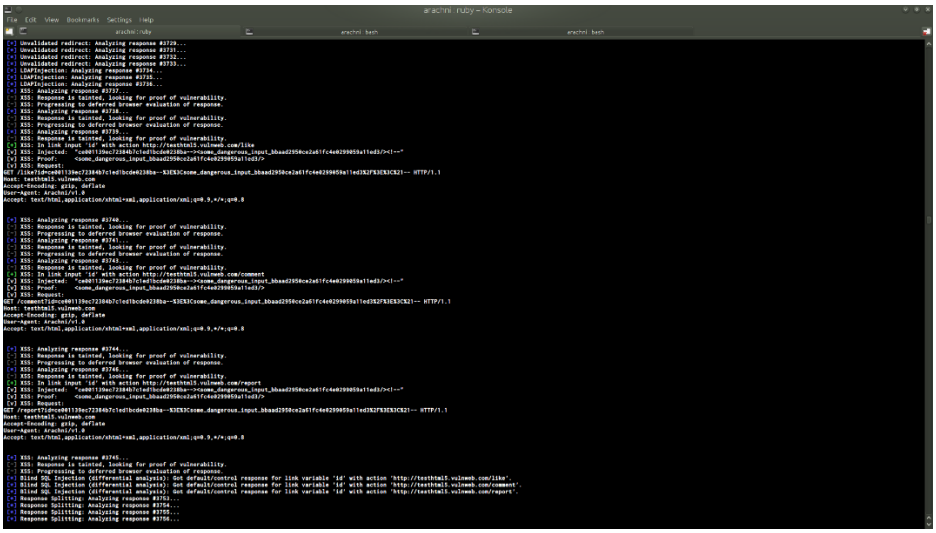


Рисунок 2.14 – Интерфейс arachni

Sqlmap — это инструмент с открытым исходным кодом для тестирования на проникновение, который автоматизирует процесс выявления и эксплуатации уязвимости SQL-инъекция и захват серверов баз данных. Он поставляется с мощным движком выявления и многими нишевыми функциями для конечного тестера на проникновение, имеет широкий набор возможностей, начиная от сбора отпечатков баз данных по полученной от них данным, до доступа к файловой системе и выполнения команд в операционной системе посредством внеполосных (out-of-band) подключений.

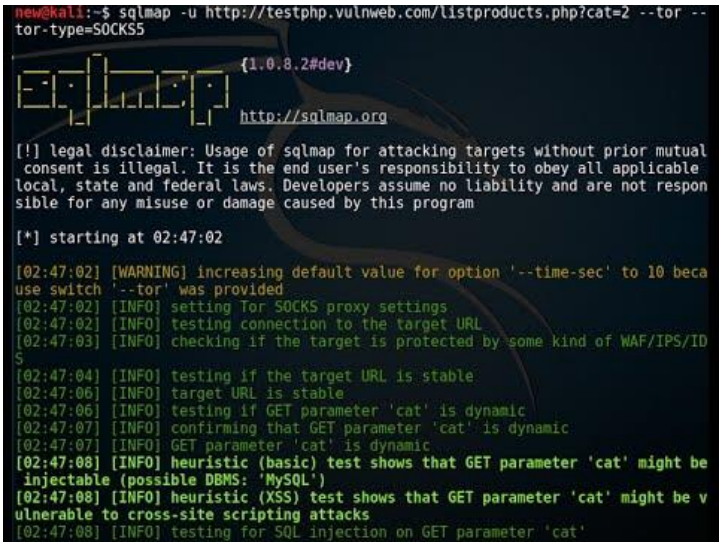


Рисунок 2.15 – Интерфейс sqlmap

Netsparker — полностью автоматический сканер защищенности/ безопасности веб-приложений (DAST, black box scanner, Web Application Security Scanner, Web Application Vulnerability Scanner) имитирует атаки на сотни и тысячи веб-приложений и анализирует их реакции для выявления уязвимостей и проблем безопасности. Простота в использовании и получении инкрементальных и ретроспективных отчетов, интеграция CI/CD, ticketing/bug tracking systems, REST API [7].

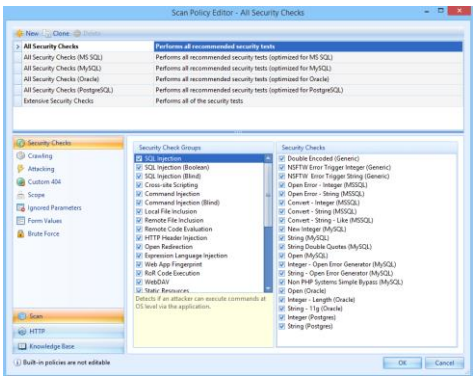


Рисунок 2.16 – Интерфейс netsparker

OWASP ZAP всемирная некоммерческая организация, деятельность которой направлена на повышение безопасности ПО. OWASP ZAP (Zed Attack Proxy) — один из самых популярных в мире инструментов безопасности. Это часть сообщества OWASP, а значит, что этот инструмент абсолютно бесплатный.

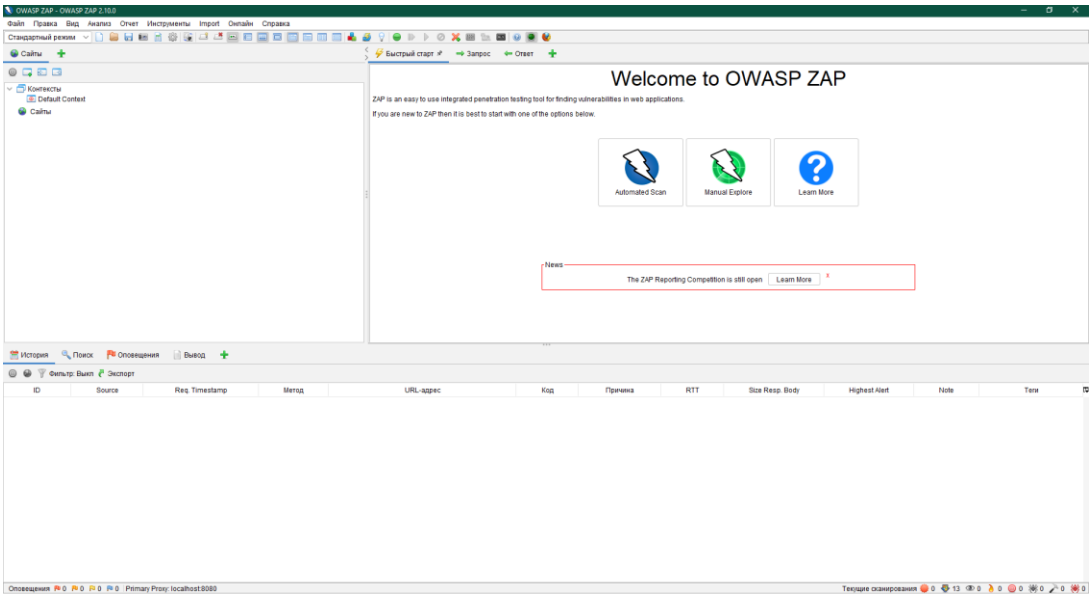


Рисунок 2.17 – Интерфейс OWASP ZAP в операционной системе Windows

После установки программного обеспечения переходим к непосредственному тестированию веб-приложения ТОО «КазГеоПозиция»

2.2 Исследование веб-приложения ТОО «КазГеоПозиция»

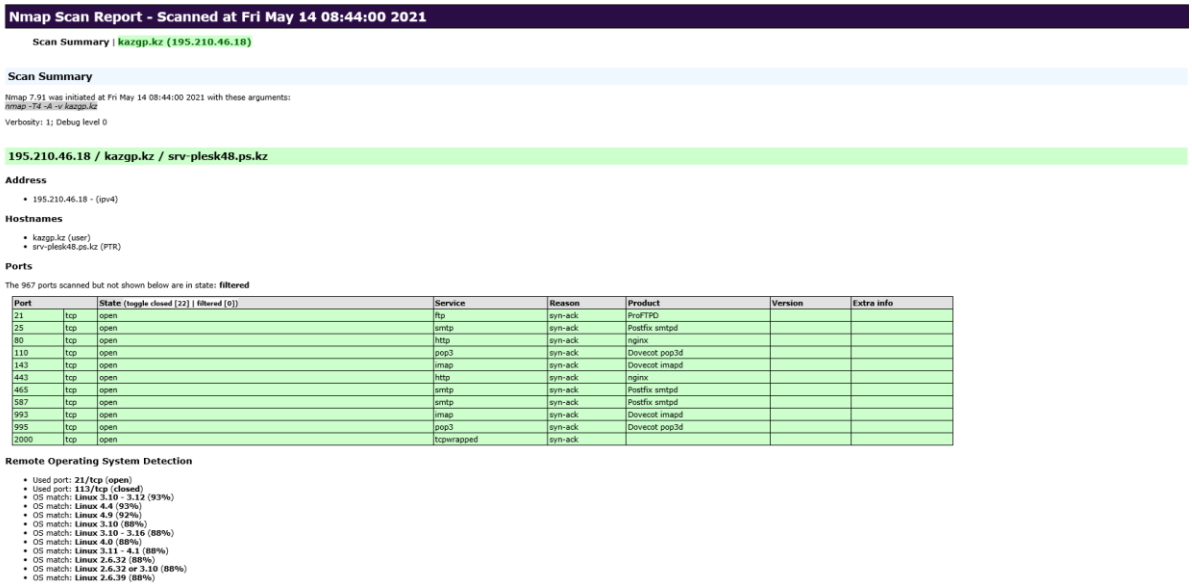
В качестве практической реализации дипломной работы выступает исследование нарушений работы веб-приложений на примере веб-сайта компании ТОО «КазГеоПозиция»

Рисунок 2.18 – Веб-сайт компании ТОО «КазГеоПозиция»

Произведем разведку с помощью nmap. Необходимо помнить, что если мы явно не задаем диапазон портов, то будет просканировано 1000 самых распространенных портов, если укажем ключ -F, то 100, а если -p-, то все 65535.

Из результатов сканирования портов мы можем узнать не только какие сетевые порты открыты, но и версии служб, если использовали ключ -sV, а также предположительную версию операционной систем (ключ -O).

Произведя сканирование, получаем результат и выгружаем его в виде xml файла.



Nmap Scan Report - Scanned at Fri May 14 08:44:00 2021

Scan Summary | kazgp.kz (195.210.46.18)

Scan Summary

Nmap 7.91 was initiated at Fri May 14 08:44:00 2021 with these arguments:
nmap -T4 -sV kazgp.kz

Verbosity: 1; Debug level 0

195.210.46.18 / kazgp.kz / srv-plesk48.ps.kz

Address

- 195.210.46.18 - (ipv4)

Hostnames

- kazgp.kz (user)
- srv-plesk48.ps.kz (PTR)

Ports

The 967 ports scanned but not shown below are in state: **filtered**

Port		State (toggle closed [22] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	ProFTPD		
25	tcp	open	smtp	syn-ack	Postfix smtpd		
80	tcp	open	http	syn-ack	nginx		
110	tcp	open	pop3	syn-ack	Dovecot pop3d		
143	tcp	open	imap	syn-ack	Dovecot imapd		
443	tcp	open	http	syn-ack	nginx		
465	tcp	open	smtp	syn-ack	Postfix smtpd		
587	tcp	open	smtp	syn-ack	Postfix smtpd		
993	tcp	open	imap	syn-ack	Dovecot imapd		
995	tcp	open	pop3	syn-ack	Dovecot pop3d		
2000	tcp	open	tcpwrapped	syn-ack			

Remote Operating System Detection

- Used port: **21/tcp (open)**
- Used port: **113/tcp (closed)**
- OS match: **Linux 3.10 - 3.12 (93%)**
- OS match: **Linux 4.4 (93%)**
- OS match: **Linux 4.9 (92%)**
- OS match: **Linux 3.10 (88%)**
- OS match: **Linux 3.10 - 3.16 (88%)**
- OS match: **Linux 4.0 (88%)**
- OS match: **Linux 3.11 - 4.1 (88%)**
- OS match: **Linux 2.6.32 (88%)**
- OS match: **Linux 2.6.32 or 3.10 (88%)**
- OS match: **Linux 2.6.39 (88%)**

Рисунок 2.19 – Отчет nmap в виде .xml таблицы

В дальнейшем данный файл можно использовать для поиска узлов с определенными портами и вести рабочие заметки с результатами тестирования каждого узла. Сейчас же мы узнали, что порты:

- 21 занят под ftp;
- 25 под smtp;
- 80, 443 прослушивается NGINX;
- 110, 995 для почтового протокола под pop3;
- 143, 993 для почтового протокола под imap;
- 465 и 587 для почтового протокола smtp.

При этом не было обнаружено первично незащищенного порт под базу данных MySQL, что говорит о том, что он скрыт. Используем скрипт, который проверит уязвимости в нашем ПО на сервере. Для этого запускаем следующую команду с указанием портов, которые мы будем проверять [7].

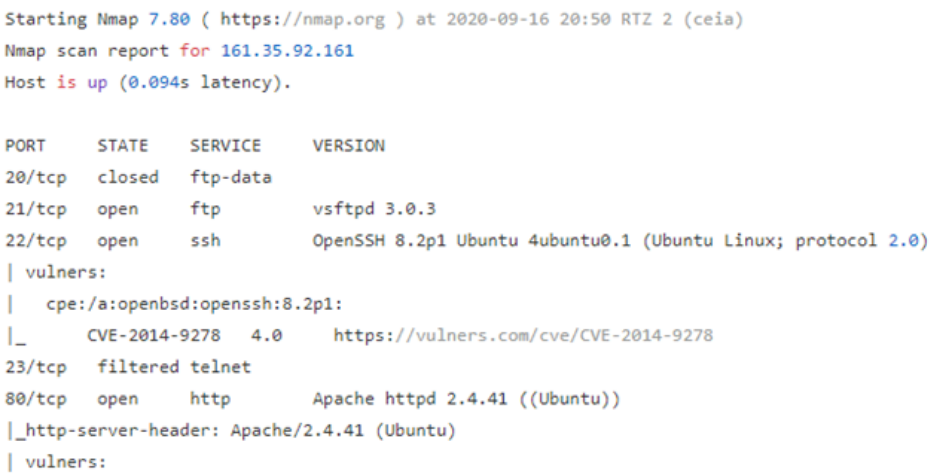


Рисунок 2.20 – Начало проверки с использованием скрипта vulners

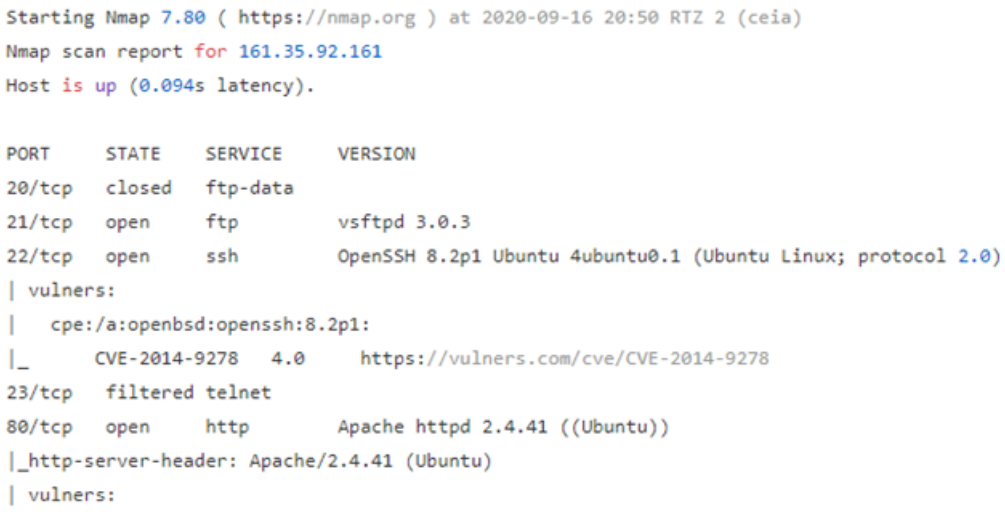


Рисунок 2.21 – Результат проверки с использованием скрипта vulners

Как мы видим из отчета, скрипт изучил активное программное обеспечение нашего сервера и показал ссылки с отображением любой обнаруженной уязвимости. Вдобавок возможно вписать результат анализа в файл, что позднее позволено скинуть серьезному разрабу либо системному администратору. Непосредственно файл итогов будет пребывать в каталоге, из которого мы запускаем скрипт.

