МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Зав. каф. «КБИС»
_____ Д.А. Короченцев
(подпись)
«___» _____ 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема: «РАЗРАБОТКА МОДУЛЯ ДЛЯ ДОПОЛНИТЕЛЬНОЙ ЗАЩИТЫ САЙТОВ, СОЗДАВАЕМЫХ НА JOOMLA»

Специальность 10.05.01 Компьютерная безопасность

Специализация Математические методы защиты информации

Обозначение ВКР 10.05.01.550000.000 Группа ВКБ61

Обучающийся _____ М.Ю. Беловолов
подпись, дата

Руководитель ВКР _____ доцент О.В. Куликова
подпись, дата

Ростов-на-Дону
2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Тема: «РАЗРАБОТКА МОДУЛЯ ДЛЯ ДОПОЛНИТЕЛЬНОЙ ЗАЩИТЫ САЙТОВ, СОЗДАВАЕМЫХ НА JOOMLA»

Обучающийся Беловолов Михаил Юрьевич

Обозначение ВКР 10.05.01.550000.000 Группа ВКБ61

Тема утверждена приказом по ДГТУ от 19.01.2023 г. № 221-ЛС-О

Срок представления ВКР к защите « » февраля 2023 г.

Исходные данные для выполнения выпускной квалификационной работы:

Федеральный закон № 149-ФЗ от 27 июля 2006 г. «Об информации, информационных технологиях и о защите информации».
Методический документ ФСТЭК России «Методика определения угроз безопасности информации в информационных системах».
Руководящий документ Гостехкоммисии РФ «Специальные требования и рекомендации по технической защите конфиденциальной информации».
Руководящий документ «Автоматизированные системы. Защита от НСД к информации. Классификация автоматизированных систем и требования по защите информации».
Материалы исследований, проводимых в ходе преддипломной практики.

Содержание выпускной квалификационной работы

Введение:
Во введении необходимо: изложить актуальность выбранной темы, обозначить объект и предмет исследования, цель и задачи выпускной квалификационной работы, теоретическую и практическую значимость работы, структуру работы.

Наименование и краткое содержание разделов:
1 Теоретические основы.
Представлена теоретическая информация о системах управления содержимым. Определение CMS, основные виды CMS, чем отличаются, преимущества и недостатки выбора системы управления содержимым для разработки.
2 Основные виды CMS.
Более подробный разбор систем управления содержимым. Рассмотрен рейтинг самых популярных систем за последний год. По каждой из систем проведен анализ, рассмотрены недостатки и преимущества по отношению к другим системам.
3 Проблемы безопасности современных CMS.
В данном разделе детально рассматриваются уязвимости самых популярных и используемых систем управления содержимым. По каждой системе рассмотрено несколько примеров проблем безопасности, рассмотрена сама суть каждой из проблем, как проблема была устранена и быстрота реагирования разработчиков систем.
4 Сравнительный анализ существующих программных решений.
Рассматриваются аналоги настоящей разработки. Их актуальность, удобство пользования и т.д. Представлен краткий обзор и сравнительный анализ аналогов.
5 Алгоритмическая реализация программного средства.
Данный раздел содержит анализ входных данных, выходных данных и структуры программного средства. Представлена блок-схема работы разработанного модуля, описан алгоритм работы.
6 Программная реализация программного средства.
Приведено обоснование выбора средств разработки и языка программирования, модули для разработки, программного средства для сканирования файлов, подробно описаны модули программного средства, иерархия и предназначения классов и основные методы модулей программного средства. Также приведен пошаговый пример использования программного средства, описаны промежуточные результаты интеграции и работы программного средства. Результат демонстрации работоспособности разработанного программного средства позволяет сделать вывод о его применимости.

Заключение:
Основные выводы о проделанной работе, и оценка достижения цели.

Перечень графического и иллюстрационного материалов:
Презентация выпускной квалификационной работы на тему: «Разработка модуля для дополнительной защиты сайтов, создаваемых на Joomla».

Руководитель проекта (работы) _____ доцент О.В. Куликова
подпись, дата

Задание принял к исполнению _____ М.Ю. Беловолов
подпись, дата

Аннотация

Данная работа посвящена разработке программного модуля для дополнительной защиты сайтов, создаваемых на CMS Joomla. Данный модуль интегрируется в систему управления контентом Joomla как расширение, после чего им можно пользоваться напрямую из административной панели системы. В выпускной квалификационной работе приведены примеры существующих на рынке аналогов программных решений, модулей и расширений, обеспечивающих безопасность сайтов, разработанных на CMS. Структурно представлены детали разрабатываемого программного модуля. Приведена демонстрация работы модуля, по результатам которой можно судить о корректности, актуальности и практической применимости разработанного программного средства.

Объем работы – 79, количество иллюстраций – 24, приложений – 5, использовано информационных ресурсов – 23.

Annotation

This work is devoted to the development of a software module for additional protection of sites loaded on the CMS Joomla. This module is integrated into the Joomla content management system as an extension, after which it can be used from the system's administrative panel. When working with a qualification assessment, there are often examples of the implementation on the market of software solutions, modules and extensions, ensuring the security of sites developed on CMS. Structurally represented part of the developed software module. A demonstration of the operation of the module is given, the results of which can be used to judge the correctness, relevance and practical applicability of the developed software.

Number of pages – 79, number of illustrations – 24, applications – 5, used information resources – 23.

Содержание

Введение..... 5

1 Теоретические основы..... 7

2 Основные виды CMS..... 11

2.1 WordPress..... 12

2.2 Joomla..... 14

2.3 1С-Битрикс..... 16

2.4 OpenCart..... 19

2.5 CMS.S3..... 21

3 Проблемы безопасности современных CMS..... 23

4 Сравнительный анализ существующих программных решений..... 30

4.1 Antivirus Website Protection..... 31

4.2 Website Antivirus Scanner..... 33

4.3 VirusTotal..... 34

4.4 JoomScan..... 38

5 Алгоритмическая реализация программного средства..... 41

6 Программная реализация программного средства..... 43

6.1 Средства разработки..... 43

6.2 Структура программного средства..... 47

6.3 Демонстрация работы программного средства..... 48

Заключение..... 53

Перечень использованных информационных ресурсов..... 54

Приложение А Техническое задание..... 57

Приложение Б Руководство системного программиста..... 61

Приложение В Руководство программиста..... 63

Приложение Г Руководство оператора..... 65

Приложение Д Листинг программы..... 66

Введение

В настоящее время для управления структурой, контентом и дизайном сайтов часто применяются системы управления содержимым сайта – Content Management System (CMS). К CMS можно отнести универсальные системы для управления контентом, интернет-магазины, блоги, форумы, фото и видео-галереи и другие компоненты для управления содержимым сайта. Такие системы позволяют разработчикам упростить программирование, дизайн, поддержку сайта и даже поручать работу с сайтом людям, незнакомым с программированием и веб-архитектурой.

В то же время, CMS, как и другие виды программного обеспечения (ПО), содержат уязвимости. В отличие от систем управления содержимым собственной разработки, если злоумышленники находят уязвимости в одной конкретной тиражируемой CMS – возникает угроза взлома всех систем этой версии. При этом, чем более распространённой является CMS и чем чаще она применяется на популярных сайтах, тем больше усилий и денег злоумышленники инвестируют в поиск её уязвимостей.

Кроме того, большинство современных CMS состоят из множества модулей, и многие уязвимости связаны с плагинами, которые обычно написаны и протестированы на безопасность хуже, чем основной код системы. Используя уязвимости CMS, злоумышленники стараются извлечь для себя выгоду за счёт чужих сайтов и посетителей. Например, на страницах сайта может быть размещён код, который заражает компьютеры посетителей вредоносными программами. Также, на сайте со взломанной CMS могут без ведома разработчика публиковать контент сомнительного содержания или автоматически перенаправлять пользователей на другие ресурсы с вирусным содержимым. В результате, под угрозой безопасность не только файлов и данных сайта, но и его посетителей, в случае посещения подобных страниц.

Целью выпускной квалификационной работы является разработка модуля для дополнительной защиты сайтов, создаваемых на Joomla. Цель определила следующие задачи:

– изучить теоретические основы систем управления содержимым;

– рассмотреть основные виды систем управления содержимым и их возможные уязвимости;

– рассмотреть основные методы дополнительной защиты сайтов;

– провести сравнение аналогов;

– разработать модуль для дополнительной защиты сайтов, создаваемых на Joomla.

Объектом выпускной квалификационной работы являются сайты, разрабатываемые на CMS Joomla.

Предметом данной работы является защита системы управления контентом CMS Joomla от вредоносного ПО.

1 Теоретические основы

Система управления содержимым (CMS) – информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым, иначе – контентом.

CMS обычно состоит из двух основных компонентов: приложения для управления контентом (CMA) в качестве внешнего пользовательского интерфейса, позволяющего пользователю добавлять, изменять и удалять контент с сайта без вмешательства разработчика, и приложение доставки контента (CDA), которое компилирует контент и обновляет сайт.

Основные функции CMS:

– предоставление инструментов для создания содержимого, организация совместной работы над содержимым;

– управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов;

– публикация содержимого;

– представление информации в виде, удобном для навигации, поиска.

В системе управления содержимым могут находиться разного вида данные: документы, фильмы, фотографии, номера телефонов, научные данные и т.д. Такая система часто используется для хранения, управления, пересмотра и публикации документации. Контроль версий является одной из важных возможностей, когда содержимое изменяется группой лиц.

В общем случае системы управления содержимым делятся на системы управления корпоративным контентом (Enterprise Content Management System) – для работы с содержимым внутри какой-либо организации и системы управления веб-содержимым (Web Content Management System) для поддержки работы сайта.

CMS позволяют управлять текстовым и графическим наполнением сайта, предоставляя пользователю интерфейс для работы с содержимым сайта, удобные инструменты хранения и публикации информации, автоматизируя процессы размещения информации в базах данных и её выдачи в HTML.

Существует множество готовых систем управления содержимым сайта, в том числе и бесплатных. Их можно разделить на три типа по способу работы, рассмотрим их ниже.

Первый тип – это генерация страниц по запросу. Системы такого типа работают на основе связки «модуль редактирования – база данных – модуль представления». Модуль представления генерирует страницу с содержанием при запросе на него, на основе информации из базы данных. Информация в базе данных изменяется с помощью модуля редактирования. Страницы заново создаются сервером при каждом запросе, что, в свою очередь, создаёт дополнительную нагрузку на системные ресурсы. Нагрузка может быть многократно снижена при использовании средств кэширования, которые имеются на современных веб-серверах.

Ко второму типу относится генерация страниц при редактировании. Системы этого типа служат для редактирования страниц, которые при внесении изменений в содержание сайта создают набор статических страниц. При таком способе значительно ухудшается интерактивность между посетителем и содержимым сайта.

Последний тип – смешанный. Следуя из названия, данный тип сочетает в себе преимущества первых двух. Может быть реализован путём кэширования – модуль представления генерирует страницу один раз, в дальнейшем она в несколько раз быстрее подгружается из кэша. Кэш может обновляться как автоматически, по истечении некоторого срока времени или при внесении изменений в определённые разделы сайта, так и вручную по команде администратора. Другой подход – сохранение определённых информационных

блоков на этапе редактирования сайта и сборка страницы из этих блоков при запросе соответствующей страницы пользователем.

Система управления – программа, предоставляющая инструменты для добавления, редактирования, удаления информации на сайте.

Ряд тиражируемых CMS имеет модульную архитектуру, модули можно подключать или не использовать, некоторые возможные модули: динамическое меню, блог, новости, опросы, поиск по сайту, статистика посещений, гостевая книга [4].

Система управления сайтом – это программный продукт, который создан для упрощения создания основных видов сайтов. Как правило, системы управления достаточно универсальны, то есть на них можно собрать что угодно: от информационного сайта до интернет-магазина или портала.

Эта платформа подходит для создания сайтов как простых, так и средних по сложности. На внешний вид публичной части сайта, т.е. на то, что видит обычный пользователь, CMS никаких ограничений не накладывают.

Есть две разновидности CMS – коробочные, готовый программный продукт, который можно скачать или купить, и самописные, которые можно получить только от разработчика. Рассмотрим только первые, так как вторые обычно представляют из себя просто тиражирование разработчиком однажды написанного проекта: отсюда проистекают проблемы с архитектурой, отсутствие документирования программного кода и, как следствие, модернизация и поддержка таких решений обычно весьма сложна.

К платным коробочным CMS можно отнести 1С-Битрикс, UMI.CMS, NetCat, CS-Cart, а к бесплатным – ModX, Drupal, Joomla, WordPress, Magento, OpenCart.

Подавляющее большинство коробочных CMS написано на PHP и использует СУБД MySQL.

Рассмотрим плюсы и минусы разработки на коробочных CMS.

На CMS достаточно просто и быстро можно создать те виды сайтов, которые заложены в функционал этой системы. При этом разработка простых решений часто даже не требует участия в проекте программиста.

Многие бизнес-процессы, которые нужны для конечных пользователей или требуются для управления сайтом, уже заложены в функционал CMS. Например, в рамках предустановленного функционала сайт на CMS уже «умеет» редактировать страницы и управлять их структурой, управлять пользователями системы и их уровнем доступа, сортировать и фильтровать различные объекты и многое другое.

Поддержка сайтов на CMS проще, чем работа с системами, написанными «с нуля»: этому способствует как наличие документации, так и высокий профессиональный уровень разработчиков CMS, который находит своё отражение в логичной и понятной архитектуре системы.

Возможно простое масштабирование в рамках заложенного функционала. Например, можно достаточно быстро добавить на информационный сайт каталог продукции, а затем сделать на базе каталога интернет-магазин.

Реализация нестандартного функционала или тонкая настройка под свои бизнес-процессы, как правило, затруднена. Процессы, заложенные в системе, обычно тесно связаны, поэтому модификация одного из них приводит к необходимости модифицировать и зависимые от него. А в некоторых случаях реализация специфического функционала в рамках CMS является и вовсе невозможной.

Обратной стороной универсальности CMS является избыточность функционала и сопутствующие этому увеличение сложности управления и использования, а также некоторые проблемы с производительностью, например, сайт на CMS выполняет гарантированно больше операции с данными, чем того требует процесс, а структура хранения данных обычно несколько «раздута» относительно реальных потребностей проекта.

2 Основные виды CMS

Среди современных систем есть подходящие для любого контента и решающие узкие задачи. Некоторые состоят из множества функциональных разделов, а другие представляют собой единую систему. Можно выделить простые, шаблонные, профессиональные и универсальные. Рассмотрим виды CMS детальнее [9].

Простые CMS – отличный вариант для сайтов из нескольких страниц с самым простым управлением. Их главные плюсы – свободный доступ в сети и отсутствие платы за пользование. К минусам можно отнести невозможность менять настройки, не предусматривается динамическое создание страниц и невозможно делегировать права адмна.

Шаблонные системы тоже состоят из ряда модулей с более сложной структурой. Их преимущества – это пропускная способность до 500000 запросов, поддержка динамических страниц, делегирование прав администратора.

Также есть профессиональные. Они отличаются гораздо более высоким уровнем сложности и изменяют структуру веб-ресурса, рассчитаны на подключение дополнительных модулей. Подойдут для информационных порталов [10].

Универсальные обладают широким функционалом. Они дорогостоящие и подходят для реализации веб-проектов, для которых важны высокая динамика, подразумевающая генерирование динамических страниц, изменение веб-дизайна, модификацию настроек и смену структуры [7].

CMS – это системы, одни из которых рассчитаны на решение конкретных задач, а другие универсальны, согласно информации, рассмотренной выше, также ПО предоставляется на платной или бесплатной основе, имеет открытый или закрытый код. Самые популярные CMS показаны на рисунке 1, где изображена диаграмма процентов различных систем от общего количества [5].