Пример такой команды: nmap -T5 -sV -Pn kazgp. kz --script=vulners. nse -p22,80,443,8080,8443,3306,20,21,23 > result. Из описанного выше, в общедоступных сетевых папках есть шанс , что масса нужной для злоумышленника информации. Имеет смысл отыскивать исходные папки как с анонимной учетной записью (пустой логин/пустой пароль), так и с учетной записью обыкновенного юзера. Для поиска SMB-ресурсов нужно прибегнуть модулем auxiliary/scanner/smb/smb_enumshares, а для NFS:

```
msf auxiliary(smb_enumshares) > use auxiliary/scanner/nfs/nfsmount
msf auxiliary(nfsmount) > set RHOSTS 192.168.1.100
RHOSTS => 192.168.1.100
msf auxiliary(nfsmount) > run

[+] 192.168.1.100:111     - 192.168.1.100 NFS Export: / [*]
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Рисунок 2.22 – Результат сканирования smb_enumshares

В нашем случае nmap определил, что на сервере есть ssh, ftp и mysql. Попробуем проверить насколько устойчивые пароли используются.
Вводим следующую команду (напомню, что вводить нужно либо в консоль, либо в поле "Команда" программы Zenmap GUI.
nmap --script ssh-brute -p21 195.210.46.18 --script-args userdb=users.lst,passdb=passwords.lst

В случае успеха должны были отразиться подобранные пары логин/пароль, но в данном случае. Пара логин/пароль не были подобраны. Что говорит о устойчивости к перебору паролей [7].

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-14 09:54
Nmap scan report for srv-plesk48.ps.kz (195.210.46.18)
Host is up (0.040s latency).

PORT   STATE    SERVICE
22/tcp filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 1.13 seconds
```

Рисунок 2.23 – Результат перебора паролей

Проверка ftp порта с помощью команды nmap -d --script ftp-brute -p 21 195.210.46.18 Аналогично, пара логин/пароль не были подобраны. Что говорит о устойчивости к перебору паролей ftp доступа.

```
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE Timing: About 0.00% done
NSE: [ftp-brute 195.210.46.18:21] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute 195.210.46.18:21] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute 195.210.46.18:21] passwords: Time limit 10m00s exceeded.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
```

Рисунок 2.24 – Запуск проверки ftp-brute

```
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
NSE: Finished ftp-brute against 195.210.46.18:21.
Completed NSE at 10:45, 607.25s elapsed
Nmap scan report for srv-plesk48.ps.kz (195.210.46.18)
Host is up, received echo-reply ttl 59 (0.044s latency).
Scanned at 2021-05-14 10:35:25  for 608s

PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 59
| ftp-brute:
|   Accounts: No valid accounts found
|_  Statistics: Performed 18740 guesses in 607 seconds, average tps: 32.2
Final times for host: srtt: 44000 rttvar: 33000  to: 176000

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 10:45
Completed NSE at 10:45, 0.00s elapsed
Read from C:\Program Files (x86)\Nmap: nmap-payloads nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 607.96 seconds
           Raw packets sent: 5 (196B) | Rcvd: 2 (72B)
```

Рисунок 2.25 – Результат ftp-brute

Следующая проверка заключается в проверке доступности анонимного входа в MySQL, командой nmap -sV --script=mysql-empty-password <target>

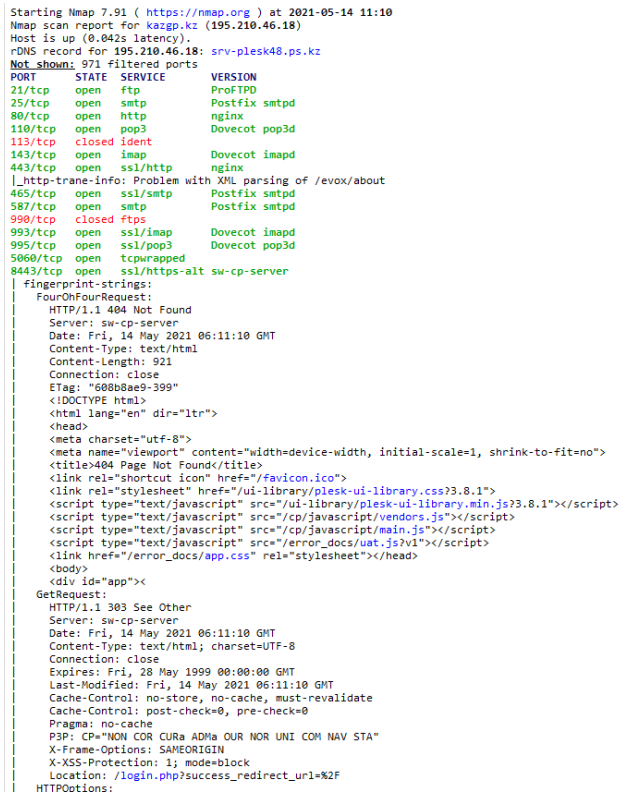


Рисунок 2.26 – Результат script=mysql-empty-password

Проводим проверку формы авторизации с помощью команды nmap -p80 --script http-auth-finder kazgp.kz В результате первичной разведки с помощью nmap веб-приложения kazgp.kz компании ТОО «КазГеоПозиция», уязвимостей найдено не было.

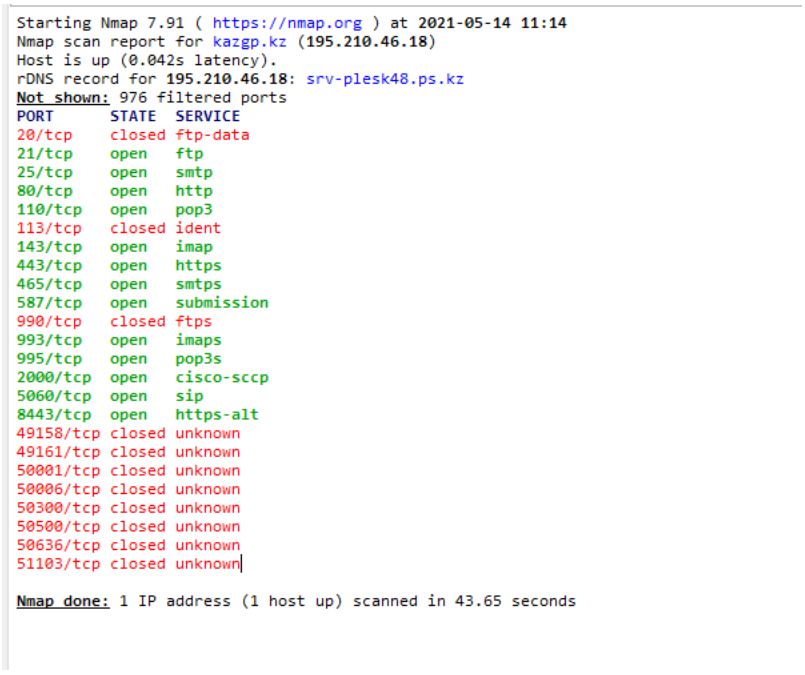


Рисунок 2.27 – Результат script http-auth-finder

После того, как нашли страницы с авторизацией, можно попробовать подобрать пароль и логин для входа в админку сайта. Параметры:

– http-brute.hostname - имя хоста;

– http-form-brute.path - адрес страницы с формой или адрес с API;

– http-brute.method - тип метода, по умолчанию POST;

– http-form-brute.uservar - устанавливает имя переменной, которая отвечает за username. Если не установлено, то скрипт возьмет имя поля из формы;

– http-form-brute.passvar - устанавливает имя переменной, которая отвечает за пароль. Если не установлено, то скрипт возьмет имя поля из формы.

Параметры нужно перечислять через запятую после -script-args.

Ежели ваша формы авторизации применяет cookies параметры либо csrf-token, то тогда выдаст ошибку. (И сие хорошо, следовательно базовую защиту вы предусмотрели). Но в качестве защиты стоит использовать стойкие пароли, а также уменьшать обилие запросов с одного IP-адреса.

Розыск спрятанных папок и файлов исполняется проверкой в програмке nmap с помощью команды: nmap -sV -p 80 -T5 --script http-enum kazgp.

Контролируем на SQL инъекции, так повелось, что большинство нынешних веб-приложений в той или иной мере применяют SQL базы данных. Как правило характеристики веб-страницы или какие-нибудь пользовательские исходные подставляются в SQL запросы и итоги запроса показываются на веб-странице. Ежели передаваемые характеристики ужасно фильтруются, то веб-сервис делается уязвимым для SQL инъекций. Ежели веб-сайт уязвим и осуществляет такие инъекции, то по сути есть возможность создавать с БД (чаще всего это MySQL) что угодно. Только этаким типом чаще всего крадут базы юзеров и их личные данные. При помощи скриптов бегло и эффективно проконтролируем веб-приложение kazgp. kz на уязвимости, с поддержкою инструмента sqlmap [7].

Производим поиск следующей командой:

python sqlmap.py -u http://161.35.92.161/page.php?id=2 --dbs -o -random-agent. Параметр --dbs означает, что нам интересны имена баз данных. В случае успеха и наличия уязвимости, после определения баз данных можно перейти к поиску таблиц и получения нужных данных. Команду необходимо вводить в консоль.

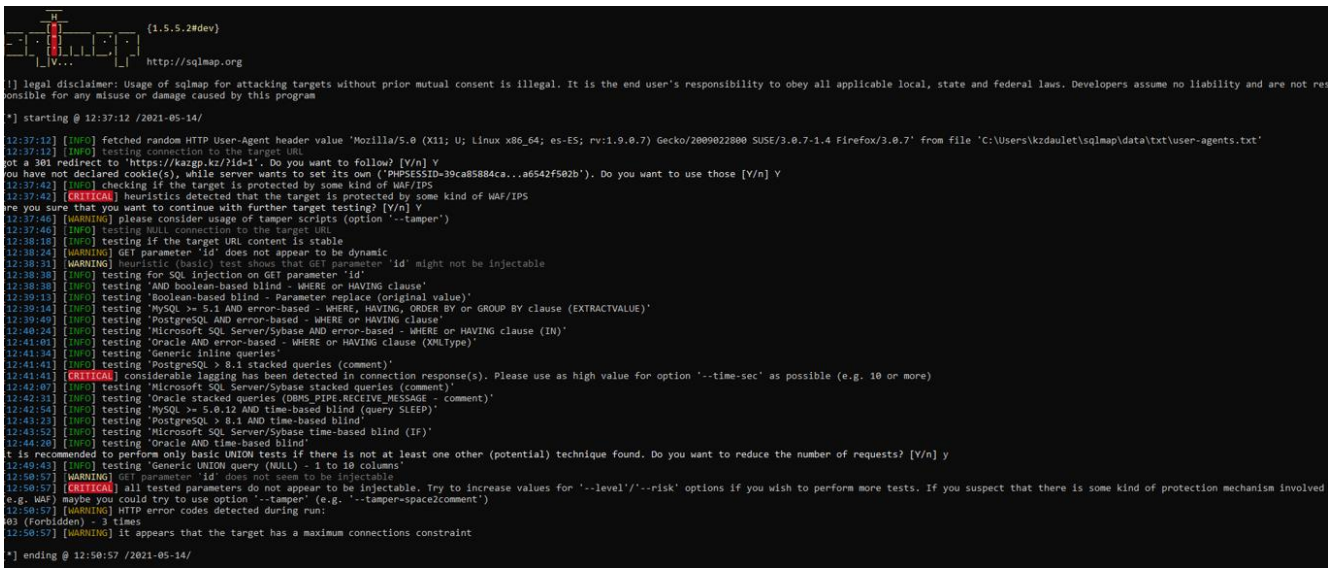


Рисунок 2.28 – Исследование sqlmap.py

Исходя из полученных данных веб-приложение kazgp.kz имеет повышенную уязвимость перед ddos атаками, при этом веб-приложение устойчиво для SQL-инъекций.

Следующим этапом исследования веб-приложения kazgp.kz является контроль доступа с помощью программы burp suite

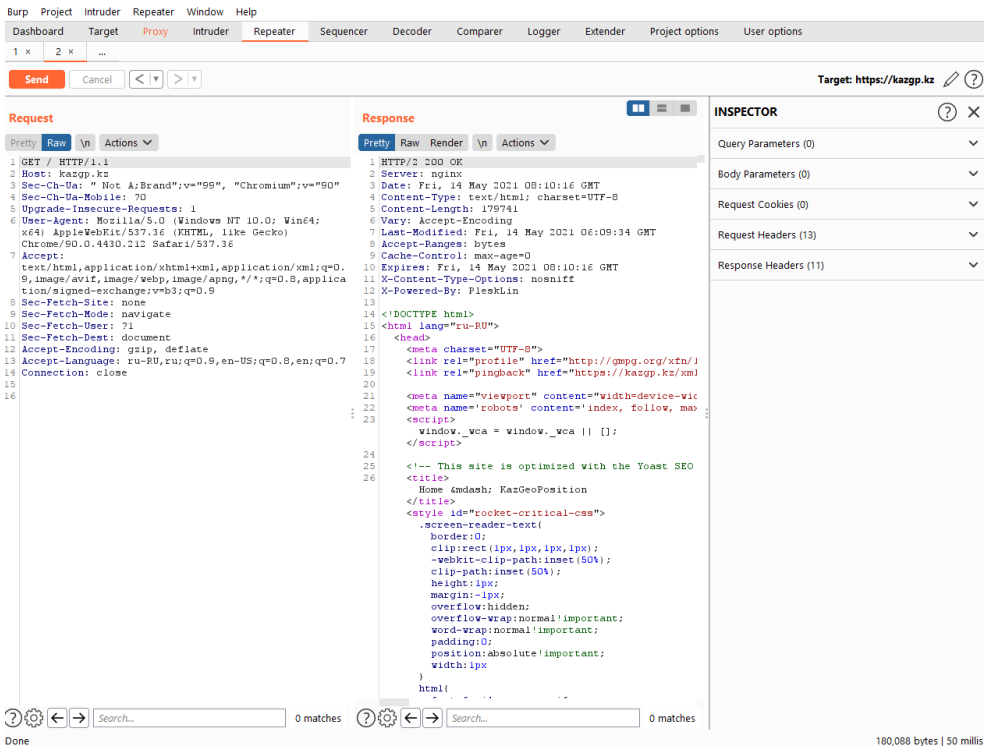


Рисунок 2.29 – Контроль доступа в burp (repeater)

По результатам работы повторяющихся запросов, не было выявлено проблем связанных с контролем доступа.

Фаззинг параметров. Проверки логики произведения веб-приложения или иначе объясняясь техника испытания программного обеспечения, зачастую самодействующая либо полуавтоматическая, заключающаяся в передаче приложению на вход неправильных, внезапных либо беспорядочных данных. объектом внимания представляется падения и зависания, нарушения внутренней логики и испытаний в коде приложения, утечки памяти, активизированные подобной информацией на входе.

Фаззинг представляется разновидностью проверочного испытания (англ. random testing), зачастую используемого на пробу проблем сохранности в программном обеспечении и компьютерных системах. В данном случае мы будем стараться достигнуть погрешности отказа в обслуживании. Как Burp Suite функционирует как Fuzzer? Burp Suite располагает интегрированный Fuzzer HTML, содержащий наименование «Burp Intruder». Дабы совершить вероятной атаку при помощи фаззинга, необходимо прибавить словарь в качестве перечня полезной нагрузки. Хотя Burp Suite Professional Edition доставляет юзерам право избрать определенные перечни нужной нагрузки, держащие преимущественно известные строчки fuzz в соответствии с типами атак. Читатели, наверное, не до конца понимают, как осуществляется фаззинг. Вот несколько несложных шагов: Во-первых, необходимо перехватить HTTP-запрос, тем самым юзер поделится им с Intruder. Как только пользователь сие сделает, будут обусловлены характеристики или «точки впрыска», где должно проложить фаззинг [8].

Далее подобает избрать тип атаки и перечень нужной нагрузки. Как только юзер запустит фаззинг-атаку, нажав на кнопку «Attack», пред его глазами явится экран с указанием всех вероятных уязвимостей. Таким образом, он сможет проверить их и найти самое слабенькое местечко в защите приложения. Фаззинг с при помощи интегрированных списков полезной нагрузки Burp сегодня читатели знают, что этакое фаззинг и как Intruder от Burp Suite помогает юзерам фаззировать веб-приложение. Подобает передвигаться дальше и понять, как перехватить какой-либо запрос, дабы фаззировать приложение, используя поставленный список полезной нагрузки burp. Фаззинг учетных данных для входа в систему Имена пользователей и пароли играют значительную роль в приложении, поэтому, ежели бы пользователь смог фаззировать их, он мгновенно б обошел фазу аутентификации. Фаззинг поля «Пароль» с поддержкою перечня паролей и перечня коротких слов. Необходимо открыть Burp Suite, дабы перехватить запрос, а впоследствии поделиться им с Intruder. Рассмотрим процесс фаззинга более подробно. Для начала необходимо обнаружить BurpSuite, перехватив запрос, пускаем его в Intruder

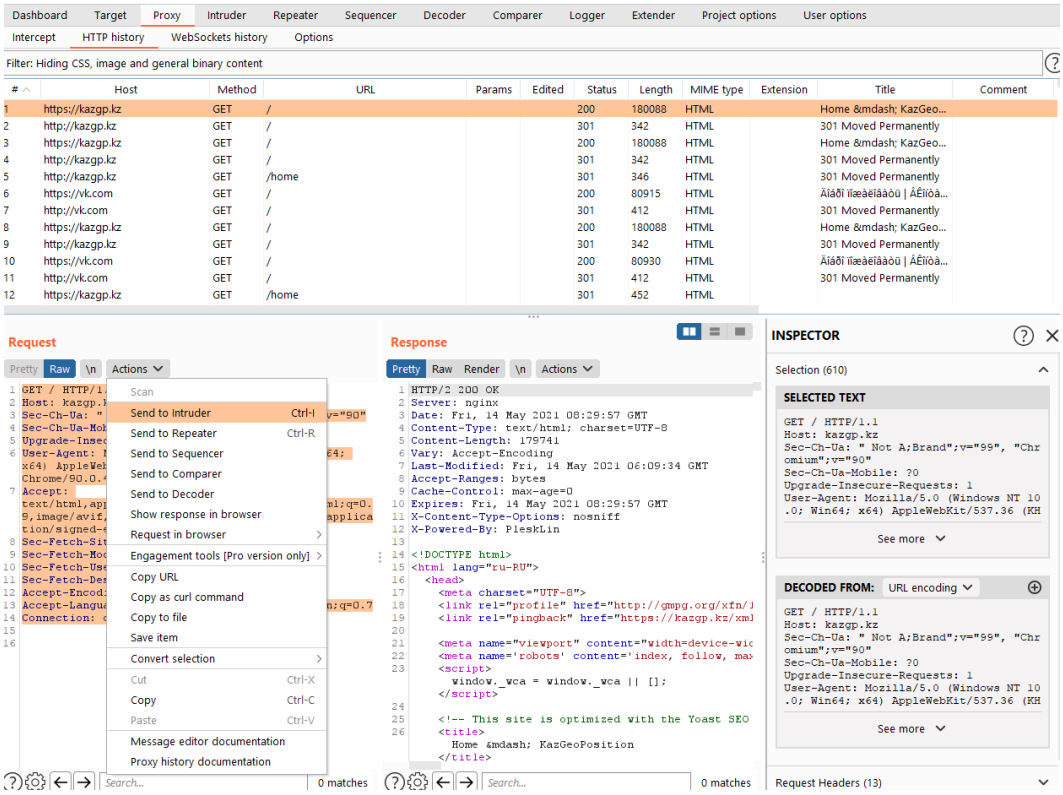


Рисунок 2.30 – Отправка запроса в Intruder

Теперь следует настроить расположение полезной нагрузки и нажать на кнопку «Add». Пользователь может выбрать тип атаки, который определяется тем, как полезная нагрузка будет атаковать точку впрыска. Расположение полезной нагрузки: (пароль пользователя). Тип атаки: «Sniper» (для одной полезной нагрузки).

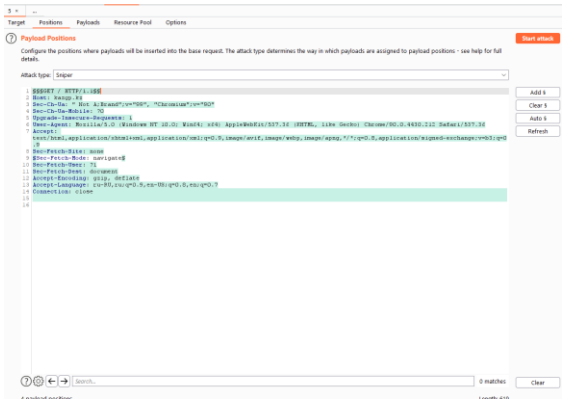


Рисунок 2.31 – Добавление полезной нагрузки

Пользователь выберет опцию полезной нагрузки, чтобы создать простой список полезной нагрузки для атаки.

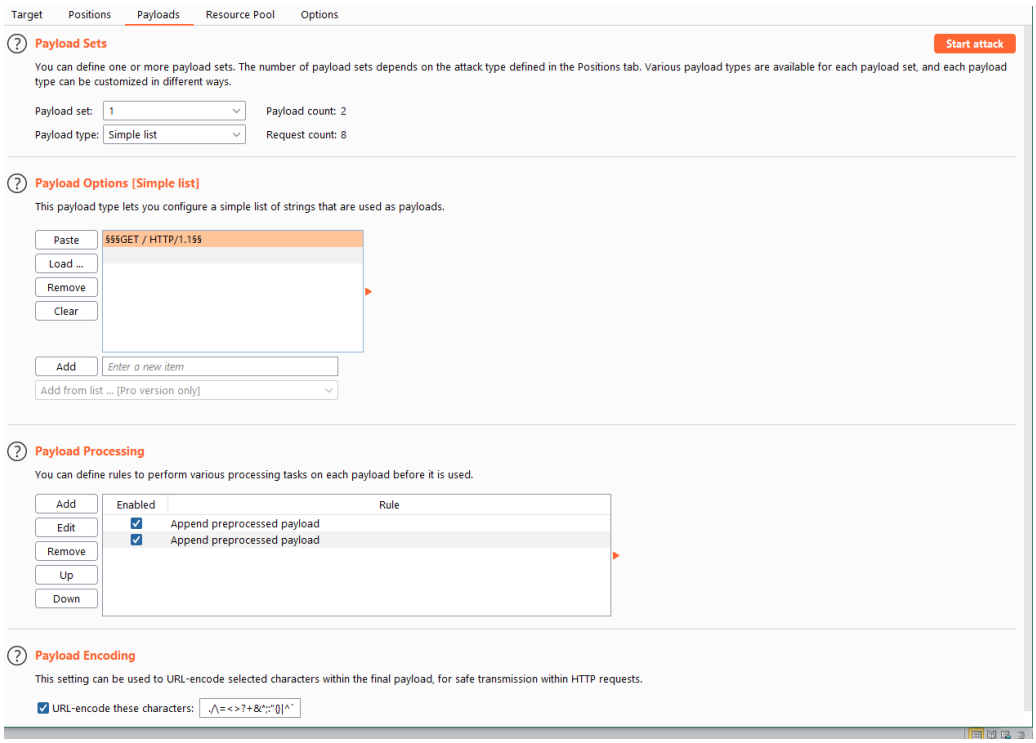


Рисунок 2.32 – Добавление полезной нагрузки в виде GET/HTTP/1.1

После инициирования процесса атаки, Burp Suite начнет атаку, отправив запросы на ввод правильного пароля для соответствующего имени пользователя. Теперь из заданного списка примененных строк пользователь дважды щелкнет на раздел «Length», чтобы отсортировать компоненты в порядке увеличения. Далее он выбирает тот компонент, который имеет наименьшую длину.

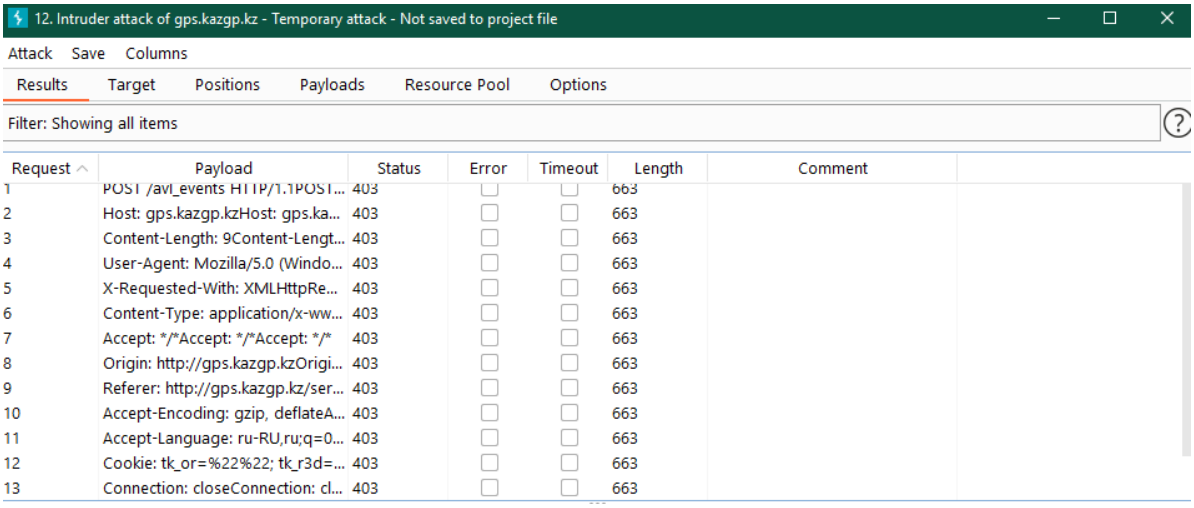


Рисунок 2.33 – Поиск наименьшей длины

Фаззинг поля «Имя пользователя» с помощью списка имен пользователей. Бывают случаи, когда есть общий пароль, но человек не знает,

сколько пользователей установили его для себя. Чтобы решить эту дилемму, burpsuite предлагает использовать еще один большой список полезной нагрузки, который содержит все распространённые имена пользователей. Нужно открыть Burpsuite и захватить запрос для входа в систему, а затем поделиться им с Intruder. Далее установить параметр «random» для точки впрыска. К примеру, в данном случае происходит подбор всех возможных пользователей с паролем «admin».

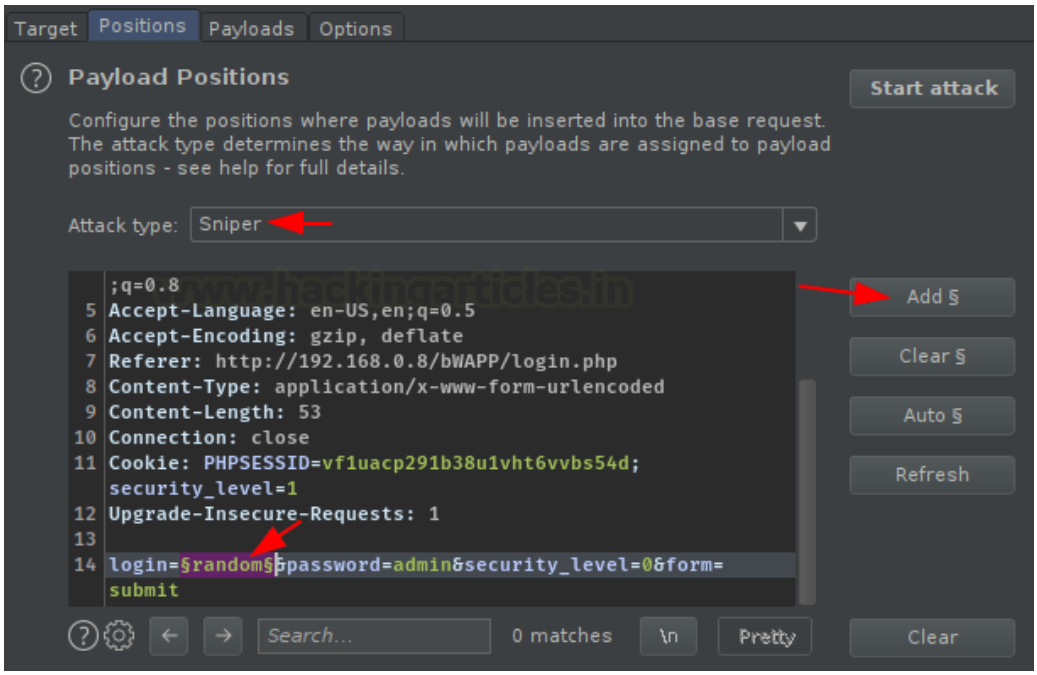


Рисунок 2.34 – Подбор самого короткого значения

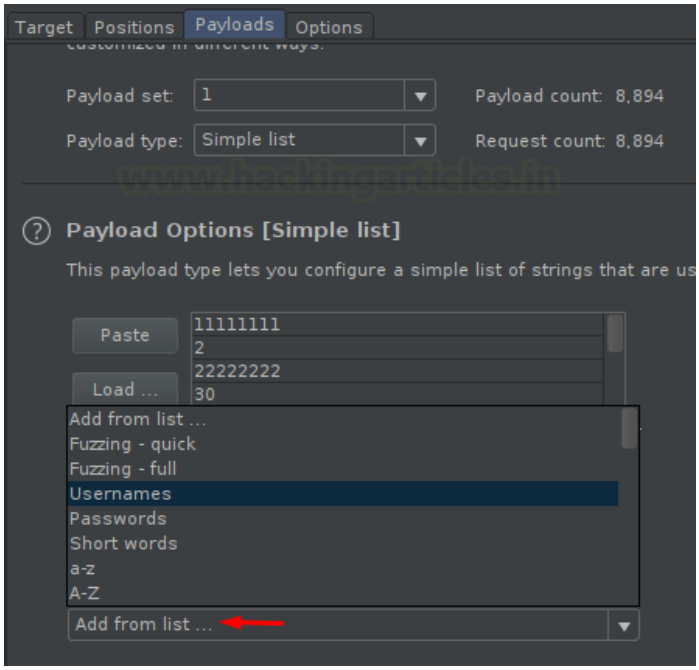


Рисунок 2.35 – Подбор всех возможных пользователей с паролем «admin»

После определения расположения полезной нагрузки пришло время выбрать список полезной нагрузки «Usernames» из программы.