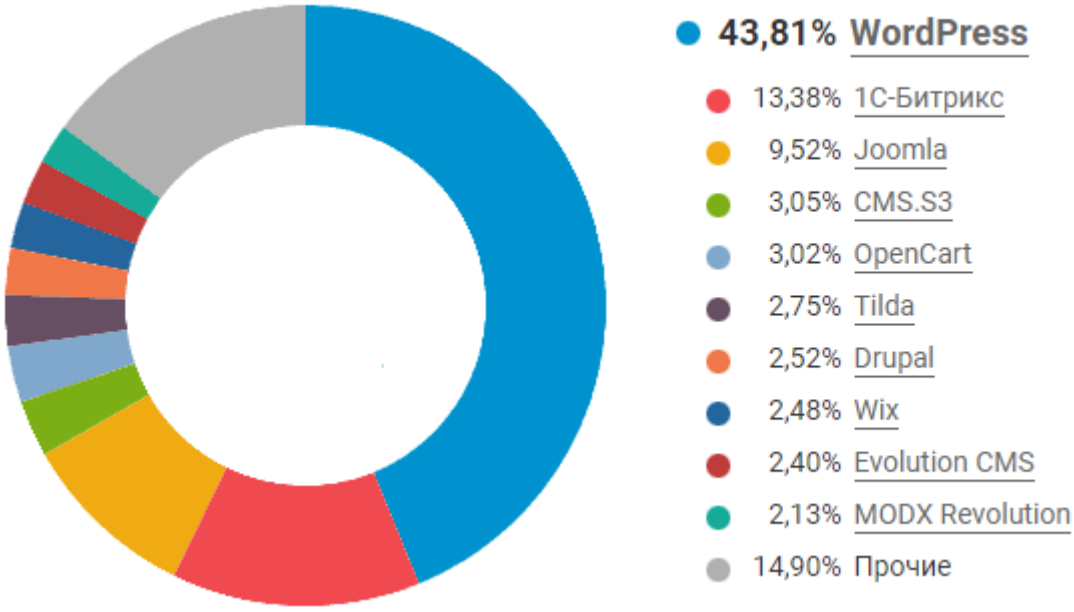


Рисунок 1 – Диаграмма популярности CMS

Рассмотрим более детально CMS, которые на данный момент можно отнести к лидерам на рынке.

2.1 WordPress

WordPress – это одна из наиболее популярных CMS в мире. С ее помощью можно создавать сайты различного характера и управлять ими без знаний и навыков программирования. Сегодня в сети существует множество различных программ или систем управления, но WordPress является самой популярной. Более трети всех сайтов в интернете работают на данной платформе. А если говорить исключительно о площадках, работающих на CMS, то на WordPress приходится больше половины сайтов, что делает ее однозначным лидером рынка.

Изначально WordPress задумывалась как платформа только для блогов. Каждый желающий мог создать свой сайт и делиться на нем новостями. Первый релиз произошел 27 мая 2003 года под версией 0.70.

Система с открытым кодом, это значит, что доступ к нему мог получить любой желающий программист. В результате люди активно стали писать свои собственные плагины, виджеты, расширения, тем самым делая функционал CMS WordPress все более и более широким. Кто-то стал на этом зарабатывать, а кто-то просто создавал бесплатные и полезные дополнения.

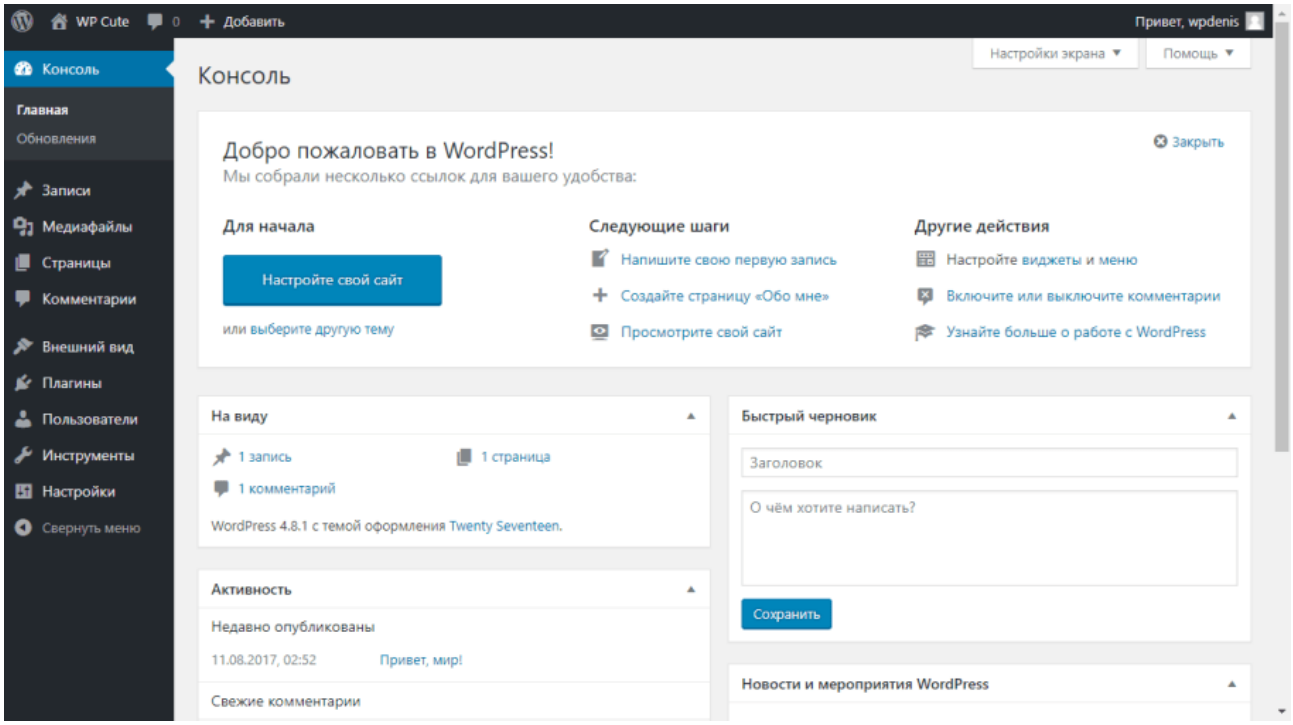


Рисунок 2 – Административная панель WordPress

Помимо того, что основной свой функционал платформа предоставляет бесплатно, а также имеет открытый доступ к коду, она является одной из самых удобных в использовании. Данный фактор немаловажен при выборе системы управления содержимым. Именно благодаря удобству и простоте использования любое физическое лицо, никогда не работающее с сайтами ранее, сможет освоить ее и создать собственный ресурс. Это удобство во многом обуславливается качественной организацией панели администратора, которая постоянно модифицировалась на протяжении долгих лет. Административная панель WordPress показана на рисунке 2.

Из-за роста популярности и постоянного расширения функционала WordPress со временем перестала позиционировать себя как исключительно платформа для блогов. Сегодня с помощью данной CMS можно создать практически любой сайт. WordPress подходит для интернет-магазинов, информационных и новостных порталов, корпоративных сайтов, блогов, коммерции.

2.2 Joomla

Joomla – это система управления контентом с открытым исходным кодом, используемая для создания веб-контента, она написана на PHP и использует базу данных MySQL для хранения данных, также использует методы объектно-ориентированного программирования. Это одна из самых популярных систем управления контентом благодаря таким функциям, как кэширование страниц, многоязычная поддержка, плагины и расширения.

Выпущена под лицензией GNU General Public License, Joomla основана на модели «View-Controlled Web Application Framework». Она очень удобна и гибка, а также является одной из быстрорастущих систем управления контентом. Как и другие системы управления контентом, Joomla также устраняет технические аспекты создания и запуска сайтов. Joomla имеет хорошую систему навигации, способную управлять несколькими иерархиями и подстраницами. Данная система также предоставляет административную панель с несколькими удобными функциями для пользователей.

Есть много преимуществ для использования Joomla. Она бесплатная и является платформой с открытым исходным кодом и использующая повторное использование кода (методология проектирования компьютерных и других систем) в соответствии с требованиями. Благодаря множеству доступных опций и функций Joomla можно использовать для создания любого типа сайта, блога,

или даже сайта электронной коммерции. Joomla легко устанавливается и легко разворачивается. Данная система управления содержимым является довольно доступной для понимания, это значит, что воспользоваться ей сможет любой пользователь, каких-то знаний программирования не требуется, все функции реализованы в виде интерфейса, что можно наблюдать на рисунке 3. Чтобы сделать сайты более привлекательными, Joomla также предоставляет различные шаблоны и темы.

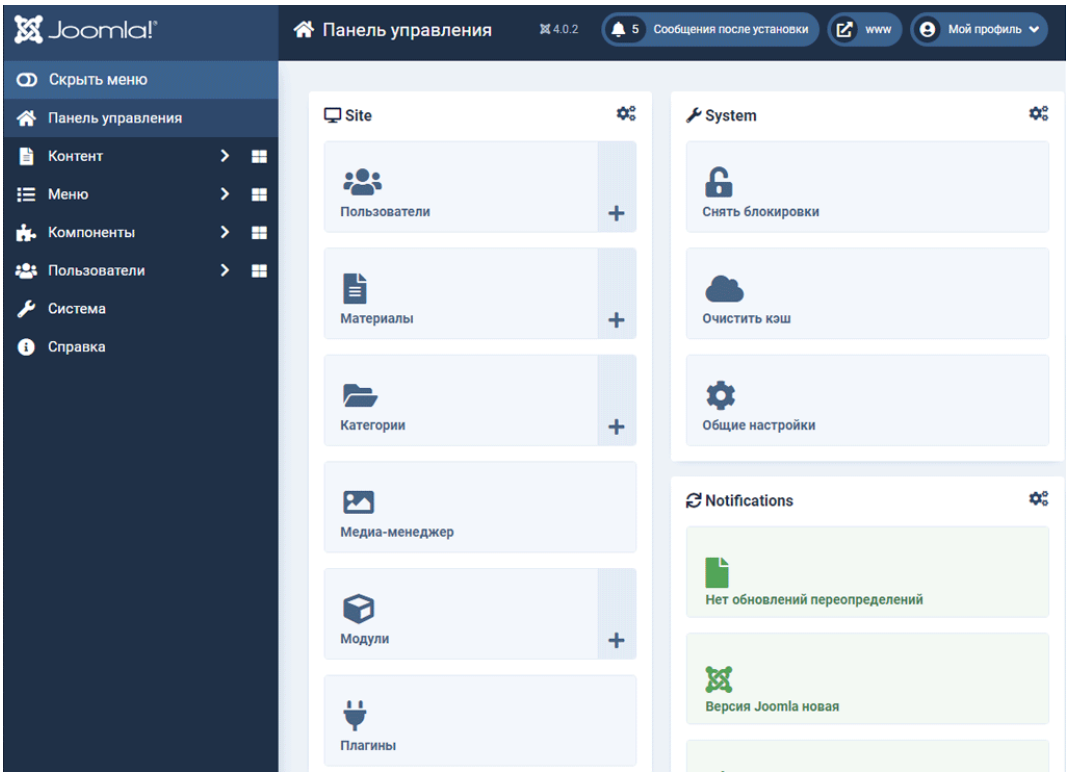


Рисунок 3 – Интерфейс административной панели Joomla

Однако есть несколько недостатков, связанных с данной системой. Некоторые из предоставленных плагинов могут работать только с определенными версиями системы. Joomla может не иметь расширенных функций, которые могут понадобиться при настройке сложных сайтов. Также данная система может усложнить загрузку и работу сайта, так как требует большого количества ресурсов сервера.

2.3 1С-Битрикс

«1С-Битрикс: Управление сайтом» – система управления содержимым веб-проекта от российской компании «Битрикс». Данная компания является, как мы уже разобрали ранее, российским разработчиком систем управления веб-проектами и корпоративной информацией. Программные продукты «1С-Битрикс» – профессиональные системы управления веб-проектами: сайтами компаний, интернет-магазинами, социальными сетями и сообществами, корпоративными порталами, системами аренды веб-приложений и другими проектами. Системы «1С-Битрикс» успешно работают на Windows и Unix платформах под управлением PHP и ASP.NET.

С помощью 1С-Битрикс можно разработать новый веб-проект или перевести существующий на новую систему управления.

Данная система состоит из большого количества модулей для управления всем сайтом: контентом, форумами, блогами, рекламой, интернет-магазинами и т.д. Как устроена классификация модулей показано на рисунке 4.

Система рассчитана как на профессиональных веб-разработчиков, так и на обычных пользователей, которые будут управлять готовым сайтом.

1С-Битрикс – хорошо интегрируется с другими продуктами 1С и подходит для реализации крупных проектов. Имеет большое количество маркетинговых инструментов. Среди них – виджеты для общения с клиентами, рассылка уведомлений для авторизованных посетителей. Преимущество 1С-Битрикс – неограниченная масштабируемость. Можно начать с маленького блога, форума, лендинга или сайта-визитки, а завершить огромным порталом с большим количеством сотрудников. А вот возможности визуального редактора ограничены, поэтому лучше использовать его для запуска целевых страниц в рамках продвижения отдельных услуг, нацеленных на бизнес и работу с клиентами и потенциальными покупателями [12].

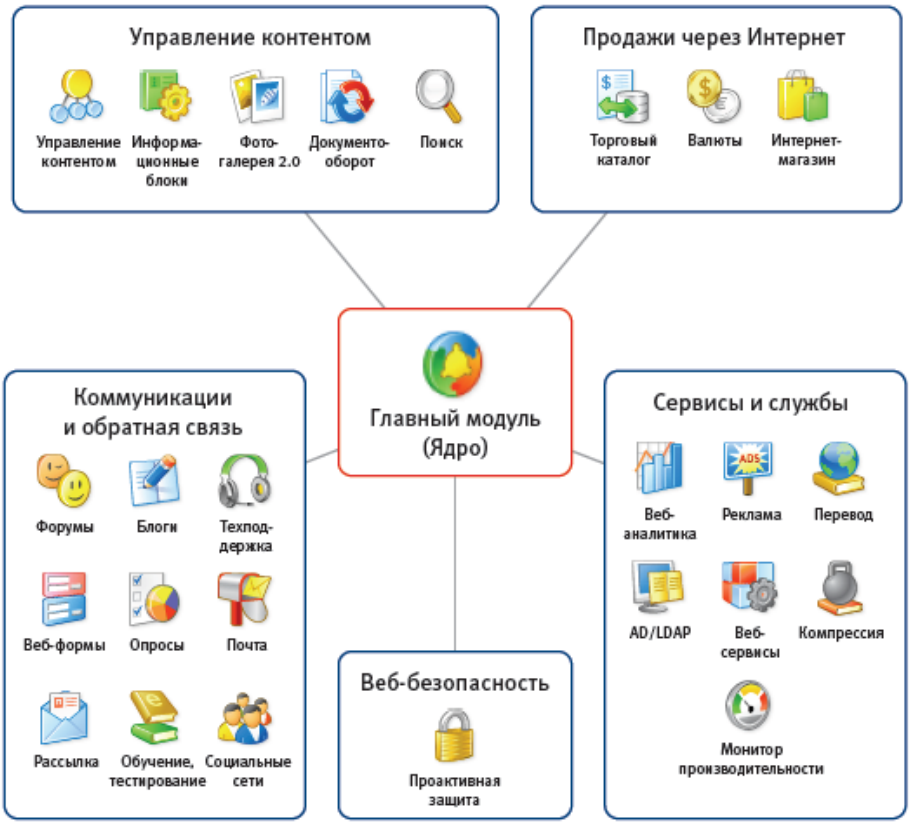


Рисунок 4 – Модули 1С-Битрикс

Для обеспечения высокого уровня защищенности сайтов от взломов проведен независимый аудит информационной безопасности. Аудит выполнен известной в области веб-безопасности российской компанией Positive Technologies. Тестирование встроенных механизмов защиты продукта подтвердило их соответствие требованиям Web Application Firewall Evaluation Criteria международной организации Web Application Security Consortium, о чем свидетельствует выданный сертификат. Качество реализации защитных механизмов 1С-Битрикс позволяет пользователю системы быть уверенным не только в надежности ядра системы, но и в безопасности решения на его основе, с учетом надстроек и доработок, выполненных партнерами компании.

Однако, в плане безопасности, регулярно возникают уязвимости, которые рассмотрим далее.

В апреле 2006 года журнал «Хакер» опубликовал статью, где рассказал о

взломе CAPTCHA в «1С-Битрикс: Управление сайтом», а в декабре 2013 года уже была опубликована информация об уязвимости в модуле e-Store позволяющая злоумышленникам узнать cookie пользователя и управлять его корзиной, удалять и добавлять товары. Уязвимости дан номер CVE-2013-6788. Начиная с версии 14.0.1 уязвимость была исправлена.

В ноябре 2015 года опубликована информация об уязвимости CVE-2015-8357 в модуле xscan. Уязвимость позволяет пользователям переименовывать произвольные файлы, получать доступ к конфиденциальным данным, и вызывать отказ в обслуживании. С версии xscan 1.0.4 уязвимость была исправлена.

В 2015 году была найдена критическая уязвимость CVE-2015-8358 в модуле mpbuilder. Уязвимость позволяет удалённо производить выход за пределы домашнего каталога, включать и исполнять удалённо код. Подвержены версии модуля mpbuilder версии ниже 1.0.12.

В феврале 2018 года «1С-Битрикс» обновила сертификаты от ФСТЭК России. Зафиксировано отсутствие недекларированных возможностей по четвёртому уровню (НДВ-4). Обновлённые сертификаты будут действовать до 2020 года.

В марте 2022 года была опубликована информация о критической уязвимости CVE-2022-27228 в модуле vote, обнаруженной Positive Technologies. Данная уязвимость позволяла исполнить произвольный код на сервере без какой-либо аутентификации.

В мае 2022 года на портале GitHub было опубликовано исследование, описывающее множество уязвимостей в системе «1С-Битрикс», и критикующее подход разработчиков к безопасности.

В День Конституции Украины, 28 июня 2022 года, множество сайтов, управляемых системой «1С-Битрикс», подверглись взлому. В том числе, пострадали ресурсы Росреестра, нескольких ВУЗов и региональных СМИ.

2.4 OpenCart

OpenCart – платформа электронной коммерции, ориентированная на создание интернет-магазинов. Является свободным программным обеспечением, распространяемым по лицензии GNU General Public License v3. Системой OpenCart поддерживаются дополнения-модули и шаблоны RU – EN, т.е. шаблоны на русском и английском языках.

Система создана и поддерживается Дениэлем Керром в репозитории GitHub. Программное обеспечение написано на языке программирования PHP, а в архитектуре использован шаблон проектирования MVC.

OpenCart ориентирована на создание интернет-магазинов, размещение в них товаров с различными вариантами доставки и оплаты.

Данная система была написана в 1998 году Кристофером Манном для Walnut Creek CDROM. Первый публичный релиз состоялся 11 мая 1999 года. Разработанный на языке Perl, изначально проект развивался слабо и окончательно был заброшен в 2000 году, когда Манн заявил, что он больше не может развивать OpenCart, по причине наличия у него других жизненных обязательств, более важных.

Вторую жизнь система обрела благодаря британскому разработчику Дэниэлу Керру, который использовал наработки Манна для создания своего собственного движка на PHP. Первый релиз обновлённого OpenCart состоялся 10 февраля 2009 года – Керр выложил свою систему на Google Code под индексом 1.1.1.

В сентябре 2014 года OpenCart стал самым популярным решением для интернет-коммерции в Китае, а по состоянию на август 2015 года на OpenCart работало 6,42% всех интернет-магазинов мира. По этому показателю OpenCart стал третьим в мире, в след за от WordPress, WooCommerce и Magento, опередив OSCommerce, ZenCart и Shopify.

В октябре 2014 года вышла версия 2.0, основными отличиями которой от версии 1.5 стал переход на HTML5, добавился адаптивный дизайн на Twitter Bootstrap, внедрились иконки и шрифты Font Awesome, появился установщик модулей и встроенный модификатор ocmod вместо популярного стороннего модуля vqmod в предыдущих версиях, также увеличена команда разработчиков и тестировщиков.

В июне 2017 года вышла версия 3.0, главными нововведениями которой стал переход на шаблонизатор Twig, произошло внедрение внутреннего магазина дополнений, доступного прямо в административной панели, осуществлена реализация языковых префиксов для полноценной поддержки многоязычности, до версии 3.0 страницы индексировались исключительно на том языке, который указан в настройках системы как основной, даже если в системе добавлено несколько дополнительных языков.

Интерфейс административной панели системы управления содержимым OpenCart показан на рисунке 5.

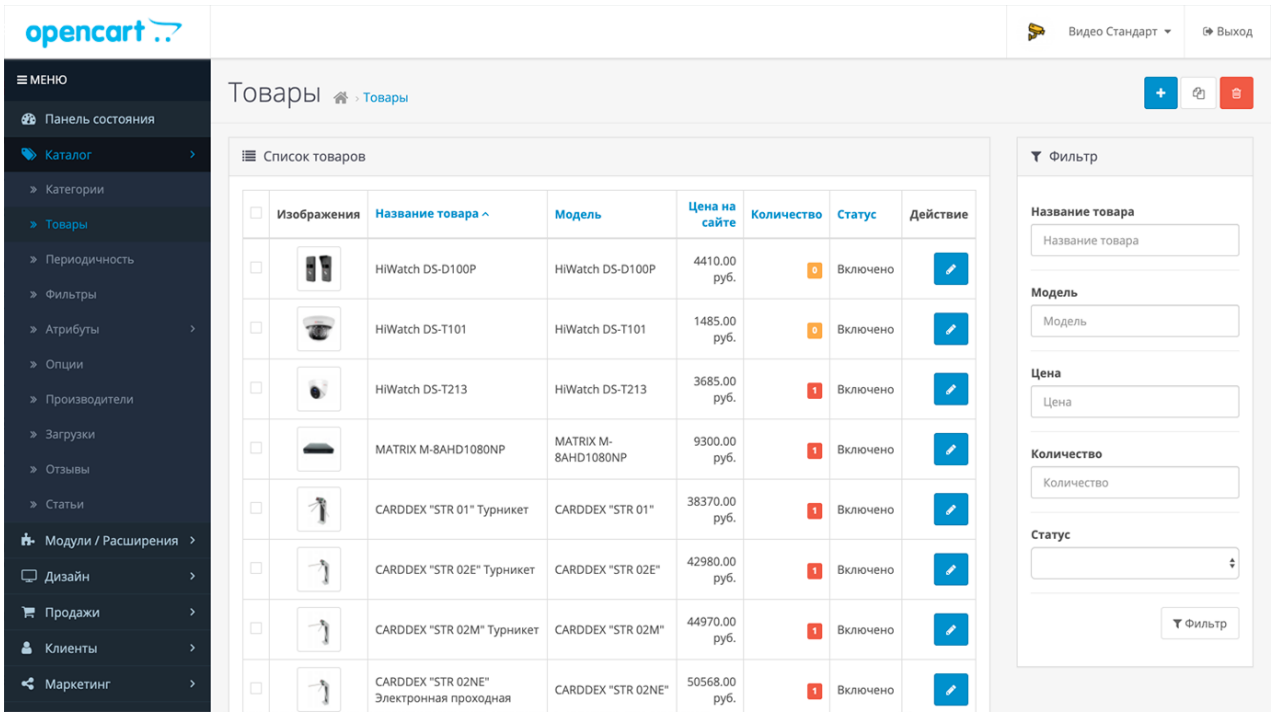


Рисунок 5 – Интерфейс административной панели OpenCart

CMS OpenCart – система управления содержимым переведён с английского на русский язык и ещё на 23 языка мира [22].

Opencart имеет открытый код и расширенные стандартные возможности для создания интернет-магазинов. Понятен даже начинающим разработчикам. Характеризуется большой базой дополнительных материалов, интегрируется с популярными сервисами оплаты и доставки покупок. Обладает собственной аналитической системой с расширенным функционалом. Основной недостаток – создание дублированных страниц. Эта ошибка устраняется только после инсталлирования платных расширений.

2.5 CMS.S3

Megagroup CMS.S3 профессиональная система управления сайтами, на которой успешно функционируют более 60000 сайтов. Система позволяет управлять содержанием и структурой сайта, осуществлять работы по продвижению и оптимизации. Любой, даже незнакомый с информационными технологиями пользователь, может совершенно самостоятельно управлять содержанием своего сайта, не прибегая при этом к услугам технического персонала [23].

Модули системы могут быть оптимизированы и настроены под конкретный проект. Это позволяет создавать сайты любой сложности: от простого лендинга до крупного интернет-представительства компании с максимумом функций.

CMS.S3 – коммерческая студийная система управления сайтом, разрабатываемая российской веб-студией «Мегагрупп.ру» с 2006 года. Система написана на языке PHP, использует Percona DB на основе MySQL в качестве базы данных. CMS позволяет пользователю управлять сайтом, не обладая специальными знаниями в веб-дизайне и программировании [6].

Интерфейс, как и у аналогов, рассмотренных выше, максимально понятен и приятен, он показан на рисунке 6.

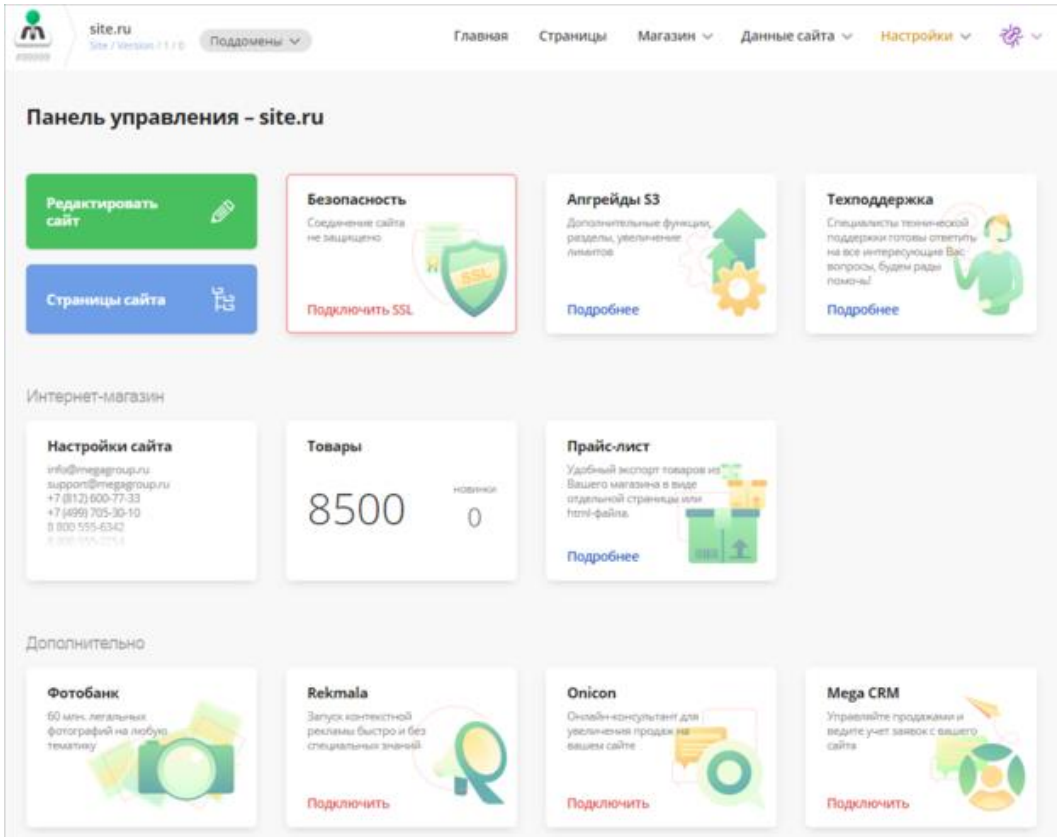


Рисунок 6 – Интерфейс административной панели CMS.S3

CMS.S3 – это третье поколение систем управления сайтом, разработанных веб-студией Мегагрупп.ру. Данная система включает в себя более 50 инструментов для редактирования содержимого сайта, рекламного продвижения, также защиты проектов, дополнительно присутствуют инструменты коммуникации. Распространяется по лицензии SaaS.

3 Проблемы безопасности современных CMS

В современном мире в интернете с каждым днем появляется все больше проектов, приложений, сайтов и т.д. Причиной востребованности данным направлением является развитие малого и крупного бизнеса, который не может существовать без веб-поддержки, рекламы и постоянной адаптации под текущее положение дел в стране и мире.

Для большинства развивающихся компаний нанять разработчика для разработки своего интернет-ресурса с «нуля» с наличием «административной панели» достаточно дорого. По этой причине огромную востребованность стали набирать CMS системы, позволяющие пользователю самому создать сайт на примере существующего шаблона или отдельные необходимые части. С появлением и развитием CMS появились хакеры и взломщики этого же направления, поэтому тема проблемы защиты ведущих CMS систем очень актуальна.

Joomla – одна из самых популярных CMS, к ней приковано внимание большого числа злоумышленников. За время существования системы, в Joomla было найдено более чем 1280 уязвимостей, согласно базе данных общеизвестных уязвимостей информационной безопасности «CVE». Большинство из них были достаточно серьезными и приводили к утечке данных неавторизованных пользователей, возможности выполнения SQL-инъекций, а также возможности получений комбинаций имени пользователя и пароля, сброс в начальное состояние, а также способность повышения своих привилегий. Большинство этих уязвимостей были исправлены в последующих версиях Joomla. Однако рассмотрим некоторые из них, представляющие интерес.

Уязвимость в компоненте EQ Event Calendar, позволявшая злоумышленнику удаленно выполнить SQL-инъекцию. Проблема заключается в том, что в обработчике поля ID не была выполнена фильтрация данных. На

данный момент не известно, принимались ли какие-то изменения в Joomla для исправления данной проблемы.

Единственная уязвимость за последние 3 года, позволяла модифицировать данные существующего пользователя. Также использовалась для сброса имени пользователя, пароля и группы. Проблема была исправлена в версии 3.6.4, однако до сих пор активно используется против сайтов, не обновивших версию CMS Joomla.

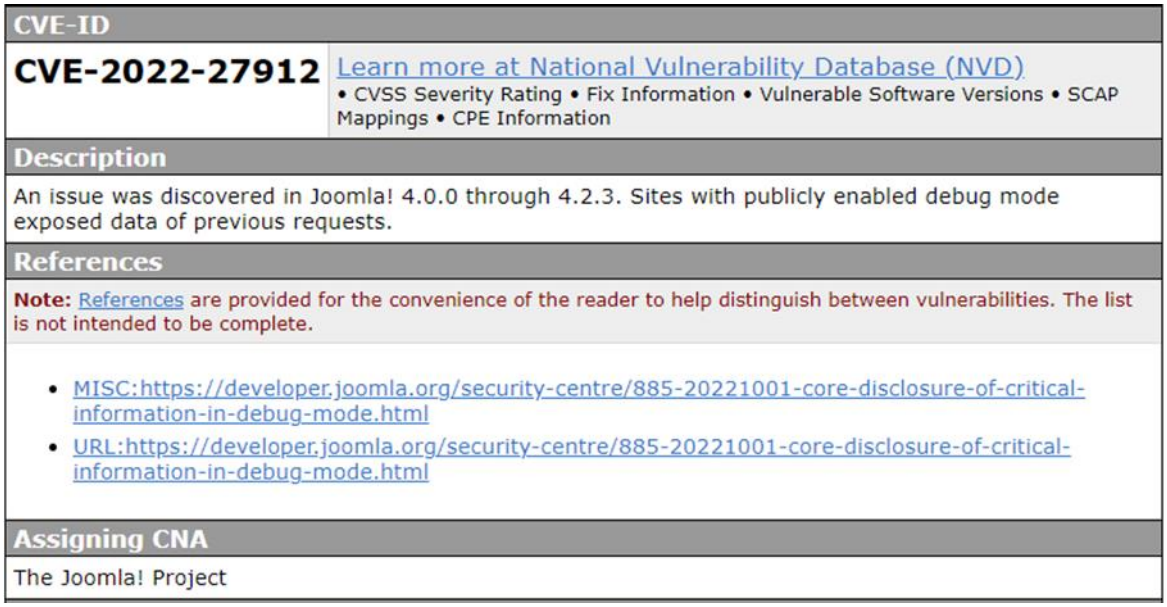
Еще одна уязвимость, затрагивающая Joomla, начиная с версии 1.7.3, выпущенной еще в 2011 году, заключалась в некорректной инвалидации кэша, приводившей к утечке содержимого форм. Исправлена в версии 3.7.2.