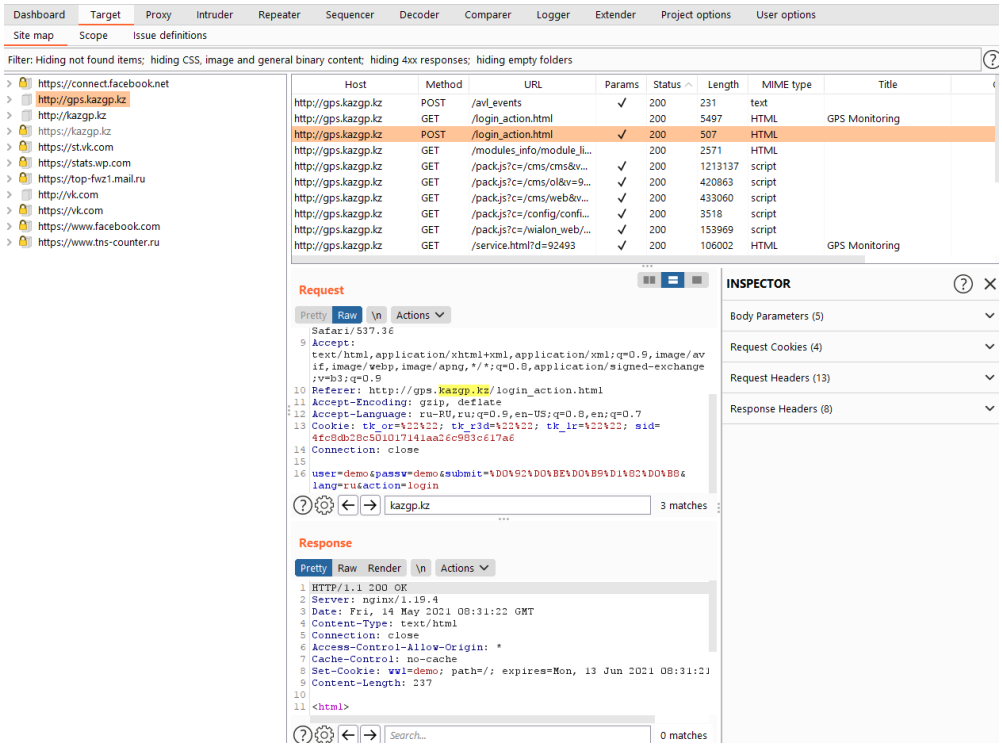


Рисунок 2.36 – Пример атаки man-in-the-middle

Как видно из скриншота 19 данные используемые для входа по адресу gps.kazgp.kz могут быть перехвачены посредством man-in-the-middle атаки. Что особенно актуально, так как многие пользователи пользуются мобильным приложением которые связывается с этой страницей. При условии зараженного или подконтрольного третьим лицам wifi публичного открытого модема, возможны полное взятие под контроль пользовательского аккаунта.

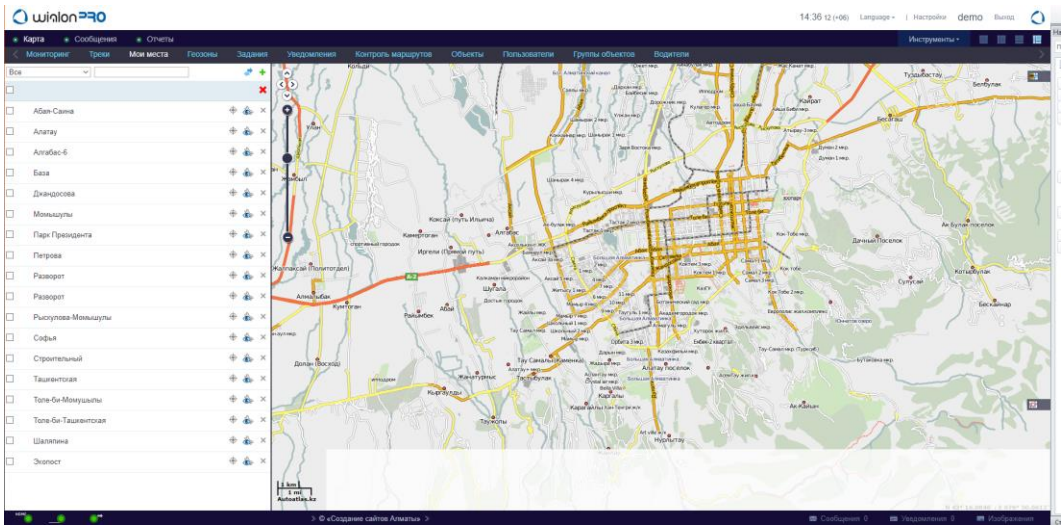


Рисунок 2.37 – Удачный вход под пользователем Demo по адресу gps.kazgp.kz

Проверки логики работы веб-приложения. В ходе анализа веб-приложения компании ТОО «КазГеоПозиция» были выявлена внутренняя структура веб-сайта, открывающая потенциальную уязвимость для внешнего воздействия. На данный момент попытки получить доступ к приватным папкам и определенным файлам невозможна, но при должном уровне подготовки и массированной атаке с участием зараженной группы машин (ddos) используя bruteforce и SQL инъекции в момент ошибки ответного реквеста, можно добиться перехвата персональных данных.

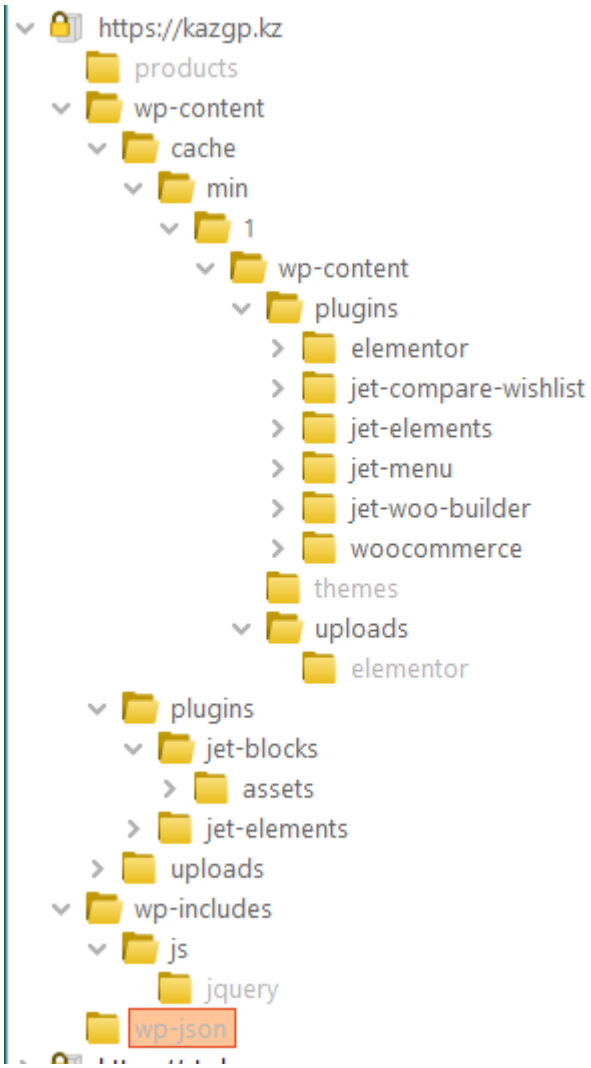


Рисунок 2.38 – Внутренняя структура веб-сайта компании ТОО «КазГеоПозиция»

2.3 Рекомендации к обеспечению безопасности для веб-приложения компании ТОО «КазГеоПозиция»

Прямой способ защиты приложений — межсетевой экран или брандмауэр. Для большего числа веб-приложений применяется прикладной сетевой экран Web Application Firewall (WAF). Если же мы говорим о бизнес-

приложениях, которые содержат базы коммерческих и персональных данных, то здесь требуется другой тип защиты — брандмауэр баз данных Database Firewall (DBF). Это позволяет защитить конфиденциальные данные на разных уровнях.

Применение специализированных решений по информационной безопасности позволяет обнаружить и предотвратить атаки на прикладном и сетевом уровне и реализовать комплекс мер, чтобы обеспечить доступность и непрерывность работы web-приложений за счёт защиты от различных классов атак.

Межсетевой экран автоматически обнаруживает и блокирует атаки на веб-приложения и определяет нелегитимных пользователей, пытающихся проникнуть в веб-приложение [8].

К основным мерам относятся:

- проверка данных на соответствие стандартам протоколов;
- контроль трафика на основе нейронных сетей;
- сигнатурный анализ;
- защита от SQL-инъекций;
- протекция от межсетевого скриптинга (XSS);
- контроль доступа к конфиденциальным данным.

Введение программно-аппаратных комплексов понижает риски несанкционированного доступа к критичной информации и эксплуатации уязвимостей системного ПО. Более того, наличие подготовленных заключений по информационной безопасности дозволяет снабдить законодательные условия по защите личных данных, а также банковских стандартов (СТО БР) и стандарта безопасности предоставленных индустрии платежных карт PCI DSS в вопросах предохранения веб-приложений. использование подготовленных систем защиты web-приложений дозволит вовремя раскрывать и предотвращать пробы несанкционированных действий злоумышленников как внутри организации, так и извне. Помимо раскрытия и блокировки атак для защиты данных в приложениях необходимо безостановочное наблюдение доступа к базам данных и расследование поведения юзеров и систем. Эти функции обеспечивают решения класса DAM (database activity monitoring). Рассмотрим детальнее принцип произведения этаких решений.

DAM и DBF — классы систем для защиты веб-приложений. Веб-приложения непосредственно сопряжены с СУБД, отчего атаки, нацеленные на них могут быть критичными. Обретая доступ к приложению, преступник может не только деактивировать его работу, но и завладеть значимой информацией, что грозит крупными экономическими и репутационными рисками. Как говорили выше, чтобы защитить бизнес-приложения от современных угроз, одного WAF и сигнатурных средств недостаточно. Потребуются специальные решения для обеспечения безопасности баз данных. Средства по обороне баз данных и веб-приложений относятся к классам Database firewall (DBF) / Database Activity.

Monitoring (DAM) – это аппаратно-программные комплексы для мониторинга, аудита и контроля доступа к информации и защиты от целевых атак на них. Есть решения, объединяющие в себе функциональные возможности по мониторингу, аудиту и защите от атак на базы данных. В качестве основных функций систем DAM/DBF выделим следующие:

- защита от внешних атак;
- выявление уязвимостей БД;
- обнаружение неучтенных конфиденциальных данных в базах приложений;
- блокировка неавторизованных или нетипичных запросов и ответов;
- проверка данных на обезличенность при передаче;
- тотальный контроль всех запросов к БД и настраиваемые политики безопасности;
- построение профилей пользователей и выявление подозрительной активности;
- автоматическое сканирование БД на наличие конфиденциальной информации;
- расследование инцидентов безопасности;
- предотвращение утечек данных.

Гарантия информационной безопасности веб-приложений с помощью подготовленных систем — это групповая задача бизнеса и возможность нивелировать риски.

Уровень контроля доступа — это своеобразный шлюз, который юзеры проходят для аутентификации и авторизации в приложении. вдобавок системы контроля могут быть развернуты несколькими способами.

Учетные данные клиента могут содержаться как в определенной базе данных, так и могут применяться локальные решения, к примеру, сетевой протокол прикладного значения для передачи запросов (LDAP). Приложения вдобавок могут законнекчиваться к шлюзам единого входа (Single Sign-on, SSO изнутри и снаружи, как в случае с услугами по федерации доступа).

Защита транспортного значения Транспортный уровень приложения отвечает за обеспечение шифрования, потому сетевые пакеты протекают сквозь ненадежные сети в интернете либо взаимодействуют с ненадежными Wi-Fi точками доступа. Зашифрованная инкапсуляция пакетов информации случается на пути к серверу веб-приложения и уже в зашифрованном варианте прибывает на сторону клиента. Система защиты транспортного уровня (TLS) вдобавок гарантирует, что злоумышленники не подделали сведения в пути и проверяет приложение на существование соответствующего сертификата домена из доверенного центра сертификации. Данный уровень охватывает в себя глобальный протокол HTTPS, TLS и несовременный протокол шифрования SSL [8].

Уровень поддержания доменного имени системы. DNS — это своеобразная адресная книга интернета. Это всемирно известная служба,

основанная на согласованных стандартах. Клиенты, подключающиеся к приложению, в значительной степени зависят от его функциональности и испытанного DNS. Ежели DNS скомпрометирован или, что хуже, был подделан – это может повлечь серьезные последствия для безопасности приложения.

Так как сами приложения могут нуждаться в подключении к иным службам извне, они безгранично зависят от правильно показанного и действующего DNS. Данный уровень охватывает все DNS-серверы, требуемые клиенту и приложению, а вдобавок подходящие регистры этих доменов.

Сетевой уровень. Обычно подключение юзеров к приложениям случается с помощью интернета. А наиболее распространенным протоколом для веб-трафика, будь то публичные вебсайты или вызовы прикладного-программного интерфейса от машины к машине, представляется HTTP. А более защищенные приложения применяют в данных целях HTTPSшифрование.

Сетевой уровень еще охватывает в себя все сетевые сервисы приложений. Сюда причисляются интернет-провайдеры (ISP), соединения «последней мили», данные теми же провайдерами ради устройств клиентов, и протоколы маршрутизации.

Клиенты приложений. Большинство приложений могут работать как серверы в физической или виртуальной среде, но им необходим клиентский интерфейс для передачи данных к и от пользователей. Сегодня весьма непросто обнаружить приложение, которое приносит пользу и при этом приходится автономным, не присоединенным к интернету (к примеру, «Блокнот» в Windows). Преимущественно известным клиентом веб-приложения прибывает веб-браузер.

Прогрессивные веб-браузеры шагнули далеко вперед в сопоставлении с Mosaic, браузером выданным Национальным центром суперкомпьютерных технологий в 1993 году. Нынче почти каждое веб-приложение готово к тому, что веб-клиенты будут запускать действующие скрипты Java или Flash. Ныне все плотнее сами приложения употребляют Active Scripting внутри самого браузера, что приводит к тому, что веб-клиент и интернет-браузер взаимодействуют посредством HTTP, передавая друг другу сведения и команды для обработки. Что, в свою очередь, приводит к появлению новоиспеченных опасностей IT-безопасности, и, как следствие, происхождению свежих условий к решениям по защите от них. Системы по разбору кода приложений вдобавок зачастую творят новоизобретенный уровень задач безопасности, поскольку временами случается нелегко установить важность информации их анализа, а они сами могут не различать код от данных, введенных юзером [8].