Также уязвимость в инсталляторе Joomla, которая не проверяла принадлежность webspace пользователю, что позволяло удаленному пользователю получить контроль над приложением. Исправлено в версии 3.7.4.

И последняя уязвимость, позволявшая пользователю обойти двухфакторную авторизацию. Проблема, показывающая, что двухфакторная авторизация не является гарантией полной защищенности пользовательских данных.

Так как Joomla является основной рассматриваемой системой контроля версий в данной работе, то дополнительно разберем еще пару примеров выявленных и запротоколированных уязвимостей, обратимся вновь к базе данных общеизвестных уязвимостей информационной безопасности «CVE».

Первый пример – это уязвимость от 25 марта 2022 года. Обнаружена проблема в версиях Joomla с 4.0.0 по 4.2.3. Сайты с общедоступным режимом отладки раскрывали данные предыдущих запросов. Данная уязвимость была устранена с последующим обновлением системы версии 4.2.4. Отчет по уязвимости в базе «CVE» выглядит согласно рисунку 7, в нем присутствует дата протоколирования уязвимости, описание, прочие подробности, также версия системы.



CVE-ID

CVE-2022-27912 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

An issue was discovered in Joomla! 4.0.0 through 4.2.3. Sites with publicly enabled debug mode exposed data of previous requests.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

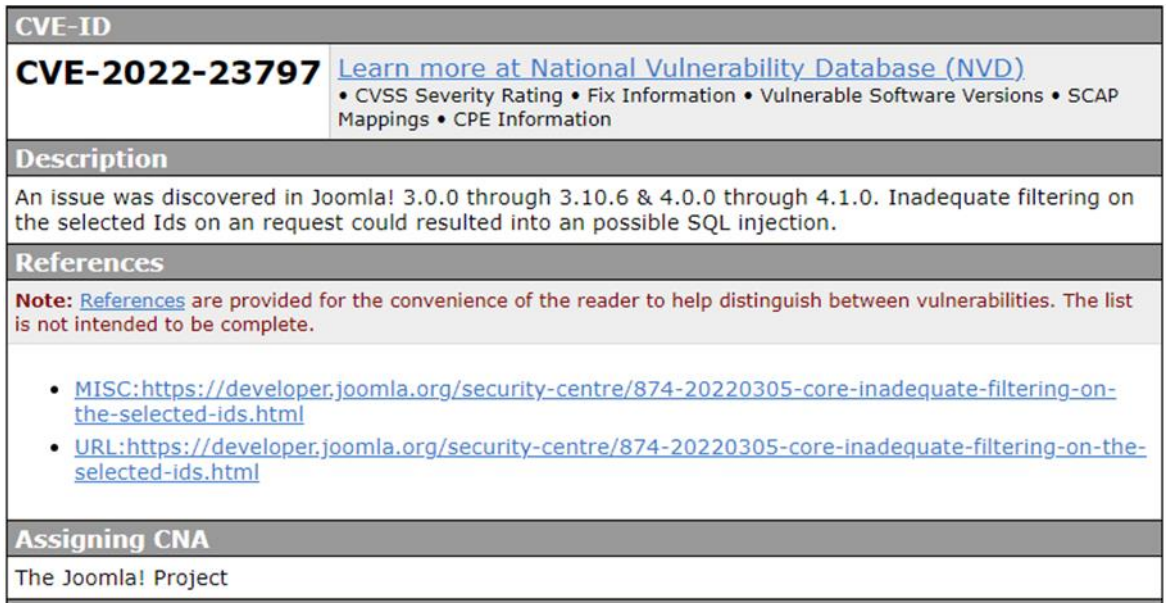
- MISC:https://developer.joomla.org/security-centre/885-20221001-core-disclosure-of-critical-information-in-debug-mode.html
- URL:https://developer.joomla.org/security-centre/885-20221001-core-disclosure-of-critical-information-in-debug-mode.html

Assigning CNA

The Joomla! Project

Рисунок 7 – Уязвимость от 25 марта 2022 года

И второй пример, уязвимость от 20 января 2022 года. Обнаружена проблема в версиях Joomla с 3.0.0 по 3.10.6 и с 4.0.0 по 4.1.0. Неадекватная фильтрация выбранных идентификаторов в запросе приводила к возможной инъекции SQL. Уязвимость также была устранена с последующим обновлением. С отчетом по данной уязвимости в базе «CVE» можно ознакомиться на рисунке 8.



CVE-ID

CVE-2022-23797 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

An issue was discovered in Joomla! 3.0.0 through 3.10.6 & 4.0.0 through 4.1.0. Inadequate filtering on the selected Ids on an request could resulted into an possible SQL injection.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:https://developer.joomla.org/security-centre/874-20220305-core-inadequate-filtering-on-the-selected-ids.html
- URL:https://developer.joomla.org/security-centre/874-20220305-core-inadequate-filtering-on-the-selected-ids.html

Assigning CNA

The Joomla! Project

Рисунок 8 – Уязвимость от 20 января 2022 года

Большинство уязвимостей исправляются своевременно после их обнаружения. Главная рекомендация пользователю – обновлять свою CMS своевременно [13].

Далее рассмотрим программные продукты компании 1С, которые являются негласным стандартом для работы бухгалтерского, управленческого и других видов учета в малом и среднем бизнесе. Многие работодатели требуют от своих сотрудников обязательных знаний и навыков работы именно с этим программным продуктом. В современном мире любой процесс автоматизации малого и среднего бизнеса начинается с продуктов 1С и продолжается доработкой необходимых блоков. Для каждой компании и или фирмы они могут быть абсолютно разными от простых отчетов в несколько строк, до целых отдельных блоков в зависимости от отрасли, где используется программный комплекс. В CMS Bitrix, используемых на данный момент можно выделить следующие проблемы безопасности:

– большое количество XSS. Административный раздел «Дополнительные поля» стал самым уязвимым местом для XSS атак непостоянного характера. Данный раздел позволяет создавать различные поля для пользователей. При их создании самыми уязвимыми стали конструкции для создания типов данных «Список» и «Видео»;

– CSRF атака. Допустимость принятия CSRF токенов, как при настройке пользователей, так и при настройке аккаунта администратора системы. Например, при смене пароля, или иных учетных данных администратора, данные отправляются на обработчик, как показано на рисунке 9.

Для взлома сайта на CMS 1С-Битрикс, пользуясь XSS и CSRF, достаточно сделать направлением атаки запрос, который, например, поменяет учетные данные доступа администратора сайта, или добавит нового, созданного атакующим. Использование XSS атаки, когда ее исполнение четко проработано, гарантирует взлом практически любого сайта, работающего под управлением

CMS 1С-Битрикс последних версий. Использование XSS атаки вместе с CSRF позволяет:

– менять учетные данные пользователей;

– создавать новый пользователей сайта, с возможностью приобретения различных привилегий;

– заменять привилегии пользователям сайта.

```
POST /bitrix/admin/user_edit.php?ID=1&lang=ru HTTP/1.1
Host: 1071lab.bitrixlabs.ru
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://1071lab.bitrixlabs.ru/bitrix/admin/user_edit.php?lang=ru&ID=1
Cookie: PHPSESSID=fdtc1nha7vd6fsgq9spuih3na0; BITRIX_SM_SOUND_LOGIN_PLAYED=Y; BITRIX_SM_GUEST_ID=1; BITRIX_SM_LAST_VISIT=13.01.2017+08%3A14%3A53;
BITRIX_SM_SALE_UID=a44218257184b130c660695f7132ea02;
BITRIX_CONVERSION_CONTEXT_s1=%7B%22ID%22%3Anull%2C%22EXPIRE%22%3A1484351940%2C%22UNIQUE%22%3A%5B%22sale_payment_add_day%22%5D%7D
Connection: keep-alive
Content-Type: multipart/form-data; boundary=---------------------------81949277201
Content-Length: 9588
```

Рисунок 9 – Пример кода

В некоторых частных ситуациях, особенно для ресурсов с недостаточным уровнем защиты на уровне сервера, возможно использование CSRF без XSS. К тому же, упомянутая атака делает обычную XSS максимальной угрозой безопасности сайта.

Wordpress – одна из самых распространенных CMS систем в сети и составляет более трети всех существующих сайтов. Именно поэтому данная система является причиной большой заинтересованности среди хакеров и взломщиков. Проблема безопасности в Wordpress с каждым годом принимает все большую актуальность. Данная CMS система имеет свои уязвимости. Большинство из них идентичны с другими схожими системами, но все же стоит рассмотреть самые ключевые из них:

– SQLI – данная уязвимость заключается в выполнении SQL-запроса

на URL-адресе атакуемого сайта;

– XSS – возможность хакера вводить необходимый код в сайт через поля ввода и другие подобные поля;

– полный перебор – подбор взломщиком информации об имени и пароле администратора сайта;

– DOS – атака сайта постоянным потоком трафика с вредоносного адреса и в результате потеря его работоспособности;

– DDOS – похож на DOS, отличается только тем, что вредоносный поток исходит из множества источников;

– Open Redirect – внедрение вредоносного кода на сайт, который осуществляет множественный переход по различным нежелательным URL-адресам;

– фишинг – кража личных данных у пользователей по средствам создания копии сайта;

– LFI – контроль злоумышленником выполнения частей кода.

Нужно учитывать, что данные проблемы составляют только часть в вопросах безопасности Wordpress. Немаловажным фактором до сих пор остается человеческие ошибки при создании сайта на данной системе, т.е. недостаточная проработка вопросов организации функционирования системы в виду наличия уязвимых мест. Так же проблемы с безопасностью зачастую связаны с несвоевременным обновлением компонентов на актуальные версии [14].

В платформе Magento существуют две значимые проблемы безопасности, первая дает возможность отобразить произвольный JavaScript код, добавленный через форму регистрации путем задания в поле с email-почтой. Выполнение кода в контексте интерфейса администратора атакующий может перехватить куки сеанса и получить доступ к сайту;

Вторая позволяет подставить JavaScript код в комментарий к заказу при использовании модуля PayFlow Pro, в дальнейшем данный код будет выполнен

при просмотре администратором списка заказов. На рисунке 10 можем увидеть пример уязвимости.

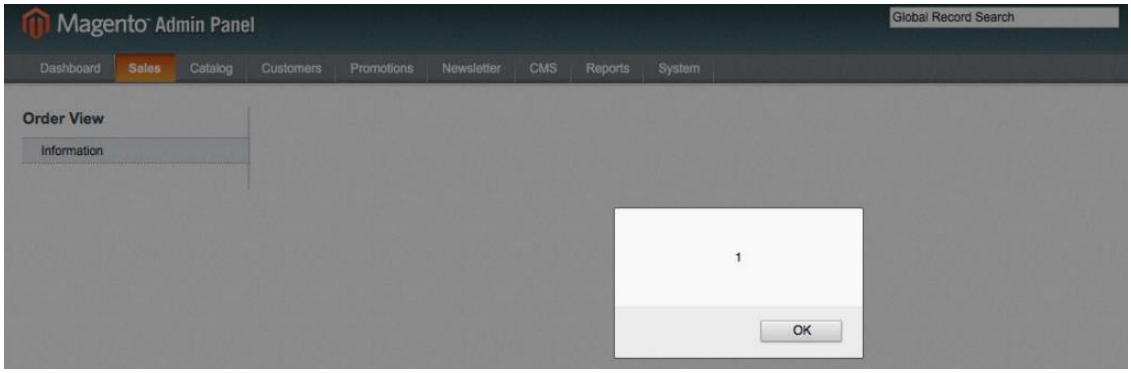


Рисунок 10 – Пример уязвимости Magento

Для устранения подобных уязвимостей разработчики Magento выпускают обновления, чтобы повысить уровень защищенности и устранить появляющиеся проблемы безопасности.

4 Сравнительный анализ существующих программных решений

Joomla – система управления содержимым с открытым исходным кодом. Именно отсутствие оплаты за использование сделало продукт таким популярным. Готовое решение помогает людям работать с контентом без глубоких знаний программирования. Кроме этого:

– доступна модификация сайта под свои нужды. Открытый код легко дополняется.

– большое разнообразие шаблонов. Можно выбрать подходящий дизайн и расположение элементов для адаптации под бренд;

– наличие компонентов и модулей для интеграции. Легко подключить платежи;

– с помощью CMS удобно наполнять сайт. Для этого не понадобится изучать инструкции – все интуитивно понятно;

– много модулей и расширений русифицированы. Их выбор, настройка, установка и управление не требуют перевода;

– совместима с большинством хостингов [8].

Но, у данной CMS есть также и отрицательные стороны. Главной уязвимостью считается слабая защищенность системы и частые случаи взлома сайтов. Разработка сайтов на Joomla предполагает использование расширений, которые открывают доступ к увеличенному функционалу. Считается, что проблема безопасности связана именно с расширениями. Поэтому следует серьезно отнестись к их выбору и установке [3].

Существует большое количество плагинов, позволяющих обезопасить сайты, разрабатываемые на Joomla, однако по-настоящему эффективных довольно мало. Рассмотрим существующие аналоги разрабатываемого модуля для сканирования файлов сайта, разрабатываемого на Joomla.

4.1 Antivirus Website Protection

Первый аналог – Antivirus Website Protection (AWP). Это плагин безопасности, предназначенный для предотвращения/обнаружения и удаления вредоносных файлов и подозрительных документов. Плагин обнаруживает бэкдоры, руткиты, троянских коней, червей, рекламу, программы-шпионы и так далее. AWP просматривает не только файлы шаблонов, а также просматривает и анализирует все файлы сайта, даже если это не часть файлов ядра Joomla [1].

Antivirus Website Protection будет особенно полезен для всех, кто загружает шаблоны и расширения с торрентов и сайтов со свободным доступом вместо покупки оригинальных копий у разработчиков на официальных источниках. Интерфейс AWP показан на рисунке 11.

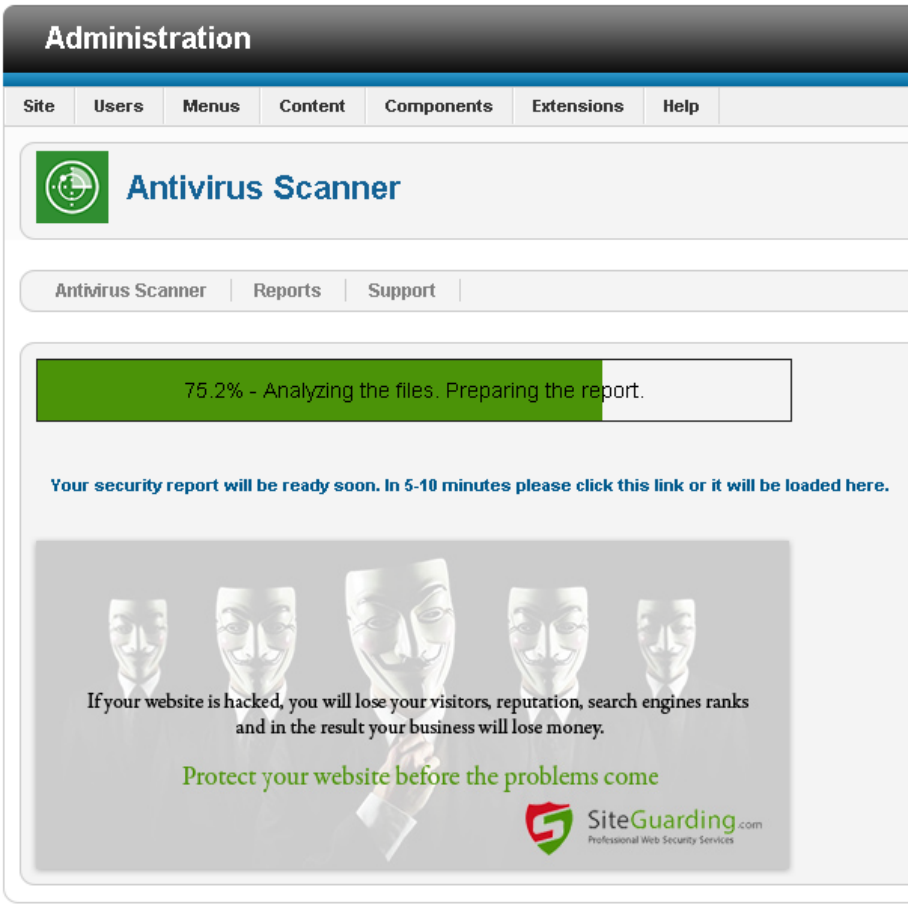


Рисунок 11 – Интерфейс Antivirus Website Protection

Основные особенности плагина:

– глубокий анализ каждого файла на сайте;

– своевременное обновление вирусной базы данных;

– возможность эвристического анализа;

– тревоги и уведомления в разделе администрирования и по электронной почте;

– сканер может обнаружить широкий список вредоносных типов объектов;

– онлайн просмотр отчетов безопасности.

Список вредоносных объектов, которые может обнаружить сканер:

– MySQL и JavaScript вставки;

– Deface – тип хакерской атаки, при которой главная страница веб-сайта заменяется на другую – как правило, вызывающего вида. Зачастую доступ ко всему остальному сайту блокируется, или же прежнее содержимое сайта вовсе удаляется;

– скрытые iframes – тип хакерской атаки, при которой хакер получает доступ к сетевому протоколу FTP (File Transfer Protocol) сайта и настраивает скрытый iframe, тем самым заражая компьютеры посетителей интернет-ресурса;

– PHPMailer – тип хакерской атаки, при которой хакер использует сайт для рассылки электронных писем со спамом от имени интернет-ресурса;

– социальная инженерия или «атака на человека»;

– фишинг – вид интернет-мошенничества, целью которого является получение доступа к конфиденциальным данным пользователей;

– перенаправления;

– бэкдоры;

– XSS;

– руткиты и другие варианты этого типа вредоносного ПО;

– троянские кони;

– интернет-черви.

Первая версия Antivirus Website Protection была добавлена в официальный магазин расширений Joomla 19 ноября 2014 года, на текущий момент версия данного ПО 5.3.1, последнее обновление выходило в марте 2022 года, совместима с Joomla 3. AWP также является самым распространенным модулем защиты системы управления содержимым Joomla, для обеспечения дополнительной безопасности сайта.

4.2 Website Antivirus Scanner

Рассмотрим еще один аналог – Website Antivirus Scanner (WAS). Это расширение безопасности для обнаружения вирусов и подозрительных кодов, SQL-инъекций и попыток взлома.

Помимо обеспечения расширенной защиты от вредоносных программ для сайтов, расширение также может обнаруживать хакерские атаки в процессе функционирования сайта, блокировать их и автоматически предотвращать его возможный взлом [2].

Дополнительно, WAS проверяет, обнаруживает и удаляет не только вредоносное ПО, заражающее сайт, но и диагностирует все возможные инъекции вредоносного кода, которые оказывают какое-либо негативное воздействие на него. Интерфейс Website Antivirus Scanner показан на рисунке 12.

Расширение было разработано SafetyBis, это кипрская компания по разработке веб-приложений с более чем 10-летним опытом создания различных типов решений B2B и B2C для бизнеса любых масштабов. SafetyBis предоставляет полный спектр услуг по разработке веб-приложений в Европе для компаний, стремящихся автоматизировать бизнес-процессы, интегрировать новые функции, снизить рабочую нагрузку и создать новые продукты или услуги SAAS.

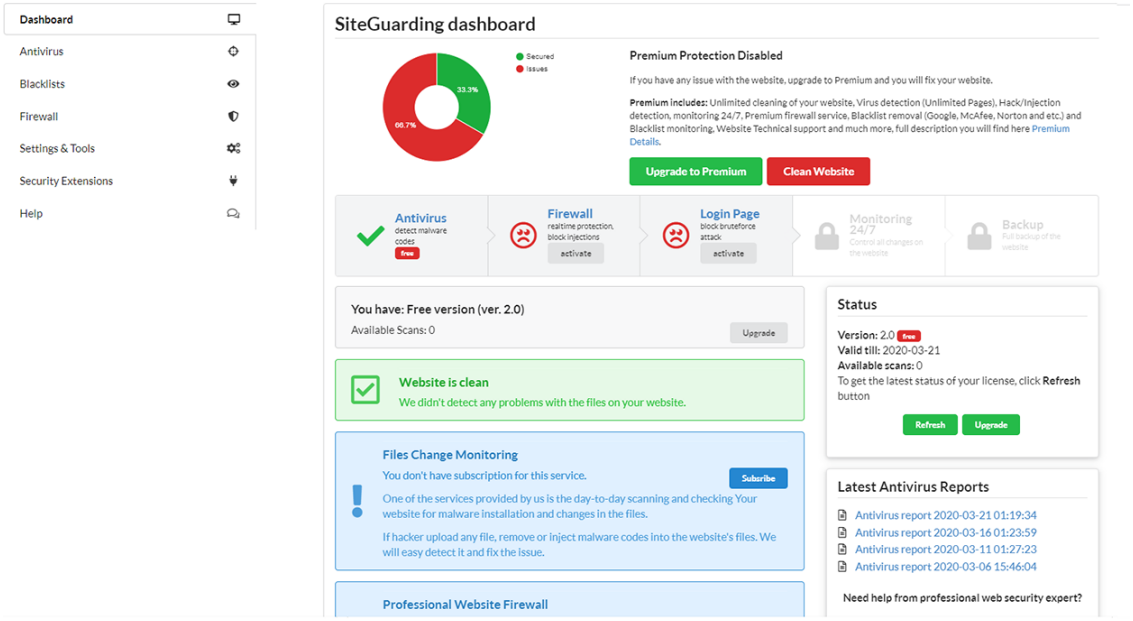


Рисунок 12 – Интерфейс Website Antivirus Scanner

Первая версия Website Antivirus Scanner была добавлена в официальный магазин расширений Joomla 30 апреля 2018 года, на текущий момент версия данного ПО 2.3.1, последнее обновление выходило в марте 2022 года, совместима с Joomla 3.

4.3 VirusTotal

Третий аналог – это уже не расширение для системы управления содержимым Joomla, а бесплатная служба, которая осуществляет анализ подозрительных файлов и ссылок на предмет выявления вирусов, троянов, червей и всевозможных вредоносных программ – VirusTotal. Данная служба была награждена американским изданием PC World Magazine как один из 100 лучших продуктов 2007 года. Имеет корректный перевод на многие языки мира, включая русский. Также, сервис полностью бесплатный.

VirusTotal использует антивирусные технологии более чем 70 сканеров, среди которых можно отменить самые популярные, это McAfee, Kaspersky,

Microsoft, DrWeb, Avast и AVware. Однако результаты проверок файлов сервисом не зависят от какого-то одного производителя антивирусов.

Антивирусы на VirusTotal не дают гарантию отсутствия вредоносного кода в файле, и не гарантируют присутствие вредоносного кода в файле, так как чёткие критерии, по которым программные продукты и файлы могут быть отнесены к категории вредоносных программ, до настоящего времени нигде чётко не оговорены.

У компаний-разработчиков антивирусного программного обеспечения есть собственные классификации и номенклатуры вредоносного ПО, поэтому при проверке файла на антивирусы на VirusTotal могут выдавать разные результаты, например, одни антивирусы посчитают файл потенциально опасным, а другие – безопасным.

Все используемые сервисом антивирусные базы постоянно обновляются. В результатах проверки указываются даты последних обновлений всех баз.

Система, после загрузки файла, вычисляет его хеш и при наличии результатов проверки файла с таким же хешем предлагает просмотреть последний анализ, при этом указав дату первой и последней проверки, либо повторить анализ [19].

Сервис постоянно развивается, что доказывает постоянное подключение новых сканеров, антивирусников и антитроянов. VirusTotal отсылает потенциально опасные и подозрительные файлы производителям антивирусов на анализ.

7 сентября 2012 года в блоге сайта было объявлено о приобретении сервиса компанией Google.

В январе 2018 года сервис стал частью Chronicle, новой компании Alphabet, специализирующейся на кибербезопасности [20].

Интерфейс VirusTotal показан на рисунке 13, как видим, доступна проверка как одного файла, так и полное сканирование сайта по ссылке.

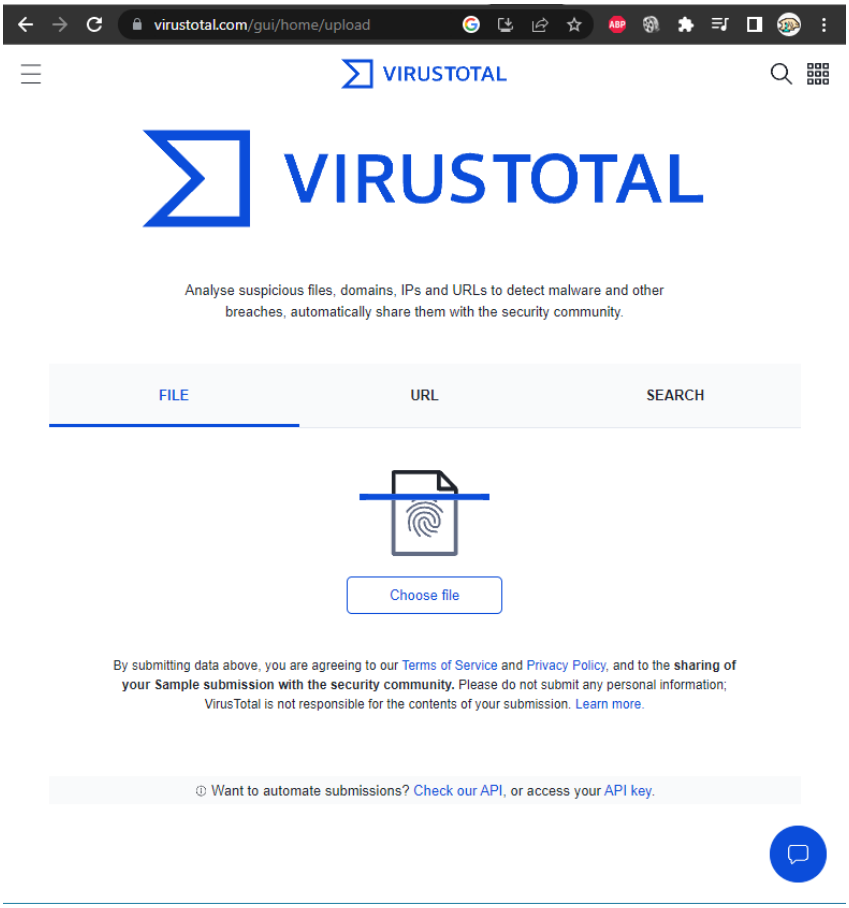


Рисунок 13 – Интерфейс VirusTotal

Стоит отметить, что сервис не заменит антивирус на компьютере, локально, поскольку проверяются только отдельные файлы и отдельные URL-адреса по требованию. Сервис также не обеспечивает постоянной защиты на компьютере пользователя и является лишь дополнением к установленному антивирусу. Хотя сервис и использует несколько антивирусных движков, результат антивирусов не гарантирует безвредности файла или URL-ссылки. Максимальный размер загружаемого файла ограничен 650 мегабайтами.

VirusTotal не предназначен для сравнения антивирусов, т.к. антивирусные движки, используемые на VirusTotal, являются консольными версиями, поэтому в зависимости от продукта они не будут вести себя точно так же, как их аналоги для настольных компьютеров. Например, версия антивируса для настольного компьютера может также использовать проактивную защиту и брандмауэр,

которые повышают вероятность обнаружения угроз. Также, в консольных версиях антивирусов на VirusTotal эвристический анализ может быть более агрессивным и параноидальным по сравнению с антивирусами для настольных компьютеров, поэтому на VirusTotal ложных срабатываний может быть больше.

К недостаткам сервиса можно отнести то, что средств и возможностей для анализа статистики, которую выдает сервис после проверки.

Также к минусам можно отнести отсутствие гарантии о безвредности или опасности файла, как было рассмотрено выше. Один и тот же файл некоторые антивирусные сканеры могут определить как троян, а другие как безвредный файл, что демонстрирует пример на рисунке 14.

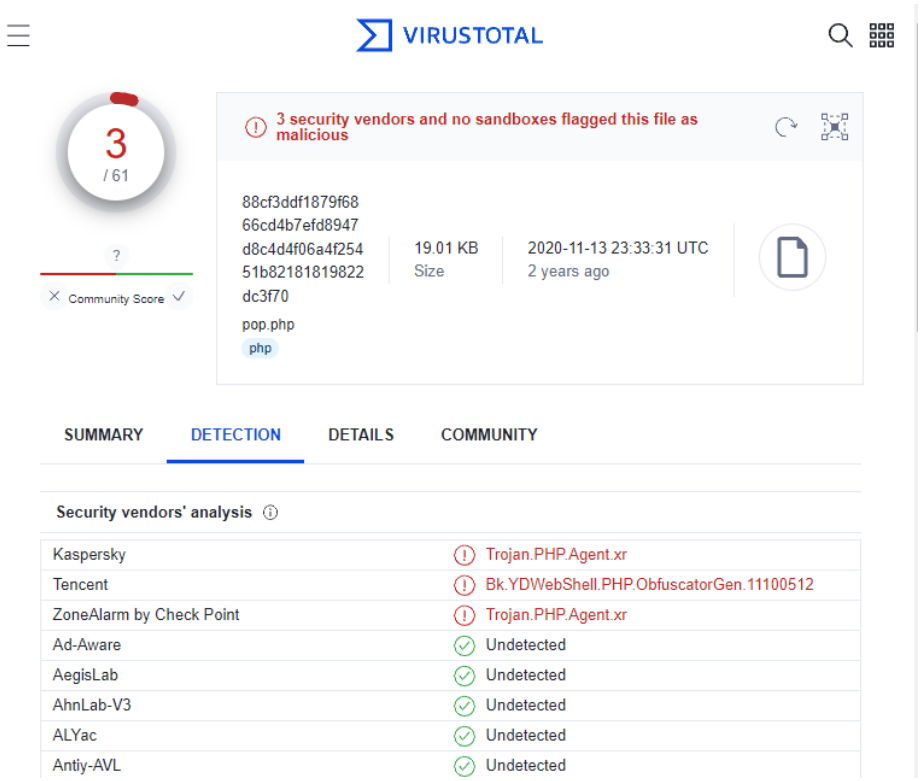


Рисунок 14 – Результаты проверки VirusTotal

Последний существенный недостаток службы – это ограничение по лимиту запросов и по размеру файла для API, как в платной версии, так и в бесплатной.

4.4 JoomScan

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. Сообщество работает над созданием статей, учебных пособий, документации, инструментов и технологий, находящихся в свободном доступе.

Фонд OWASP – это благотворительная организация, которая оказывает поддержку и осуществляет управление проектами и инфраструктурой OWASP. Кроме того, Фонд зарегистрирован как некоммерческая организация в Европе с июня 2011 года.

OWASP не осуществляет предпринимательскую деятельность ни с одной компанией, занимающейся разработкой технологий, но он поддерживает грамотное использование технологий безопасности. Проект избегает аффилирования, так как полагает, что свобода от влияния со стороны других организаций может облегчить распространение независимой, полезной и дешевой информации о безопасности приложений.