К клиентам приложений вдобавок причисляются мобильные приложения, которые зачастую презентуют собой первоначально сконфигурированные интерфейсы веб-браузера для уже существующих веб-приложений. Приложения и IoT-устройства вдобавок могут независимо перекидываться информацией с другими приложениями посредством API либо при помощи утверждения сочетания с сервисными службами.

3. Технико-экономическое обоснование проекта

3.1 Цели и задачи, решаемые в экономической части

В следствии выполнения предоставленной дипломной работы будет разработана децентрализованная концепция (сайт) на основе реализации смарт-контрактов в виде кошелька с транзакциями и блоками.

Для разработки сайта потребуется участие группы специалистов, состоящей из проектного менеджера и программиста-разработчика. Проектный руководитель будет нести ответственность за распределение шагов работы и следование графиков исполнения. Обязанностями программиста-разработчика представлены в виде поиска верного решения, разработка и испытание программного продукта.

Технико-экономическое доказательство дипломного проекта включает в себя последующие пункты:

– определение объёма и трудоёмкости разработки программного обеспечения;

– расчёт затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ;

– смета затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ;

– расчёт возможной цены разрабатываемого ВЕБ-ПРИЛОЖЕНИЯ;

– расчёт эксплуатационных затрат при использовании ВЕБ-ПРИЛОЖЕНИЯ;

– расчёт результатов от создания и использования ВЕБ-ПРИЛОЖЕНИЯ;

– расчёт основных показателей экономической эффективности и срока окупаемости проекта;

– выводы технико-экономической части.

3.2 Определение объёма и трудоёмкости разработки ПО

Основой для расчёта сметы затрат на разработку программного продукта являются объём и трудоёмкость процесса разработки. Для их определения процесс разработки был разделён на несколько этапов, которые представлены в таблице 3.1.

Таблица 3.1 – Этапы разработки ВЕБ-ПРИЛОЖЕНИЯ и оценка их трудоёмкости

Описание работы	Трудоёмкость, чел.× ч.
Постановка задачи проекта	5
Разработка и утверждение технического задания на разработку ВЕБ-ПРИЛОЖЕНИЯ	10

Продолжение таблицы 3.1

Веб-приложения и изучение литературы	26
Разработка программно-аппаратные части приложения	70
Разработка клиентской части приложения	35
Тестирование и устранение Веб-приложения неполадок программы	25
Внедрение готового программного продукта	18
Итог	189

Продолжительность рабочего дня равна восьми часам, однако у нас есть час на обед, итог 7 часов рабочего дня. Таким образом, для создания разрабатываемого программного обеспечения потребуется 189 / 7 = 27 рабочих дней.

3.3 Расчёт затрат на разработку программного продукта

Для нахождения затрат на разработку ПП необходимо вычислить следующие статьи расходов:

– затраты на материалы и специальные программные средства, необходимые для разработки ВЕБ-ПРИЛОЖЕНИЯ;
– затраты на электроэнергию;
– затраты на оплату труда специалистов;
– затраты на социальный налог;
– амортизация основных фондов.

3.3.1 Расчёт затрат на материалы и специальные программные средства. Затраты на материалы для разработки ВЕБ-ПРИЛОЖЕНИЯ представлены в таблице 3.2.

Таблица 3.2 – Затраты на материалы

Наименование материального ресурса	Марка	Единица измерения	Количество	Цена за единицу, тг	Общая сумма, тг
Блокнот	«Vip Line»	шт.	1	500	500
Ручка	«Pilot»	шт.	2	330	660
Итого					1160

При разработке будет использоваться ноутбук ACER NIRO 5, мощности которого достаточно для выполнения поставленной задачи. Ноутбук содержит установленную операционную систему Windows 10; программное обеспечение, необходимое для разработки ВЕБ-ПРИЛОЖЕНИЯ (Microsoft

Visual Code), устанавливается бесплатно, поэтому производить дополнительные затраты на ОС и ВЕБ-ПРИЛОЖЕНИЯ не требуется. Затраты на специальные программные средства представлены в таблице 3.3 [9].

Таблица 3.3 – Затраты на специальные программные средства

Наименование программного средства	Модель/ Название	Единица измерения	Коли-чество	Цена за единицу, тг	Общая сумма, тг
Ноутбук	ACER NITRO 5	шт.	1	299 990	299 990
Мышь компьютерная	Steelseries KANA	шт.	1	8 590	8 590
Батареи 1,5 V	«Сони»	шт.	2	220	440
Интернет	ID NET	месяц	1	4 990	4 990
Операционная система	Windows 10	год	1	27 315	27 315
Среда разработки ВЕБ-ПРИЛОЖЕНИЯ	Microsoft Visual Code 2020	шт.	1	–	–
Итого					341 325

Общая сумма затрат на материалы и специальные программные средства ($З_м$) определяется Веб-приложения формуле:

$$З_м = \sum_{i=1}^{n} P_i * Ц_i, \quad (3.1)$$

где P_i – расход i-го вида материального ресурса, натуральные единицы;

$Ц_i$ – цена за единицу i-го вида материального ресурса, тг;

i – вид материального ресурса;

n – количество видов материальных ресурсов.

$$З_м = (1160 + 341\,325)\text{ тг} = 342\,485\text{тг}$$

Таким образом, для разработки ВЕБ-ПРИЛОЖЕНИЯ понадобится материалов и специальных программных средств на сумму тенге.

3.3.2 Расчёт затрат на электроэнергию. Так как для разработки ВЕБ-ПРИЛОЖЕНИЯ также потребуется электроэнергия, необходимо произвести расчёт затрат на электроэнергию, которая будет потрачена в течение всего времени разработки ВЕБ-ПРИЛОЖЕНИЯ (224 часа). Так как принтер не будет использоваться Веб-постоянно, для него расчёт будет произведён для периода в 24 часа.

Тариф на электоэнерию для юридических лиц города Алматы с 1 января 2020 года составляет 19,17 тенге за 1 кВт×ч с учётом НДС (согласно данным, опубликованным на официальном сайте ТОО «АлматыЭнергоСбыт»). Общая сумма затрат на электроэнергию ($З_Э$) рассчитывается Веб-приложения формуле:

$$З_Э = \sum_{i=1}^{n} M_i \times K_i \times T_i \times Ц\ , \qquad (3.2)$$

где M_i – паспортная мощность i-го электрооборудования, кВт;

K_i – коэффициент использования мощности i-го электрооборудования (K_i = 0,7..0,9);

T_i – время работы i-го оборудования за весь период разработки ВЕБ-ПРИЛОЖЕНИЯ, ч.;

Ц – цена электроэнергии (тариф), тг/кВт×ч.;

i – вид электрооборудования;

n – количество электрооборудования.

Результаты расчётов представлены в таблице 3.4.

Таблица 3.4 – Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования, ч	Цена электроэнергии, тг/кВт·ч	Сумма, тг
Сплит	1	1,2	150	19,17	2400
Ноутбук	0,7	0,9	175	19,17	2085
Освещение	0,3	0,7	175	19,17	705
Итого					4490

Согласно выполненным расчётам, затраты на электроэнергию составляют 2790 тенге.

3.3.3 Расчёт затрат на оплату труда специалистов. Фонд оплаты труда специалистов ($З_{ФОТ}$) можно вычислить Веб-приложения формуле:

$$З_{ФОТ} = З_{тр} + З_{доп.}, \qquad (3.3)$$

где $З_{тр}$ – основная заработная плата специалистов, тг;

$З_{доп.}$ – дополнительная заработная плата специалистов, тг.

Основная заработная плата рассчитывается Веб-приложения следующей формуле:

$$З_{тр} = \sum_{i=1}^{n} ЧС_i * T_i \quad (3.4)$$

где $ЧС_i$ – часовая ставка i-го работника, тг;

T_i – трудоёмкость разработки ВЕБ-ПРИЛОЖЕНИЯ, чел.×ч.;

i – категория сотрудника;

n – количество сотрудников.

Дополнительная заработная плата составляет 10% от основной зар. платы и находится Веб-приложения формуле:

$$З_{доп.} = З_{тр} * 0{,}1. \quad (3.5)$$

Часовая ставка сотрудника будет рассчитана Веб-приложения формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (3.6)$$

где $ЗП_i$ – месячная заработная плата i-го сотрудника, тг;

$ФРВ_i$ – месячный фонд рабочего времени i-го сотрудника, час.

Месячный фонд рабочего времени сотрудника определяется Веб-приложения формуле:

$$Ч_м = N_м * Ч_{рд}, \quad (3.7)$$

где $Ч_м$ – количество рабочих часов сотрудника за месяц;

$N_м$– количество рабочих дней за месяц;

$Ч_{рд}$ – количество рабочих часов в день.

Вычислим месячный фонд каждого сотрудника Веб-приложения формуле (3.7):

$$Ч_м = 28×7 = 196 \text{ ч.}$$

В разработке ВЕБ-ПРИЛОЖЕНИЯ будут задействованы два сотрудника: проектный менеджера и программист-разработчик. Средняя заработная плата проектного менеджера в Казахстане в 2020 году составляет 250 000 тенге, а программиста-разработчика – 270 000 тенге. Вычислим часовую ставку каждого сотрудника Веб-приложения формуле (3.6):

$$ЧС_{проект.менеджер} = \frac{250\,000}{196} = 1275{,}51 \text{ тг/ч}$$

$$ЧС_{прогр.-разр.} = \frac{270\,000}{196} = 1377{,}55 \text{ тг/ч}$$

Чтобы определить трудоёмкость разработки для каждого сотрудника, восВеб-приложенияльзуемся данными из таблицы 3.1. От проектного менеджера Веб-приложениятребуется участие в постановке задачи проекта, разработке и утверждении технического задания, разработке клиентской части приложения, внедрении готового программного продукта:

$$Т_{проект.менеджер} = 5 + 10 + 18 = 33 \text{ чел.}\times \text{ч}$$

Таким образом, трудоёмкость руководителя проекта составит 33 чел.×ч.

Программист-разработчик будет занят разработкой и утверждением технического задания, поиском и изучением сопутствующей литературы и подобных программ, разработкой клиентской и серверной частей приложения, тестированием, отладкой и устранением неполадок программы и, наконец, внедрением готового программного продукта:

$$Т_{прогр.-разр.} = 5 + 10 + 26 + 72 + 35 + 25 = 173 \text{ чел.}\times \text{ч}$$

Трудоёмкость программиста-разработчика составит 173×часов.

Согласно формуле (3.4), основная заработная плата специалистов составит:

для проектного менеджера:

$$З_{тр_{проект.менеджер}} = 1275{,}51 \text{ тг/ч} * 33 \text{ чел.}\times \text{ч} = 42\,091 \text{ тг}$$

Для программиста-разработчика:

$$З_{тр_{прогр.-разр.}} = 1377{,}55 \text{ тг/ч} * 173 \text{ чел.}\times \text{ч} = 238\,316 \text{ тг}$$

Общая сумма основной заработной платы труда специалистов равна:

$$З_{тр} = З_{тр_{проект.менеджер}} + З_{тр_{прогр.-разр.}} = 42\,091 + 238\,316 = 280\,407 \text{тг}$$

Затраты на основную заработную плату труда специалистов показаны в таблице 3.5.

Таблица 3.5 – Затраты на основную заработную плату труда специалистов

Специалист	Трудоёмкость разработки ВЕБ-ПРИЛОЖЕНИЯ, чел.×ч.	Часовая ставка специалиста, тг/ч	Сумма осн. з/п, тг

Продолжение таблицы 3.5

Проектный менеджер	33	1274,28	42 091
Программист-разработчик	173	1428,57	238 316
Итого			280 407

Найдём дополнительную заработную плату Веб-приложения формуле (4.5):

$$З_{доп.} = 280\ 407 * 0{,}1 = 28\ 040 \text{ тг.}$$

Таким образом, фонд оплаты труда специалистов согласно формуле (3.3) составит:

$$З_{ФОТ} = 280\ 407 + 28\ 040 = 308\ 447 \text{ тг.}$$

3.3.4 Расчёт затрат на налоги. Для подсчёта налогов, уплаченных юридическим лицом, необходимо вычислить размер обязательных пенсионных взносов (ОПВ), социальные отчисления (СО), отчисления на обязательное социальное медицинское страхование (ОСМС) и социальный налог (СН). Расчёты представлены в таблице 3.6. Итоговые налоговые отчисления составляют 10,46% от фонда оплаты труда (согласно Налоговому Кодексу РК).

Пенсионные отчисления составляют 10% от фонда оплаты труда. Таким образом,

$$ОПВ = З_{ФОТ} * 0,1 = 308\ 447 * 0{,}1 = 30\ 847{,}7 \text{ тг.}$$

Таблица 3.6 – Расчёт затрат на налоги

Налог	Фонд оплаты труда ($З_{ФОТ}$), тг	Формула для вычисления	Сумма, тг
СО	308 447	($З_{ФОТ}$ – ОПВ)*3,5%	9 715,97
ОСМС		$З_{ФОТ}$*2%	6 168,94
СН		($З_{ФОТ}$ – ОПВ – ОСМС)*9,5% – СО	16 069,91
Итого			**31 954**

3.3.5 Амортизация основных фондов (ОФ). Общая сумма амортизационных отчислений определяется Веб-приложения формуле:

$$З_А = \sum_{i=0}^{n} \frac{Ф_i * Н_{A_i} * N_i}{100 * 12 * n}, \quad (3.10)$$

где $Ф_i$ – стоимость i-го ОФ, тг;

$Н_{A_i}$ – годовая норма амортизации i-го ОФ, %;

N_i – время работы i-го ОФ за весь период разработки ВЕБ-ПРИЛОЖЕНИЯ, дней;

n – количество рабочих дней в месяце;

i – вид ОФ.

Основными фондами в данном дипломном проекте выступают ноутбук и принтер.

Годовая норма амортизации ОФ вычисляется Веб-приложения формуле:

$$H_{Ai} = \frac{100}{T_{Ni}}, \tag{3.11}$$

где $Т_{Ni}$– возможный срок использования i-го ОФ, год;

Норма амортизации для ноутбука составит:

$$Н_{А_{ноут.}} = \frac{100}{4} = 25\%.$$

Норма амортизации для принтера составит:

$$Н_{А_{прин.}} = \frac{100}{5} = 20\%.$$

Ноутбук за весь период разработки ВЕБ-ПРИЛОЖЕНИЯ понадобится для поиска и изучения сопутствующих референсов и литературы Веб-приложения программам, разработки клиентской и объектно-ориентированной части, тестирования, отладки и устранения неполадок программы. Согласно таблице 3.1, время работы в днях составит:

$$N_{ноут.} = \frac{(26 + 70 + 35 + 25)\ ч}{7\ ч} = \frac{156}{7} = 22\ дня.$$

Количество рабочих дней в месяце примем равным 21 (n=21).

Найдём общую сумму амортизационных отчислений Веб-приложения формуле (3.10):

$$З_А = \frac{299\,990 * 25 * 22}{100 * 12 * 22} = 6249{,}7\ тг.$$

Таблица 3.7 – Амортизация основных фондов

Наименование оборудования	Стоимость оборудования, тг	Годовая норма амортизации, %	Время работы оборудования, дни	Сумма, тг
Ноутбук	299 990	25	22	6249,7
Итого				6249,7

3.4 Смета затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ

На основе произведённых расчётов оформляется смета затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ, которая представлена в таблице 3.7. Соответствующая диаграмма продемонстрирована на рисунке 3.1.

Таблица 3.8– Смета затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ

Наименование статей затрат	Сумма, тг	% от общей суммы
Затраты на материалы	1210	0,18
Затраты на специальные программные средства	341 325	51,27
Затраты на электроэнергию	4490	0,47
Затраты на оплату труда специалистов	280 407	47,86
Затраты на социальный налог	31 954	9,5
Амортизация основных фондов	6249,7	0,79
Итого	665 635,7	100

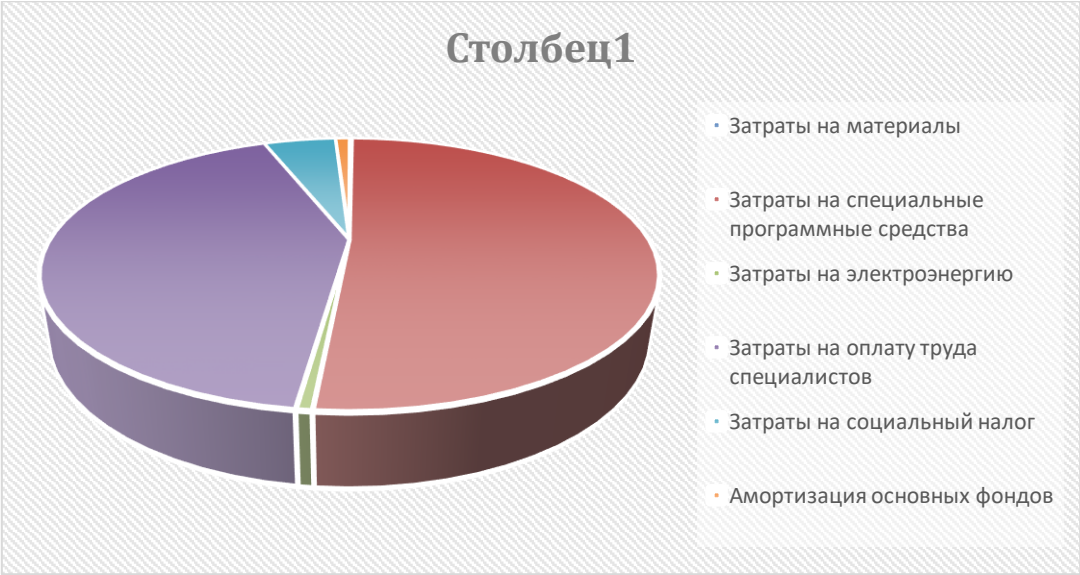


Рисунок 3.1 – Смета затрат на разработку ВЕБ-ПРИЛОЖЕНИЯ

3.5 Расчёт возможной (ориентировочной) цены ВЕБ-ПРИЛОЖЕНИЯ

Стоимость программного обеспечения ($Ц_Д$) формируется в зависимости от затрат на его разработку и уровня рентабельности ВЕБ-ПРИЛОЖЕНИЯ и вычисляется Веб-приложения формуле:

$$Ц_Д = З_{нир}\left(1 + \frac{P}{100}\right), \quad (3.12)$$

где $З_{нир}$ – затраты на разработку программного обеспечения, тг;

Р – средний уровень рентабельности ВЕБ-ПРИЛОЖЕНИЯ, (%) (принят равным 25%).

$$Ц_Д = 665\,635{,}7\left(1 + \frac{25}{100}\right) = 832\,044.6 \text{ тг.}$$

Согласно Налоговому Кодексу РК, оплата налога на добавленную стоимость (НДС) является обязательной. Ставка НДС в 2020 году составляет 12% для ВЕБ-ПРИЛОЖЕНИЯ. Определить реализацию ВЕБ-ПРИЛОЖЕНИЯ с учётом НДС можно Веб-приложения формуле:

$$Ц_р = Ц_д(1 + НДС), \quad (3.13)$$

$$Ц_р = 832\,044.6(1 + 0{,}12) = 931\,889.9 \text{ тг.}$$

получившуюся суму можно округлить до 931 890 тенге.

3.6 Расчёт эксплуатационных затрат при использовании ВЕБ-ПРИЛОЖЕНИЯ

Годовые эксплуатационные текущие затраты в условиях функционирования ВЕБ-ПРИЛОЖЕНИЯ (C) рассчитываются по формуле:

$$С = М + ЗП + А + ОТ + НР = Ц_д(1 + НДС), \quad (3.14)$$

где М – годовые материальные затраты на сопровождение программного продукта, тг;

ЗП – годовые затраты на оплату труда специалиста (программиста-разработчика), тг;

А – затраты на амортизацию, тг;

ОТ – отчисления на социальный налог, тг;

НР – накладные расходы, тг.

Затраты на материалы в условиях функционирования ВЕБ-ПРИЛОЖЕНИЯ не ожидается, соответственно $М = 0$.

Годовые затраты на заработную плату сотрудников определяются Веб-приложения формуле:

$$ЗП = \frac{О_с \cdot Ч_с \cdot 12}{Ф_{р.в.}} \cdot t_{общ} \cdot 12 \cdot (1 + К_Д), \quad (3.15)$$

где O_c – оклад специалиста, тг/мес.;

$Ч_c$ – численность специалистов, участвующих в процессе, чел.;

$Ф_{р.в.}$ – годовой фонд рабочего времени, час;

$t_{общ}$ – трудоёмкость решения задач в условиях функционирования ВЕБ-ПРИЛОЖЕНИЯ в месяц, час;

$K_д$ – коэффициент дополнительной заработной платы.

Трудоёмкость решения задач вычисляется следующим образом:

$$t_{общ} = \sum_{k=1}^{n} t_k \cdot K_k, \quad (3.16)$$

где t_k – затраты времени на решение k-й задачи, час;

K_k – количество решаемых k-х задач в месяц, ед.

В ходе эксплуатации программы предполагается возникновение программных ошибок. На поиск и исправление одной ошибки в среднем уходит около полутора часа. В течение месяца эксплуатации ВЕБ-ПРИЛОЖЕНИЯ может возникать до 7 ошибок. помимо этого, ожидается внесение изменений во фронт-енд часть, а также доработка компонентов Веб-приложения требованию заказчика. Всего компонентов в системе 4, на доработку одного уходит в среднем около 5 часов.

$$t_{общ} = \ 1{,}5 \cdot 9 + 5 \cdot 4 = 13{,}5 + 2 = 30{,}5 \text{ часов.}$$

Годовой фонд рабочего времени равен 1968 часам (разработка клиентской и объектно-ориентированной части 4.1). Таким образом,

$$ЗП = \frac{270\,000 \cdot 1 \cdot 12}{1968} \cdot 30{,}5 \cdot 12 \cdot (1 + 0{,}1) = 662\,817 \text{ тенге.}$$

Амортизационные отчисления выражаются в процентах к балансовой стоимости оборудования и вычисляются Веб-приложения формуле:

$$А = \sum_{i=1}^{n} С_{обор._i} \cdot Н_{а_i}, \quad (3.17)$$

где $С_{обор._i}$– первоначальная стоимость оборудования;

$Н_{а_i}$– норма амортизации i-го оборудования;

i – вид оборудования;

n – количество видов оборудования.

Затраты на амортизацию составят:

$$А = 299\,990 * 0{,}25 = \ 74\,997{,}5 \text{ тг.}$$

Расчёты отчислений на социальный налог и накладные расходы производятся так же, как и при разработке ВЕБ-ПРИЛОЖЕНИЯ.

$$ОТ = ЗП * 10{,}46\% = 662\,817 \ * 0{,}1046 = 69\,330 \text{ тенге.}$$
$$НР = ЗП \ \cdot \frac{Н_{нр}}{100} = 662\,817 * 0{,}7 = 463\,971 \text{ тенге.}$$

Эксплуатационные затраты показаны в таблице 3.7.

Таблица 3.9 – Эксплуатационные затраты

Вид затраты	Сумма, тг	В процентах от общей суммы, %
Материальные затраты (М)	0	0
Заработная плата (ЗП)	662 817	52,11
Амортизационные отчисления (А)	74 997,5	5,89
Социальные отчисления и налоги (ОТ)	69 330	5,45
Накладные расходы (НР)	463 971	36,48
Итого	1 271 116	100

Годовые эксплуатационные затраты в условиях функционирования ВЕБ-ПРИЛОЖЕНИЯ составят 1 271 116 тенге.

3.7 Расчёт результатов без использования ВЕБ-ПРИЛОЖЕНИЯ

Разрабатываемое ВЕБ-ПРИЛОЖЕНИЕ обеспечит систему, где все ваши данные будут хранится в безопасности и создаст условия выполнения транзакций с выводом кошелька. Больше не придется использовать посредников(3-лиц) таких как банки, органы, так как вся информация будет в электронном виде

Для анализа экономии в результате использования разрабатываемого ВЕБ-ПРИЛОЖЕНИЯ необходимо сравнить эксплуатационные расходы с внедрением ВЕБ-ПРИЛОЖЕНИЯ и без [9].

Статьи затрат при использовании ВЕБ-ПРИЛОЖЕНИЯ включают в себя:

– заработная плата специалиста, который осуществляет поддержку и сопровождение системы;

– износ оборудования;

– накладные расходы.

Использование ВЕБ-ПРИЛОЖЕНИЯ не предполагает материальных расходов.

В организации имеются три аккаунт менеджера, заработная плата которых 120 000 тенге в месяц. Основной задачей каждого аккаунт менеджера является коммуникация с клиентом, и создания условий для работы.

Годовые затраты на оплату труда двух администраторов рассчитываются Веб-приложения формуле (3.15):

$$ЗП = \frac{130\,000 \cdot 3 \cdot 12}{1100} \cdot 24 \cdot 12 \cdot (1 + 0{,}1) = 1\,347\,840 \text{тенге.}$$

Социальные отчисления и налоги:

$$ОТ = 1\,347\,840 * 10{,}46\% = 140\,984 \text{ тенге.}$$

Для ведения записей аккаунт менеджеру потребуются блокнот (500тг за экземпляр, около 3 экземпляров в месяц, итого ~1500 тенге), канцелярских принадлежностей на сумму 5 000 тенге. Итого затрат на материалы и оборудование – 6500

Затраты на амортизацию составят:

$$А = 6500 * 0{,}25 = 1625 \text{ тенге}$$

Накладные расходы составят:

$$НР = ЗП \cdot \frac{Н_{нр}}{100} = 1\,347\,840 * 0{,}7 = 943\,488 \text{ тенге.}$$

Сравним следующие статьи затрат без и с применением разрабатываемого ВЕБ-ПРИЛОЖЕНИЯ: заработная плата специалистов, расход на материалы и оборудование, износ оборудования и накладные расходы в таблице 3.8:

Таблица 3.10 – Сравнение статей затрат с использованием ВЕБ-ПРИЛОЖЕНИЯ и без

Статья затрат	Без использования ВЕБ-ПРИЛОЖЕНИЯ, тг	С использованием ВЕБ-ПРИЛОЖЕНИЯ, тг
Материальные затраты (М)	130 000	0
Заработная плата (ЗП)	1 347 840	662 817
Амортизационные отчисления (А)	1625	62 747
Социальные отчисления и налоги (ОТ)	140 984	69 330
Накладные расходы (НР)	943 488	463 971
Итого	2 563 937	1 271 116

Вычислим ожидаемую условно-годовую экономию Веб-приложения формуле:

$$Э_{уг} = С_1 - С_2 + \sum Э_i, \quad (3.18)$$

где $Э_{уг}$ – величина экономии, тг;

$С_1$ и $С_2$ – показатели затрат Веб-приложения базовому и внедряемому вариантам, тг:

$\Sigma Эi$ – ожидаемый дополнительный эффект от различных факторов, тг.

Таким образом, ожидаемая условно-годовая экономия составит 2 563 937-1 271 116=1 292 822 тенге.

3.8 Расчёт основных показателей экономической эффективности

Эффективность разрабатываемого ВЕБ-ПРИЛОЖЕНИЯ формируется за счёт экономии в сравнении с предыдущим периодом работы без его использования.

Для расчёта величины ожидаемого годового экономического эффекта от внедрения ВЕБ-ПРИЛОЖЕНИЯ используется формула:

$$Э_г = Э_{уг} - К * Е_Н, \quad (3.19)$$

где $Э_г$ – ожидаемый годовой экономический эффект, тг:

$Э_{уг}$ – ожидаемая условно-годовая экономия, тг;

К – капитальные вложения, тг;

$Е_Н$ – нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений вычисляется Веб-приложения формуле:

$$Е_Н = \frac{1}{Т_Н}, \quad (4.20)$$

где $Т_Н$ – нормативный срок окупаемости капитальных вложений, лет. Для ВЕБ-ПРИЛОЖЕНИЯ примем равным четырём годам ($Т_Н = 4$).

Следовательно,

$$Е_Н = \frac{1}{4} = 0{,}25.$$

Ожидаемый годовой экономический эффект согласно формуле (3.19) равен:

$$Э_{уг} = 931\,890 \quad - 1\,292\,822 * 0{,}25 = \ 608685 \text{ тенге.}$$

Расчетный коэффициент экономической эффективности капитальных вложений находится Веб-приложения формуле:

$$E_р = \frac{Э_г}{К}, \tag{3.21}$$

где $E_р$ – расчётный коэффициент экономической эффективности капитальных вложений;

$Э_г$ – ожидаемая годовая экономия, тг;

К — капитальные вложения на создание системы, тг.

$$E_р = \frac{1\,292\,822}{931\,890} = 1{,}39$$

Расчетный срок окупаемости капитальных вложений составляет:

$$Т_Р = \frac{1}{E_Р} = \frac{1}{1{,}39} = 0{,}72 = 8{,}7 \text{ месяцев.}$$

Основные показатели экономической эффективности проекта представлены в таблице 3.9:

Таблица 3.11 – Основные показатели экономической эффективности

Показатель	Значение
Ожидаемый годовой экономический эффект, тг	1 292 822
Коэффициент экономической эффективности капитальных вложений	1,39
Срок окупаемости капитальных вложений, месяцев	8,7

3.9 Выводы Веб-приложения технико-экономическому обоснованию

В данной главе были проанализированы и вычислены затраты на разработку программного обеспечения децентрализованной системы. ВЕБ-ПРИЛОЖЕНИЯ обеспечит систему, где все ваши данные будут хранится в безопасности и создаст условия выполнения транзакций с выводом кошелька. Больше не придется использовать посредников(3-лиц) таких как банки, органы, так как вся информация будет в электронном виде.

Ожидаемый годовой экономический эффект составит 1 292 822тенге. Приложение окупится в первые 8,7 месяцев его использования.

4. Безопасность жизнедеятельности

4.1 Эскиз и схема помещения

И так мы имеем помещение размером 8.80х3.80 м. Площадь 33.44 м^2. В данном помещени находятся 4 сотрудника

Схема помещения представлен на рисунке 4.1

Наше помещение а это стены, потолок, паркет и мебель выполнено строго в белых тонах для лучшего комфорта работы, а высота потолков составляет 3 метра.