Участники сообщества OWASP делают приложения безопаснее, учитывая человеческий фактор и технологический уровень.

OWASP организовал серию конференций AppSec для дальнейшего построения сообщества, посвященного безопасности приложений.

OWASP создаёт стандарты, первый из которых был опубликован под названием OWASP Application Security Verification Standard (ASVS). Основная цель OWASP ASVS – это стандартизация диапазона охвата и уровня строгости доступных на рынке приложений, обеспечивающих безопасность. Целью ASVS также являлось создание набора коммерчески успешных открытых стандартов, приспособленных для специализированных веб-технологий.

Сборник для Веб-Приложений уже был опубликован. Сборник для Веб-Сервисов в процессе разработки.

Самыми распространёнными инструментами OWASP являются тренировочная среда, прокси-анализатор WebScarab и .NET инструменты. Также, одним из инструментов OWASP является JoomScan.

JoomScan – это сокращение от полного названия проекта компании OWASP, оно выглядит следующим образом – OWASP Joomla! Vulnerability Scanner. Это проект с открытым исходным кодом, разработанный с целью автоматизации задачи обнаружения уязвимостей и обеспечения надежности в развертываниях Joomla CMS. Реализован на Perl, это инструмент, обеспечивающий беспрепятственное и легкое сканирование установок Joomla, оставляя минимальный след благодаря своей модульной архитектуре. Он не только обнаруживает известные опасные уязвимости, но также способен обнаруживать множество неверных конфигураций и недостатков на уровне администратора, которые могут быть использованы злоумышленниками для взлома системы. Кроме того, JoomScan предоставляет удобный интерфейс и компилирует окончательные отчеты как в текстовом, так и в HTML-формате для простоты использования и сведения к минимуму накладных расходов [21].

JoomScan включен в дистрибутивы Kali Linux и имеет следующие инструменты:

- определяет версию Joomla, которая используется на сайте;
- детектор уязвимостей;
- детектор используемых компонентов;
- детектор уязвимостей компонентов;
- детектор брандмауэра;
- отчетность в виде текста или HTML;
- поиск лог-файлов;
- поиск бэкап-файлов.

JoomScan это консольный инструмент, поэтому в качестве интерфейса выступает текст, выводимый в консоль, пример можно наблюдать на рисунке 15.

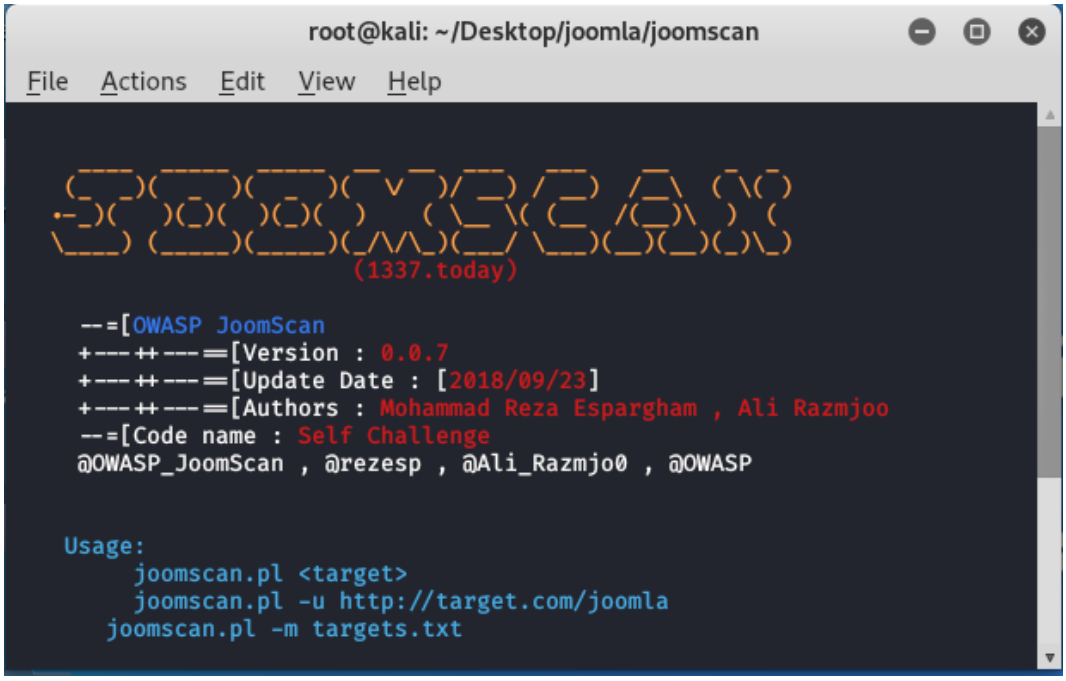


Рисунок 15 – Консоль с JoomScan

К минусу данного инструмента можно отнести отсутствие кроссплатформенности, JoomScan доступна только на Kali Linux, и отсутствие графического интерфейса, пользоваться инструментом можно только через консоль.

5 Алгоритмическая реализация программного средства

Для дальнейшей разработки программного средства была разработана оптимальная модель его работы. В основу программы входит алгоритм проверки папки, содержащей все файлы сайта, на наличие вирусных сигнатур.

На рисунке 16 представлен алгоритм сканирования файлов сайта.

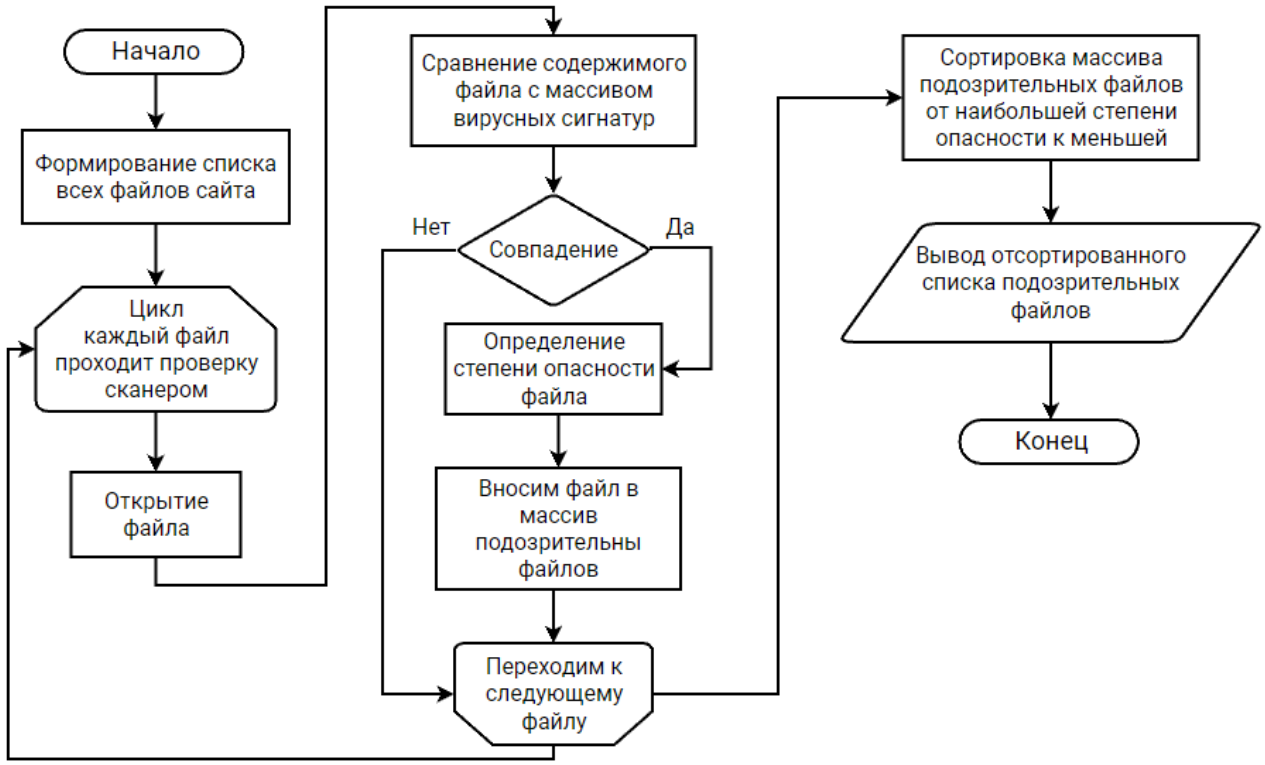


Рисунок 16 – Алгоритм сканирования файлов

Алгоритм состоит в следующем: после установки программного средства в CMS появляется кнопка Simple virus scanner в разделе Компоненты. По нажатию на данную кнопку запускается наш алгоритм. Сначала формируется список абсолютно всех файлов сайта, далее цикл перебирает каждый файл из этого списка, путем его открытия и сравнивания его содержимого с имеющейся базой вирусных сигнатур. В случае наличия совпадения определяется степень опасности файла, и далее уже вносится в список-массив (базу) подозрительных

файлов сайта. Переходим к следующему файлу, если он есть. В случае отсутствия совпадения переход к следующему файлу осуществляется сразу же.

Затем, когда все файлы были просканированы, мы имеем список-массив (базу) всех подозрительных файлов, и, следующем шагом будет сортировка данного списка от наибольшей степени опасности, который может нанести данный файл, к меньшей.

Далее, выводим отсортированный список в виде Html-отчета, на этом этапе алгоритм завершается, т.е. на выходе, как результат работы программного средства, имеем отчет с результатами сканирования.

6 Программная реализация программного средства

6.1 Средства разработки

Для написания программного средства был использован язык программирования PHP.

PHP – это язык программирования, который изначально был создан для разработки веб-приложений, но позже он эволюционировал в язык общего назначения. Также, это язык с динамической типизацией – код на нем прост для чтения и понимания, а разработка – быстрая. PHP – интерпретируемый язык. Это значит, что компьютер понимает код на PHP и воспроизводит программы без предварительного перевода в машинный код [16].

На данном языке программирования можно делать ресурсы любой сложности: от простого лендинга до социальной сети – именно на PHP написана «ВКонтакте». Код на PHP легко встраивается в классический HTML – нужно добавить соответствующий тег в разметку. Язык поддерживают все популярные операционные системы: Windows, macOS, Linux, UNIX и не только.

Также PHP позволяет работать с такими веб-серверами, как IIS в Windows и Apache в macOS и Linux. Благодаря такой широкой совместимости у разработчиков практически нет ограничений в выборе веб-сервера и операционной системы.

Разработчиком языка программирования PHP является датский программист Расмус Лердорф. В 1994 году он создал набор скриптов на языке Perl – знаменитую «персональную домашнюю страницу», которая легла в основу PHP. Написанные им скрипты позволяли вести статистику просмотров его резюме на сайте. После того как Расмус понял, что быстроты и функциональности не хватает, он написал новый интерпретатор на языке Си.

В 1995 году Лердорф сделал код открытым, что позволило разработчикам

всего мира подключиться к улучшению, исправления ошибок языка [17].

PHP – один из старейших языков, который разрабатывается силами сообщества по модели Open source. Сейчас его поддерживает и разрабатывает группа энтузиастов во главе с компанией Zend Technologies. Компанией руководят Зеев Сураски и Энди Гутманс, в 1997 году выпустившие третью версию PHP [18].

В качестве среды разработки был использован Visual Studio Code.

Visual Studio Code (VS Code) – текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

Visual Studio Code основан на Electron и реализуется через веб-редактор Monaco, разработанный для Visual Studio Online.

VS Code был анонсирован 29 апреля 2015 года компанией Microsoft на конференции Build, и вскоре была выпущена бета-версия.

18 ноября 2015 года Visual Studio Code был выпущен под лицензией MIT, исходный код был опубликован на GitHub, также анонсирована поддержка расширений, а 14 апреля 2016 года редактор вышел из стадии бета-тестирования.

В ходе разработки программного средства также были использованы такие расширения для VS Code как EditorConfig for VS Code, PHP Intelephense, PHP Debug, PHP Sniffer, Semicolon Insertion Shortcut.

EditorConfig – это конфигурационный файл и набор расширений к большому количеству редакторов кода. Он берёт настройки из файла

editorconfig, который, как правило, размещается в корне проекта. Расширение автоматически настроит отступы и перевод строк единообразно для всех разработчиков, использующих его. PHP код чаще всего выполняется на *nix системах, поэтому необходимо использовать стандарт.

В редакторе уже есть поддержка синтаксиса и подсказок стандартных функций языка. Но без специального дополнения редактор не будет подсказывать пользовательские функции из других частей проекта. Поэтому для поддержки автодополнения, анализа кода, перехода к месту, где создана функция, класс или переменная (с помощью сочетания клавиш Alt+Click), используется дополнение PHP Intelephense.

При разработке может возникнуть ситуация, когда простых функций отладки и логирования становится недостаточно. Тогда может помочь специальный инструмент дебаггер. Для PHP есть расширение xdebug (PHP Debug), которое позволяет расставить точки останова и посмотреть окружение в предполагаемом месте ошибки, выполняя код поэтапно либо до следующей точки.

В языках программирования есть понятие стиль кодирования. Но не все разработчики знают об этом. Программа, которая отвечает за проверку на соответствие стандартам, называется линтер. В PHP в качестве линтера используется PHP CodeSniffer. Нас интересуют стандарты PSR-1 и PSR-12. Именно эти два стандарта касаются кодирования и правил оформления.

Расширение Semicolon Insertion Shortcut добавляет необходимый символ в конец строки с помощью горячих клавиш. Это необходимо, т.к. PHP требует разделять инструкции с помощью точки запятой.

Для вывода отчета с потенциальными угрозами был использован стандартизированный язык гипертекстовой разметки – HTML.

Язык гипертекстовой разметки HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986 – 1991 годах в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и

технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов – дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.

Первым общедоступным описанием HTML был документ «Теги HTML», впервые упомянутый в Интернете Тимом Бернерсом-Ли в конце 1991 года. В нём описываются 18 элементов, составляющих первоначальный, относительно простой дизайн HTML. За исключением тега гиперссылки, на них сильно повлиял SGMLguid, внутренний формат документации, основанный на стандартном обобщенном языке разметки (SGML), в CERN. Одиннадцать из этих элементов всё ещё существуют в HTML 4.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). Текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью. Однако современное применение HTML очень далеко от его изначальной задачи [15].

Также, для правильной установки плагина в CMS Joomla необходим корректно составленный XML-документ, в связи с этим в разработке дополнительно был использован расширяемый язык разметки XML, который позволяет определять и хранить данные совместно используемым способом. XML поддерживает обмен информацией между компьютерными системами, такими как веб-сайты, базы данных и сторонние приложения. Предопределенные правила упрощают передачу данных в виде XML-файлов по любой сети.

6.2 Структура программного средства

Программное средство состоит из пяти модулей, реализующих все основные функции разрабатываемой программы. Модули следующие: simplevirusscanner, provider, htmlView, displayController и default. Представим описание функций модулей.

Модуль simplevirusscanner используется для правильной настройки и обработки установщиком CMS Joomla разработанного плагина.

Модуль provider является конструктором корректного отображения и внедрения в CMS дополнительного программного обеспечения.

Модуль htmlView необходим для формирования HTML-отчета, в который будут переданы результаты сканирования подозрительных файлов из модуля default.

Модуль displayController является главным по взаимодействию с административной панелью системы управления содержимым сайта, с помощью него мы вносим изменения в панель администратора, добавляя кнопку, по нажатию на которую переходим на страницу с готовым отчетом, который был сформирован в результате правильной работы разработанного программного средства.

Модуль default является основным, т.к. в нем описан алгоритм проверки файлов на вирусы либо вирусные сигнатуры. Он состоит из следующих функций, таких как buildScanList, scanFiles, sortScanList и printScanList.

Функция buildScanList формирует список файлов, подлежащих сканированию, на основе имеющейся базы с исключениями, в виде файлов настроек администратора и других системных, необходимых для корректного функционирования CMS, файлов.

Функция scanFiles сканирует файлы на наличие вирусов, вирусных сигнатур и вредоносных скриптов.

Функция sortScanList формирует список файлов, прошедших проверку функцией scanFiles, сортируя их от наиболее потенциально опасных и подозрительных к менее.

Функция printScanFiles, непосредственно, выводит отсортированный функцией sortScanList список файлов, на которые стоит обратить внимание разработчику, администратору или владельцу сайта.

6.3 Демонстрация работы программного средства

Разработанный плагин нужно загрузить через административную панель в CMS, что можно сделать перейдя во вкладку Система, далее в разделе установки нажать на Расширения, как это должно выглядеть представлено на рисунке 17.

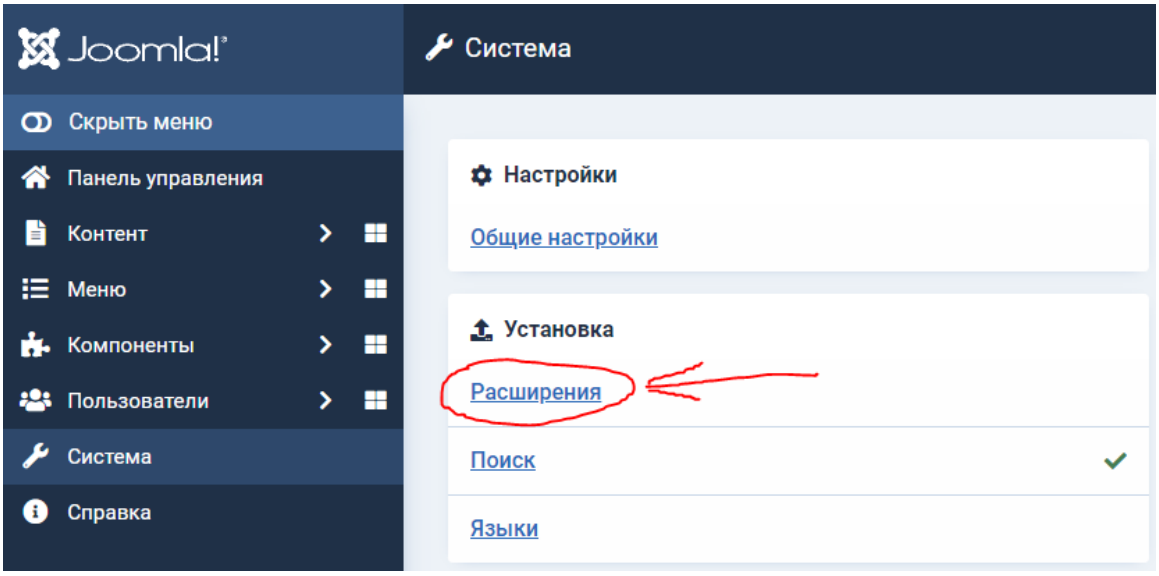


Рисунок 17 – Вкладка Система, раздел установки

Далее, в открывшемся разделе установки выбираем наше программное средство в расширении ZIP. После установки видим сообщение о том, что установка компонента успешно завершена, а также описание установленного расширения, что можно наблюдать на рисунке 18.

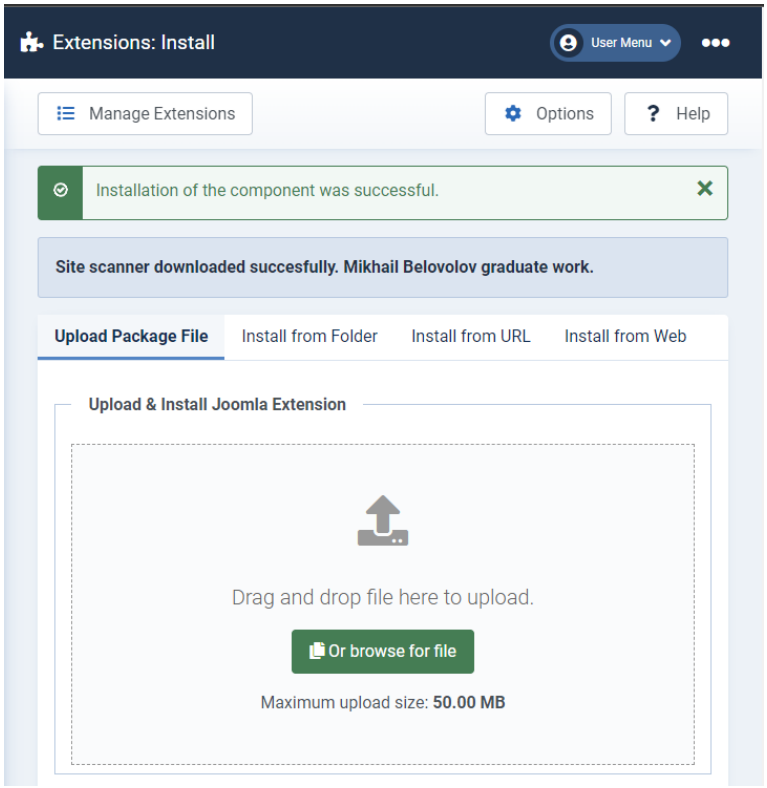


Рисунок 18 – Установка расширения

Также, в административной панели CMS, во вкладке Компоненты, должна появиться кнопка с названием нашего расширения, как это должно выглядеть показано на рисунке 19.

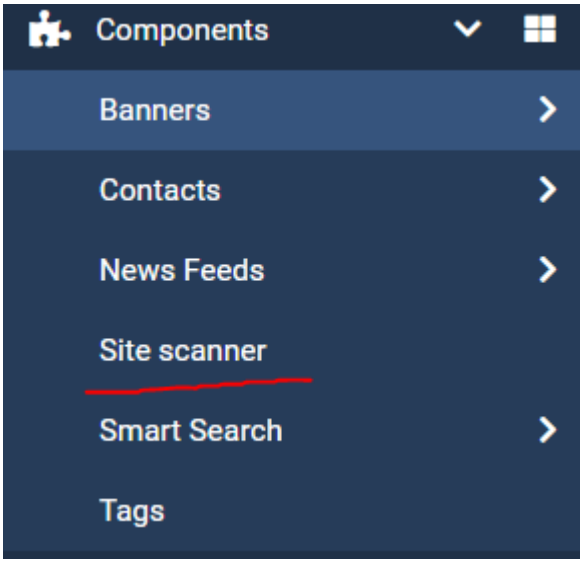


Рисунок 19 – Добавленное расширение

Дополнительно, в разделе расширений по ключевому слову «scanner» можем найти наше установленное расширение, здесь можно найти тип расширения, ее версию, среду распространения, автора и дату разработки модуля. Также, в данном разделе можно удалить расширение, либо приостановить его работу нажав на зеленую галочку. Раздел расширений можно наблюдать на рисунке 20.

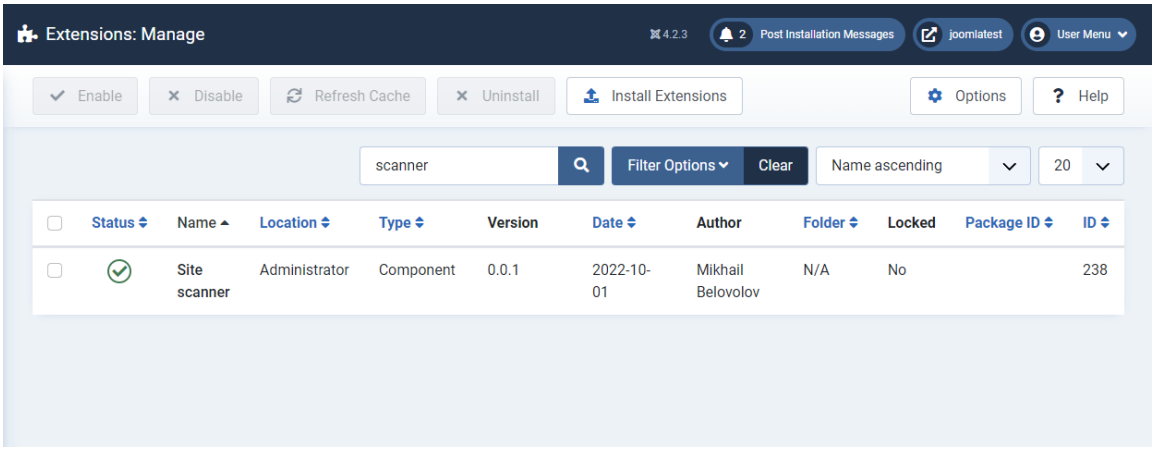


Рисунок 20 – Раздел расширений

На рисунке 21 представлен интерфейс разработанного программного средства.

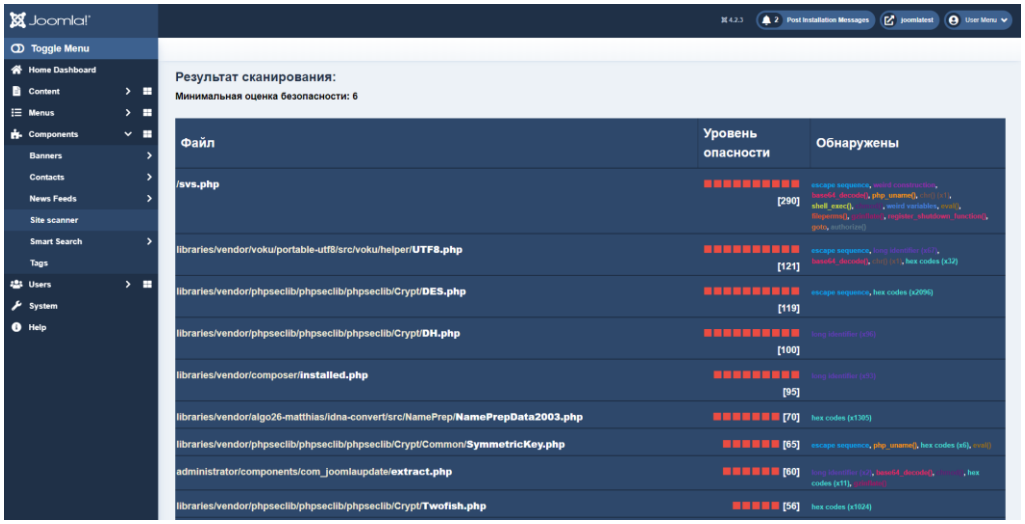


Рисунок 21 – Интерфейс программы в CMS Joomla

Интерфейс представлен в виде отчета-таблицы с тремя колонками, такими как название файла, уровень опасности и обнаруженные вирусные сигнатуры.

В конце списка располагаются файлы, путь к которым указан в массиве исключений, в который пользователь может заранее добавлять файлы, в безопасности которых уверен, но полагает, что сканер может определить его как потенциально опасным. Как выводятся исключения в Html-отчете показано на рисунке 22.

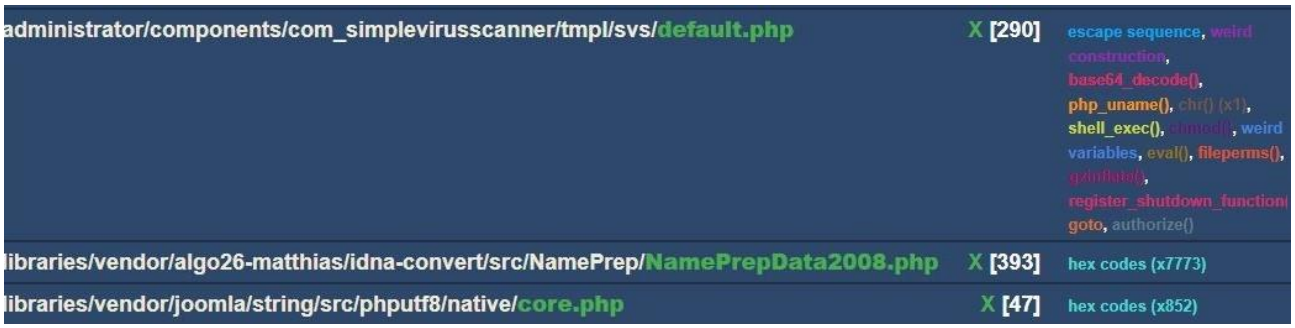


Рисунок 22 – Исключения в отчете

Также, снизу отчета указана дополнительная информация, что можно наблюдать на рисунке 23.

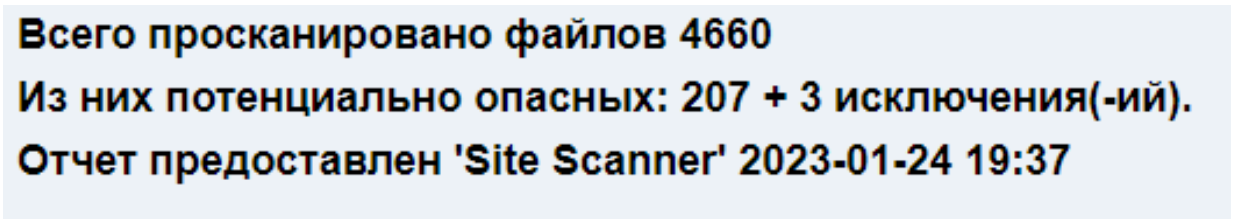


Рисунок 23 – Дополнительная информация из отчета

Среди дополнительной информации видим количество проверенных файлов, количество подозрительных файлов, количество исключений и время и дата проведения сканирования.

В случае, если удалось интегрировать расширение в систему управления содержимым CMS Joomla, это не дает гарантии, что расширение будет работать

и корректно отображаться, даже если в административной панели появилась кнопка запуска алгоритма программного средства. Пример вывода ошибки показан на рисунке 24.

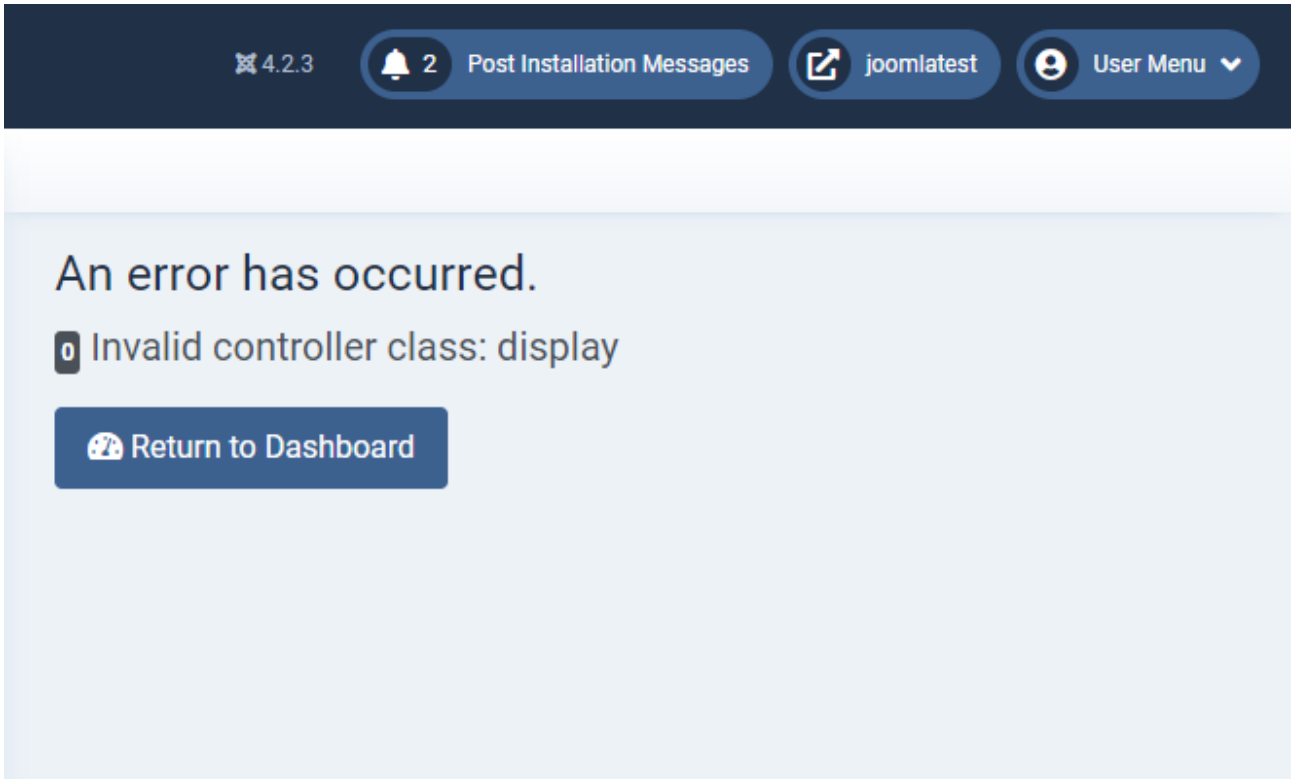


Рисунок 24 – Ошибка отображения

Данная ошибка отображения не единственная возможная ошибка при сбое или некорректной работе программного средства. Также рассмотрены другие случаи и сценарии, при возникновении которых оператор будет уведомлен подобным сообщением о типе ошибки.