Таблица 4.1 – описание помещения

Площадь помещения	33.44 м^2
Виды освещения	Искусственное и естественное
Количество ламп	12
Количество рабочих мест	4
Количество окон	4
Компьютеры	Ноутбук Acer Nitro (4шт)
Сетевое оборудование	Wifi роутер Huawei

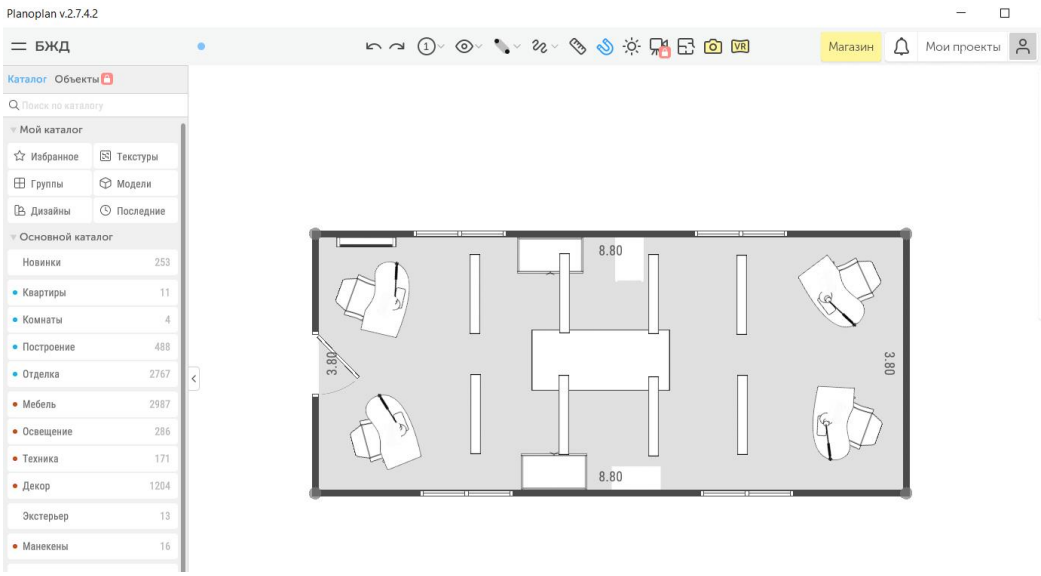


Рисунок 4.1 – Схема помещения

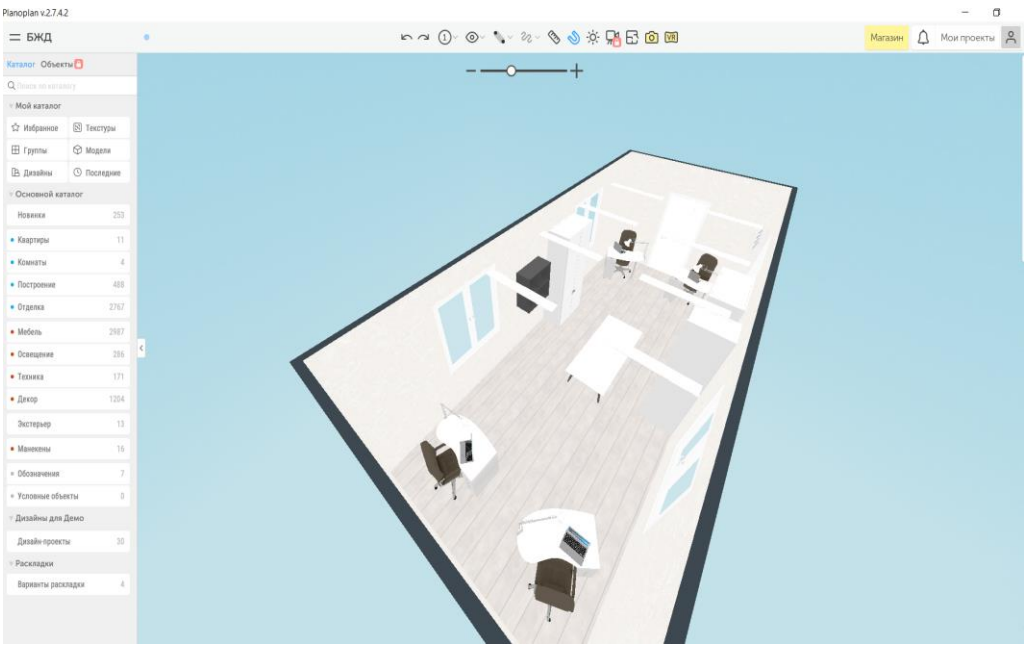


Рисунок 4.2 – Эскиз помещения

4.2 Разработка системы место освещения рабочего места

Ошибочный свет может повлиять на фактор производственного травматизма. Свет воздействует на производительность труда и качество издаваемой продукции. Освещение трудящегося - серьезная причина в создании хороших условий труда. Нехорошее освещение возможно абсолютно полностью поменяет информацию, получаемую людям посредством зрения, более того, оно не только утомляет зрение, а вызывает усталость организма в целом, отрицательно воздействует на состоянии центральной нервной системы. вдобавок немощное освещение возможно повергнет к чрезвычайным ситуациям на предприятии [10].

Свет воздействует на производительность труда и качество продукта. Так при выполнении акта четкой производительности повышения освещенности с 50 до 1000 лк позволяет заполучить увеличение производительности труда до 25 % и при выполнении работ небольшой точности, не требующих наибольшего зрительного напряжения, повышение освещенности рабочего места увеличивает эффективность труда на 2-3%

Оптической сферой диапазона именуется элемент электромагнитного спектра с длиной волны = 10 - 340 нм. Она делится на:

– инфракрасное излучение (= 340 - 770 нм), которое проявляется в основном в тепловом воздействии;

– видимое излучение (= 770 - 380 нм): в зависимости от длины волны вызывает у человека, различные световые и цветовые ощущения: от фиолетового (= 400 нм) до красного (= 750 нм). Зрение наиболее чувствительно к излучению с длиной волны = 550 нм. что соответствует желто-зеленому цвету: к границам видимого спектра чувствительность уменьшается;

– ультрафиолетовое излучение (= 380 - 10 нм). УФ излучение оказывает биологически положительное воздействие на организм человека, вызывая загар При высокой интенсивности УФ излучение способно вызвать ожог кожи. глаз. УФ излучение возникает при электро и газовой сварке, при работе кварцевых ламп, электрической дуги высокой интенсивности, лазерных установок. Защита от УФ излучений проста - их пропускают на ткань одежды и очки с простым стеклом [10].

Основные световые величины и параметры, определяющие зрительные условия работы

К количественным показателяи производственного освещения относятся:

– лучистый поток;
– световой поток;
– сила света;
– яркость;
– освещенность.

Лучистый поток (Ф) - общая мощность электромагнитного излучения оптическом диапазоне длин волн. Единицей измерения служит /Вт/.

Фон - поверхность, непосредственно прилегающая к объекту различения, на которой он рассматривается. Фон характеризуется коэффициентом отражения , под которым понимается способность поверхности отражать падающий на нее световой поток.

В зависимости от величины коэффициента отражения фон может быть:

– светлым (р > 0,4);
– средним (р = 0,2 т 0,4);
– темным (р < 0,2').

Контраст может быть:

– большим (К > 0,5);
– средним (К = 0,2 ^ 0,5);
– малым (К < 0,2).

Расчёт искусственного освещения

Таблица 4.2 характеристика светильника

Тип Светильника	ЛСП 01
Длина	18
Ширина	18
Высота	5,1
Разряд зрительных работ	III

Продолжение таблицы 4.1

Подразряд зрительных работ	а
Коэффициенты отражения потолка	50%
Коэффициенты отражения стен	30%

1) Площадь, подлежащая освещению.

$$S = A * B$$

где S – площадь, подлежащая освещению
A – длина помещения,
B – ширина помещения.

$$S = 8.8 * 3.8 = 33.44 \text{ м}^2$$

Норма освещённости на рабочих поверхностях в зависимости от зрительных работ по СниП 23-05-95.

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Разряд зрительной работы	Подразряд зрительной работы	Контраст объекта с фоном	Характеристика фона	Искусственное освещение					Естественное освещение		Совмещенное освещение	
						Освещенность, лк			Сочетание нормируемых величин показателя ослепленности и коэффициента пульсации		КЕО $e_н$, %			
						при системе комбинированного освещения		при системе общего освещения			при верхнем или комбинированном освещении	при боковом освещении	при верхнем или комбинированном освещении	при боковом освещении
						всего	в том числе от общего		P	$K_п$, %				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Высокой точности	От 0,30 до 0,50	III	а	Малый	Темный	2000	200	500	40	15	-	-	3,0	1,2
						1500	200	400	20	15				
			б	Малый Средний	Средний Темный	1000 750	200 200	300 200	40 20	15 15				
			в	Малый Средний	Светлый Средний	750	200	300	40	15				
				Большой	Темный	600	200	200	20	15				
			г	Средний Большой	Светлый "	400	200	200	40	15				
				"	Средний									

Рисунок 4.3 – СниП 23-05-95

Характеристики зрительной работы:

– разряд зрительных работ - III;
– подразряд зрительных работ – а;
– степень точности - высокой точности;
– освещенность - 400 лм.

2) Тип и количество ламп.

Тип светильника ЛСП 01.

Лампы - люминесцентные.

Количество ламп в светильнике - 2 шт.

Общее количество ламп - 8 шт.

3) Коэффициент использования светового потока.

$$\eta=50\%$$

4) Величина светового потока для одной лампы.

$$\Phi = \frac{100 * E_H * S * z * k}{N * n * \eta}$$

где Ф – световой поток лампы, лм;

E_H – нормируемая минимальная освещённость = 400лк

S – площадь освещаемого помещения, м^2; S = 33,4

z – коэффициент минимальной освещённости; = 1,15;

k – коэффициент запаса; 1,1

N – число светильников в помещении; = 8 шт.

n – число ламп в светильнике. = 2 шт.

$$\Phi = \frac{100 * 400 * 33{,}4 * 1.15 * 1.1}{8 * 2 * 50} = 2112 \text{ лм}$$

И так мы получили световой поток одной лампы. Теперь высчитаем процент отличия табличного значения от нашего расчетного.

4.3 Система и виды производственного освещения

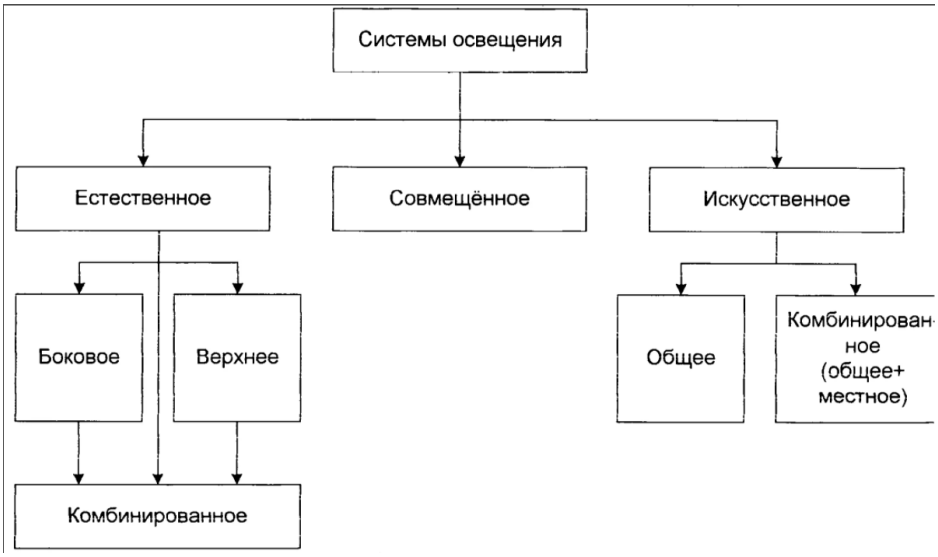


Рисунок 4.4 – Классификация систем оповещения

Системы производственного освещения можно классифицировать в зависимости от источника света и по конструктивному исполнению.
По источнику света производственное освещение может быть:

- естественным, созданным небесным светом;
- искусственным, осуществляемым электрическими лампами;
- совмещенным, представляющим собой сочетание естественного и искусственного.

Влияние параметров световой среды на здоровье человека.

В зависимости от спектрального состава свет может оказывать возбуждающее действие и усиливать чувство тепла (оранжево-красный), или, наоборот, успокаивающее (желто – зеленый), или усиливать тормозные процессы (сине – фиолетовый). Это используется при эстетическом оформлении производственных помещений, окраске оборудования и стен:

- холодные тона – при высоких температурах и наличии источников тепловыделений, в жарком климате;
- теплые тона – в случае понижения температуры, необходимости тонизирующего влияния производственной среды на работающих.

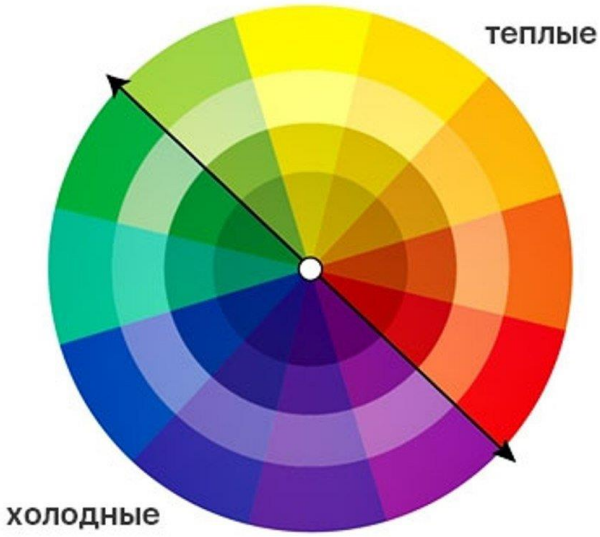


Рисунок 4.5 – Тональности «теплых» и «холодных» цветов

Наиболее широко используется зеленый цвет, оказывающий благоприятное психологическое воздействие.

Наиболее значительное влияния освещения оказывает на функцию зрения, а через нее на производительность труда. Рациональное освещение играет важную роль в профилактике производственного травматизма.

Возможность отрицательного воздействия условий освещения на работников определяется рядом факторов:

- отсутствием или недостаточностью естественного света;
- пониженной освещенностью;
- повышенной яркостью;
- прямой или отраженной блескостью;
- повышенной пульсацией освещенности;
- повышенным уровнем ультрафиолетового излучения.

С отсутствием естественного света связанно явление «светового голодания».

Для компенсации ультрафиолетовой недостаточности используются УФ – облучательные установки длительного действия (совмещенные с осветительными установками) и облучательные установки кратковременного действия (фонари).

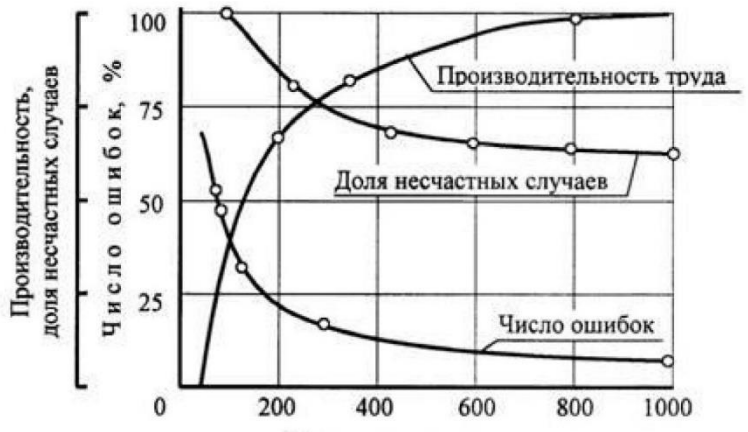


Рисунок 4.6 – Влияние освещенности на эффективность зрительной работы

Плохие условия для зрительных занятий появляются не только лишь при пониженной, но и чрезмерной освещенности. При высокой освещенности поверхности и высочайшем коэффициенте отражения в связи с высокой яркостью может образовываться ослепляющее действие, положение зрительного дискомфорта.

Предотвращению отрицательного влияния высокой яркости является точное устройство осветительных приборов, соблюдение призываемых степеней освещенности.

Помимо освещенности на эффективность зрительной работы воздействуют вдобавок показатели свойства освещения.

В частности, выполнение задания в условиях освещения пульсирующим светом понижает трудоспособность органа зрения, активизирует увеличенное утомление, головные боли и т. Мало того, присутствие в поле зрения перемещающихся и вертящихся предметов, даже при невысоких значениях коэффициента пульсации, возможно вызывание стробоскопического эффекта что приводит к производственному травматизму [10].

4.4 Разработка рабочего места с учётом эргономических требований

В настоящее время стремительное формирование нынешней техники значимым образом меняет дело человека. В связи с этим появляется проблема функционального вырабатывания возможностей человека в соответствии с требованиями, что предъявляет к нему технический прогресс, и возможностями, что пред ним открываются с развитием техники. Рабочее место сообразно системой рабочего места обязано быть снабжено исполнение трудовых действий в пределах зоны досягаемости моторного поля.

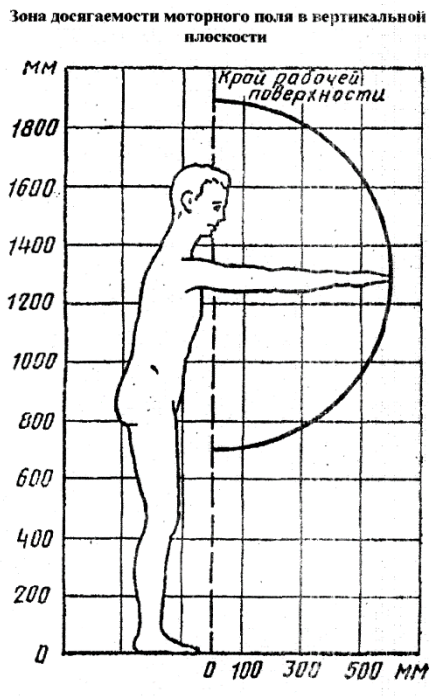


Рисунок 4.7 — Зона досягаемости моторного поля в вертикальной плоскости.

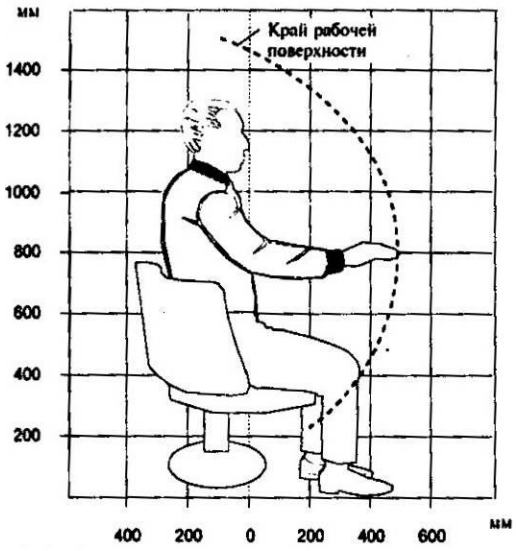


Рисунок 4.8 — Зона досягаемости моторного поля в горизонтальной плоскости при высоте рабочей поверхности над полом 800 мм.

Выполнение частых трудовых операций должно быть обеспечено в пределах зоны легкой досягаемости и оптимальной зоны моторного поля.

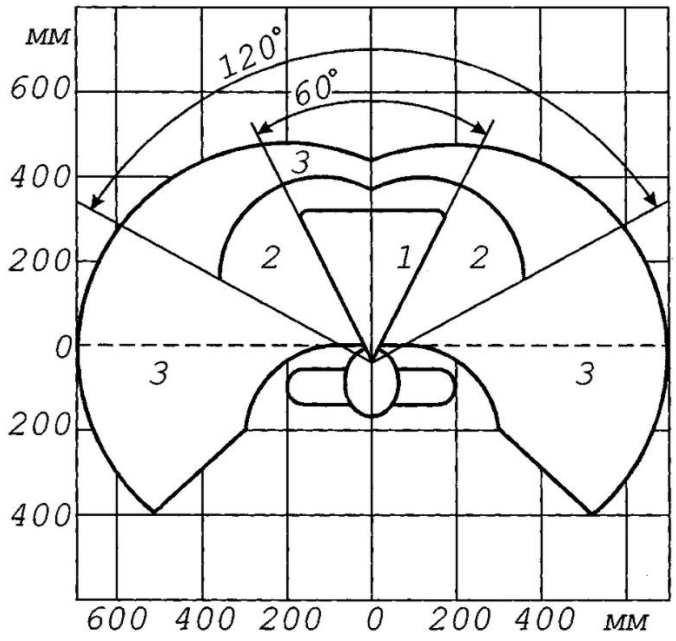


Рисунок 4.9 — Зоны для выполнения ручных операций:

1 — участок для размещения особенно значительных и чаще используемых объектов (оптимальная область моторного поля).

2 — участок на размещение ежеминутно используемых объектов (зона свободной досягаемости моторного поля).

3 — область для размещения редко используемых объектов (зона досягаемости моторного поля). Габаритные характеристики рабочего места установка рабочего места обязана гарантировать приемлемое расположениие сотрудника, что достигается постредством регулирования высоты сиденья и подставки для ног [10].

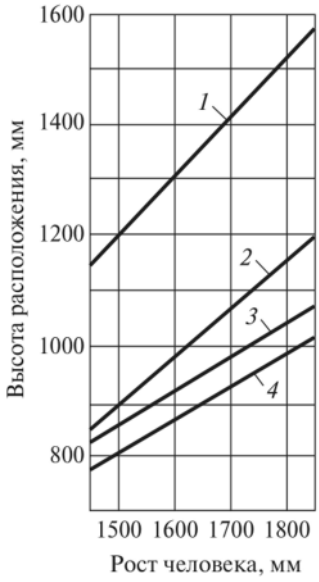


Рисунок 4.10 — Номограмма зависимости высоты рабочей поверхности

Места для ног а также высоты рабочего сиденья зависят от роста человека. значительным моментом приходится место под столом, оно должно составлять довольно большое пространство дабы комфортно сгибать и разгибать колени.

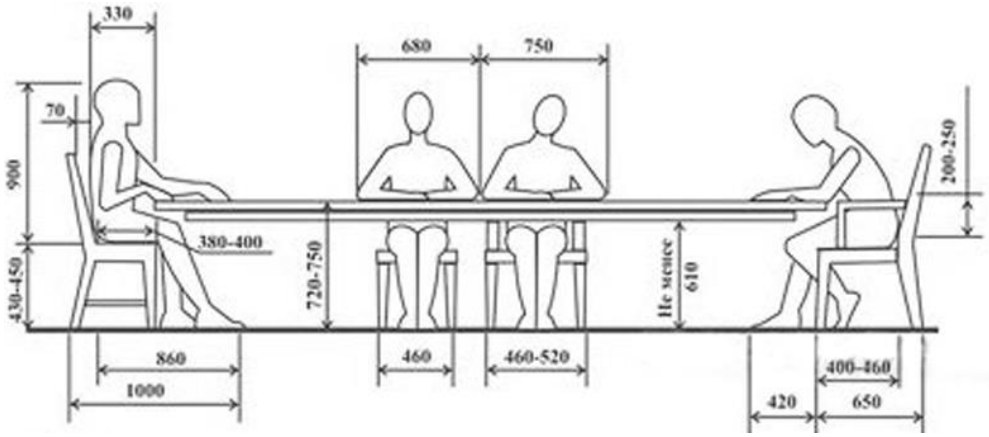


Рисунок 4.11 — Пространство для ног (ширина не менее 500 мм):

a — расстояние от сиденья до нижнего края рабочей поверхности не менее 150 мм.

h — высота пространства для ног не менее 600 мм. Площадь столешницы не должна быть менее 1 м2.

Чем массивнее стол, тем лучше, меньше пульсации от техники. вредоносными для самочувствия являются: высокая либо пониженная температура воздуха; мебель обязано соблюдать физически разумную рабочую позу, при которой не изменяется циркуляция крови и не происходит и не воздействуют другие факторы.

Для этой реализации нужно дабы у кресла имелась гибкая спина анатомической формы, которая сбавит нагрузку на позвоночник. вдобавок для этого нужно убирать нагрузку с мышц плечевой зоны кресло непременно должно быть с подлокотниками и иметь функцию поворота, изменения высоты а также угол наклона сиденья и спинки. Для работы нужен плоскоэкранный дисплей с диагональю как минимум 17” или 19” приемлемое разрешение — 1024*768 или 1280*1024 [11].

4.5 Вывод по БЖД

В данной работе мы узнали влияние освещения рабочего места на производительность и скорость работы а также удобность расположения рабочих элементов. Измерили и выяснили какое свободное расстояние требуется для ног и расстояние от глаз к монитору. Как влияют цвета на настроение и настрой работников. И что к каждому человеку нужен свои определенные условия связанные с его здоровьем.

Заключение

С учетом роста сети Интернет, количества веб-приложений, пользователей и неопытных разработчиков проблема безопасности веб-приложений будет оставаться актуальной на всех уровнях и требовать комплексного подхода. При разработке методологии исследования нарушений работы веб-приложений, представленной в дипломной работе, были обнаружены проблемы со значительной фрагментарностью рынка, не позволяющую подойти к процессу поиска и исправления уязвимостей централизовано.

Одновременно с этим, существующие пользовательские средства защиты, включая antisql, плагины безопасности встроенные в CMS и прочие средства разработки веб-приложений, не могут обеспечить приемлемый уровень безопасности. Описанная в дипломной работе методология поиска уязвимостей, предполагает использование внушительного комплекса программного обеспечения, несколько установленных операционных систем и умение систематизировать данные полученные в ходе анализа. Что приводит к выводу о необходимости создании более полных комплексов разновариативного поиска уязвимостей и нарушений безопасности реализованных в одном. Что позволит специалистам проводить аудит системы согласно разработанной методологии в одном окне.

Получая общий консолидированный отчет и упрощая перемещение и обмен информацией между инструментами. В результате проведенного анализа веб-приложения компании ТОО «КазГеоПозиция», размещенного по адресу kazgp.kz, с использованием разработанной методологии поиска нарушений безопасности, выявленные ошибки и уязвимости были переданы в отдел веб-разработки и даны рекомендации по возможному способу их устранения [11].

Также в ходе дипломной работы были решены следующие задачи:

– рассмотрен и проанализирован теоретический материал;

– определено необходимое программное обеспечение для исследования нарушения работы веб приложений;

– разработана методология исследования для выявления нарушений безопасности веб-приложения;

– проведено практическое исследование веб-приложения компании «КазГеоПозиция»;

– проанализирован экономический аспект проведения исследования веб-приложения компании «КазГеоПозиция».

Методологическую и информационную основу проводимых исследований составили труды отечественных и зарубежных экспертов. Теоретическая работа позволила разработать методологию исследования нарушений работы веб-приложений и практически ее использовать при проведении аудита безопасности веб-приложения компании ТОО «КазГеоПозиция» размещенного по адресу http://kazgp.kz/. Благодаря

современному подходу, использованию новых теоретических данных, системному подходу к анализу, была достигнута существенная экономия времени и учтен экономический фактор внедрения специалиста по аудиту веб-приложений.

Перечень сокращений

Интернет – всемирная система объединённых компьютерных сетей для хранения, обработки и передачи информации.

ПО – программное обеспечение или программа, или же программный комплекс который используется для управления компьютером

ОС – это комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

Инструментирование это отслеживание параметров уровня производительности кода, возможность диагностировать ошибки и записывать информацию для отслеживания причин их возникновения.

Concolic представляет собой смесь между исполнением CONCrete и исполнения symbOLIC, с целью технико-экономического обоснования.

Дебаг (debug) — программа-отладчик, которую используют для проверки и отладки выполняемых файлов. Использовалась при операционной системе DOS. Под более поздние версии операционных систем работает через эмулятор DOS и имеет ограниченные возможности. Также иногда называют процесс отладки программы.

Релиз — выпуск, демонстрация, публикация, показ — фильма, книги, пластинки, продукталогирование/

Логи (лог-файлы) — это файлы, содержащие системную информацию работы сервера или компьютера, в которые заносятся определенные действия пользователя или программы. Иногда также употребляется русскоязычный аналог понятия — журнал.

OWASP Open Web Application Security Project, это открытый проект обеспечения безопасности веб-приложений. Сообщество OWASP включает в себя корпорации, образовательные организации и частных лиц со всего мира.

GDPR — General Data Protection Regulation Общий регламент по защите данных Европейского союза, обновленный в мае 2018 года.

Bug Bounty — это программа, предлагаемая некоторыми веб-сайтами и разработчиками программного обеспечения, с помощью которой люди могут получить признание и вознаграждение за нахождение ошибок, особенно тех, которые касаются эксплойтов и уязвимостей. Эти позволяют разработчикам обнаружить и устранить ошибки, прежде чем общественность узнает о них.

UDP (англ. User Datagram Protocol — протокол пользовательских датаграмм) — один из ключевых элементов набора сетевых протоколов для Интернета. С UDP компьютерные приложения могут посылать сообщения (в данном случае называемые датаграммами) другим хостам по IP-сети без необходимости предварительного сообщения для установки специальных каналов передачи или путей данных.

Wi-Fi — технология беспроводной локальной сети с устройствами на основе стандартов IEEE 802.11паттернов

Бэкдор, тайный вход (от англ. back door — «чёрный ход», буквально «задняя дверь») — дефект алгоритма, который намеренно встраивается в него разработчиком и позволяет получить несанкционированный доступ к данным или удалённому управлению операционной системой и компьютером в целом.

Сетевой червь — разновидность вредоносной программы, самостоятельно распространяющейся через локальные и глобальные (Интернет) компьютерные сети.

Троянская виирусная программа (также — троян, троянец) — разновидность вредоносной программы, проникающая в компьютер под видом легитимного программного обеспечения, в отличие от вирусов и червей, которые распространяются самопроизвольно. В данную категорию входят программы, осуществляющие различные неподтверждённые пользователем действия.

CSRF (англ. cross-site request forgery — «межсайтовая подделка запроса», также известна как XSRF) — вид атак на посетителей веб-сайтов, использующий недостатки протокола HTTP.

Плагин – это независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Плагины обычно выполняются в виде библиотек общего пользования

Список литературы

1) Shabtai A., Mimran D., Elovici Y. Evaluation of Security Solutions for Android Systems // arXiv preprint arXiv:1502.04870. — 2015.

2) Zhou X. et al. The peril of fragmentation: Security hazards in android device driver customizations // Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, 2014. P. 409—423.

3) Hoffmann J. From Mobile to Security: Towards Secure Smartphones: дис. – 2014.

4) Джоел Скембрей Д. Вонг Майк Шема, Секреты хакеров. Безопасность Web-приложений - готовые решения 340-350.

5) Sun M., Tan G. NativeGuard: Protecting android applications from third-party native libraries // Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks. ACM, 2014. P. 165—176.

6) Tan D. J. J. et al. Securing Android: A Survey, Taxonomy, and Challenges // ACM Computing Surveys (CSUR). 2015. Vol. 47. № 4. P. 58.

7) ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2020/

8) phdays.ru/program/fast-track/45984

9) copperdroid.isg.rhul.ac.uk/copperdroid

10) securityaffairs.co/wordpress/37667/hacking/nfc-attack-credit-card.html

11) zerodayinitiative.com/advisories/ZDI-15-092/