Заключение

В процессе исследования и разработки выпускной квалификационной работы были определены основные системы управления содержимым и их возможные уязвимости. Рассмотрены основные методы дополнительной защиты сайтов. Проведено сравнение аналогов, в результате которого был составлен перечень основных достоинств и недостатков каждого из аналогов. Рассмотрена алгоритмическая реализация программного средства.

Также был объяснён выбор программных средств разработки программы. Продемонстрирована работа функционала разрабатываемого программного средства.

Разработанное программное средство демонстрирует свою применимость в рамках эффективной реализации сканирования файлов сайта. Разработанное программное средство представлено в виде интегрируемого в систему управления содержимым CMS Joomla расширением, которую, после успешной интеграции, можно использовать лишь по нажатию на кнопку в административной панели системы.

Поставленная цель была достигнута, модуль для дополнительной защиты сайтов, создаваемых на Joomla, был разработан.

Для достижения цели работы были выполнены следующие задачи:

– изучены теоретические основы системы управления содержимым;

– рассмотрены основные виды систем управления содержимым и их возможные уязвимости;

– рассмотрены основные методы защиты сайтов;

– проведено сравнение аналогов разработки;

– разработан модуль для дополнительной защиты сайтов, создаваемых на Joomla.

Перечень использованных информационных ресурсов

1. Antivirus Website Protection : официальный сайт. – Кипр. – Обновляется в течение суток. – URL: https://extensions.joomla.org/extension/antivirus-website-protection/ (дата обращения: 20.10.2022).

2. Website Antivirus Scanner : официальный сайт. – Кипр. – Обновляется в течение суток. – URL: https://extensions.joomla.org/extension/website-antivirus-scanner-for-joomla/ (дата обращения: 21.10.2022).

3. Joomla! : официальный сайт. – Соединенные Штаты Америки. – Обновляется в течение суток. – URL: https://joomla.ru/about (дата обращения: 19.10.2022).

4. Рынок заказной веб-разработки в 2021 году: Адаптация бизнес-модели веб-студий к потребностям заказчиков / Д.А. Федоров, Н.Н. Покровская, Д.В. Голохвастов. – Текст : электронный. – Санкт-Петербург : Известия Санкт-Петербургского государственного экономического университета : [сайт]. – 2021. – URL: https://cyberleninka.ru/article/n/rynok-zakaznoy-veb-razrabotki-v-2021-godu-adaptatsiya-biznes-modeli-veb-studiy-k-potrebnostyam-zakazchikov (дата обращения 11.10.2022).

5. iTrack – Рейтинг CMS : сайт. – Москва, 2021 – URL: https://itrack.ru/rating-cms/ (дата обращения: 25.10.2022).

6. iTrack – Рейтинг SaaS-CMS : сайт. – Москва, 2021 – URL: https://itrack.ru/rating-cms/saas/ (дата обращения: 25.10.2022).

7. iTrack – Исследование CMS : сайт. – Москва, 2022 – URL: https://itrack.ru/research/cmsrate/#!cms-overall-tab (дата обращения: 23.10.2022).

8. Колисниченко, Д.Н. Движок для вашего сайта. CMS Joomla! / Д.Н. Колисниченко. – Санкт-Петербург : БХВ-Петербург, 2018. – 368 с. – ISBN 978-5-9775-0258-0.

9. Системы управления контентом как инструментарий разработки

интернет-ресурсов / И.Д. Рудинский, М.Ю. Михайловский. – Текст : электронный // Калининград: Вестник науки и образования Северо-Запада России : [сайт]. – 2016. – URL: http://vestnik-nauki.ru (дата обращения: 13.11.2022).

10. Выбор системы управления контентом / В.В. Таборовец, Д.В. Богумил. – Текст : электронный. – Минск: Наука, техника и образование. – 2018. – URL: https://cyberleninka.ru/article/n/vybor-sistemy-upravleniya-kontentom (дата обращения 7.10.2022).

11. Анализ защищенности современных систем управления контентом / М.Л. Лопатин, О.Ю. Пескова. – Текст : электронный. – Таганрог : Известия Южного федерального университета : [сайт]. – 2017. – URL: https://cyberleninka.ru/article/n/analiz-zaschischennosti-sovremennyh-sistem-upravleniya-kontentom (дата обращения 10.10.2022).

12. 1С-Битрикс! : официальный сайт. – Россия. – Обновляется в течение суток. – URL: https://www.1c-bitrix.ru/about/ (дата обращения: 19.10.2022).

13. Проблемы безопасности современных CMS / О.М. Бакунова, В.А. Высоких, В.А. Кузнецов, А.В. Матусевич, А.Ю. Иваньков. – Текст : электронный // Elibrary: [сайт]. – 2018. – URL: https://www.elibrary.ru/item.asp?id=32334125 (дата обращения: 12.12.2022).

14. Канаван, Т. Справочник по безопасности CMS: полное руководство для WordPress, Joomla, Drupal, и Plone / Т. Канаван. – США: Wiley, 2018. – 432 с. – ISBN 978-0470916216.

15. Изучаем HTML, XHTML и CSS / Э. Робсон, Э. Фримен. – Санкт-Петербург: Питер, 2018. – 720 с. – ISBN 978-5-4461-1247-0.

16. Зандстра, М. PHP. Объекты, шаблоны и методики программирования / М. Зандстра. – Санкт-Петербург: Диалектика, 2019. – 864 с. – ISBN 978-5-907458-23-9.

17. Самоучитель PHP 7 / М. Кузнецов, И. Симдянов. – Санкт-Петербург :

БХВ-Петербург, 2018. – 448 с. – ISBN 978-5-9775-3817-6.

18. Разработка веб-приложений с помощью PHP и MySQL / Л. Веллинг, Л. Томсон. – Санкт-Петербург : Диалектика, 2019. – 672 с. – ISBN 5-93772-090-3.

19. VirusTotal Documentation! : официальный сайт. – Соединенные Штаты Америки. – Обновляется в течение суток. – URL: https://support.virustotal.com/hc/en-us/articles/115002100149-API (дата обращения: 2.12.2022).

20. VirusTotal and Chronicle! : официальный сайт. – Соединенные Штаты Америки. – Обновляется в течение суток. – URL: https://blog.virustotal.com/2018/01/virustotal-and-chronicle.html (дата обращения: 24.11.2022).

21. OWASP Joomla Vulnerability Scanner Project : сайт – Соединенные Штаты Америки. – Обновляется в течение суток. – URL: https://wiki.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Scanner_Project (дата обращения: 4.12.2022).

22. Документация OpenCart! : официальный сайт. – Соединенные Штаты Америки. – Обновляется в течение суток. – URL: https://opencart.club/doc/ (дата обращения: 20.10.2022).

23. CMS.S3! : официальный сайт. – Россия. – Обновляется в течение суток. – URL: https://megagroup.ru/cms (дата обращения: 21.10.2022).

Приложение А

Техническое задание

"СОГЛАСОВАНО"	"УТВЕРЖДЕНО"
к.ф-м.н., доц. каф. «КБИС»	Зав. каф. «КБИС», к.т.н.
_____ О.В. Куликова	_____ Д.А. Короченцев
«_____» _____2022 г.	«____» _____2022 г.

А.1 Введение

А.1.1 Наименование программы

Наименование программы – «Модуль для дополнительной защиты сайтов, создаваемых на CMS Joomla».

А.1.2 Область применения

Областями применения данного программного средства являются сайты, разрабатываемые и функционирующие на системе управления содержимым CMS Joomla.

А.2 Основания для разработки

Разработка ведётся на основании приказа об утверждении тем дипломных работ, утвержденного Донским государственным техническим университетом. Тема разработки: «Модуль для дополнительной защиты сайтов, создаваемых на CMS Joomla».

А.3 Назначение разработки

А.3.1 Функционально назначение

Функциональным назначение программного средства является реализация сканирования файлов сайта, основанного на использовании различных вирусных сигнатур.

А.3.2 Эксплуатационное назначение

Эксплуатационное назначение: программное средство предназначено для эксплуатации физических лиц, компаний, организаций, использующих систему управления содержимым CMS Joomla для администрирования и управления сайтом.

А.4 Требования к программе

А.4.1 Требования к функциональным характеристикам

Программное средство должно хранить вирусные сигнатуры, для проверки, также список исключений для игнорирования выбранных файлов. Программное средство должно содержать функционал, позволяющий проверить все файлы сайта на наличие вредоносного ПО. Также необходим функционал интеграции расширения в систему управления содержимым CMS Joomla и добавления интерфейса в административную панель системы.

А.4.2 Требования к надежности

Надежность функционирования обеспечивается корректным функционированием системы управления содержимым CMS Joomla и своевременным обновлением модуля.

А.4.3 Условия эксплуатации

Условия эксплуатации совпадают с условиями эксплуатации персональных ЭВМ IBM PC и совместимых с ними персональных компьютеров. Требуется наличие одного оператора. Программа рассчитана для опытных пользователей персональных компьютеров. Результатом работы является Html-отчет о выполненном сканировании файлов сайта.

А.4.4 Требования к составу и параметрам технических средств

Для работы программы требуется наличие IBM PC – совместимого персонального компьютера с тактовой частотой процессора не менее 1,5 Гц, объемом свободной оперативной памяти не менее 2 Гб, выходом в интернет, клавиатурой и монитором.

А.4.5 Требования к информационной и программной совместимости

Разработанный программный модуль является кроссплатформенным, корректный запуск и работа которого были протестированы на таких операционных системах, как Windows 10, а также Ubuntu 18.05.01 LTS. Базовый язык программирования – PHP. Среда разработки – Visual Studio Code.

А.4.6 Требования к маркировке и упаковке

Требований к маркировке и упаковке не предъявлялось.

А.4.7 Требования к транспортированию и хранению

Требований к транспортированию и хранению не предъявлялось.

А.5 Требования к программной документации

Программная документация должна включать следующие документы:

– документ «Техническое задание» (ГОСТ 19.201-78);

А.6 Стадии и этапы разработки

1. Системный анализ.
2. Общесистемное проектирование.
3. Подготовка технологических средств.
4. Программная реализация, рабочий проект.
5. Тестовые испытания программного средства.

А.7 Порядок контроля и приемки

Контроль и приемка разработки осуществляется на основе испытаний отладочных примеров. Примеры должны демонстрировать правильность работы используемых в программе структур и алгоритмов в различных ситуациях, которые могут возникнуть при выполнении программы. При этом проверяется выполнение всех функций программы и полнота документации.

Разработчик: Беловолов Михаил Юрьевич

Дата начал разработки: «_ _» _____ 2022 г.

Приложение Б

Руководство системного программиста

Б.1 Общие сведения о программе

Оператор осуществляет взаимодействие с программным средством, являющимся макетом программного обеспечения, предназначенным для выявления вредоносного ПО в файлах сайта.

Программное средство предназначено для работы физических лиц, сотрудников предприятия, организаций, компаний, использующих систему управления содержимым CMS Joomla для функционирования и пользования сайта.

Б.2 Структура программы

Программное средства состоит из исполняемого скрипта и модулей для интеграции скрипта в систему управления содержимым CMS Joomla, которые можно интегрировать из Windows 10, Ubuntu 18.05.01 LTS и других популярных на сегодняшний день операционных систем в любую версию CMS Joomla.

Б.3 Настройка программы

Для настройки программного средства необходимо интегрировать модуль в CMS Joomla через административную панель в разделе установки расширений.

Б.4 Проверка программы

Проверка работоспособности программного средства осуществляется путем ее запуска согласно руководству оператора.

Б.5 Сообщения системному программисту

Для системного программиста в данном программном продукте предназначены сообщения о состоянии проверки файлов сайта.

Приложение В

Руководство программиста

В.1 Общие сведения о программе

Программист осуществляет взаимодействие с программным средством, предназначенным для сканирования файлов сайта на наличие вредоносного ПО.

Программное средство предназначено для работы физических лиц, сотрудников предприятия, организаций, компаний, использующих систему управления содержимым CMS Joomla для функционирования и пользования сайта.

В.2 Характеристика программы

Программное средство позволяет провести сканирование файлов сайтов, создаваемых и функционирующих на системе управления содержимым CMS Joomla, а также результатом программы является Html-отчет о результатах сканирования.

В.3 Обращение к программе

Обращение к программному средству осуществляется после его установки в CMS Joomla, где в случае удачной интеграции, в административной панели появляется кнопка с названием расширения. По нажатию на данную кнопку происходит запуск программного средства.

В.4 Входные и выходные данные

В качестве входных данных программа получает список файлов сайта, а также пути до файлов из списка исключений.

В результате работы программного средства получаем Html-отчет о результатах сканирования.

В.5 Сообщения

Для программиста в разработанном программном продукте предусмотрены информационные сообщения, которые выводятся в интерфейс по мере необходимости.

Приложение Г

Руководство оператора

Г.1 Общие сведения о программе

Оператор осуществляет взаимодействие с программным средством, предназначенным для сканирования файлов сайта на наличие вредоносного ПО.

Программное средство предназначено для работы физических лиц, сотрудников предприятия, организаций, компаний, использующих систему управления содержимым CMS Joomla для функционирования и пользования сайта.

Г.2 Условия выполнения программы

Для успешного выполнения разработанного программного обеспечения необходимо выполнение всех аппаратных и программных требований и корректные входные данные.

Г.3 Выполнение программы

Для работы с приложением нужно интегрировать разработанное расширение в CMS Joomla, в разделе установки. Далее, в случае успешной интеграции, в административной панели появляется кнопка с названием расширения «Site scanner». Для запуска программного средства необходимо нажать на данную кнопку для старта сканирования.

Приложение Д

Листинг программы

1 Листинг Д.1 – XML-функция интеграции в систему

```
<?xml version="1.0" encoding="utf-8"?>
<extension type="component" method="upgrade">
    <name>Simple virus scanner</name>
    <creationDate>October 2022</creationDate>
    <author>John Smith</author>
    <copyright>Mikhail Belovolov</copyright>
    <license>GPL v3</license>
    <version>0.0.1</version>

    <description>
        Site scanner downloaded succesfully. Mikhail Belovolov
graduate work.
    </description>

    <namespace
path="src/">JohnSmith\Component\SimpleVirusScanner</namespace>

    <administration>
        <menu link="index.php?option=com_simplevirusscanner">Site
scanner</menu>
        <files folder="admin/">
            <folder>services</folder>
            <folder>src</folder>
            <folder>tmpl</folder>
        </files>
    </administration>
</extension>
```

2 Листинг Д.2 – Функция-интегратор модулей отображения

```
<?php
defined('_JEXEC') or die;
use Joomla\CMS\Dispatcher\ComponentDispatcherFactoryInterface;
use Joomla\CMS\Extension\ComponentInterface;
use Joomla\CMS\Extension\MVCComponent;
use Joomla\CMS\Extension\Service\Provider\ComponentDispatcherFactory;
use Joomla\CMS\Extension\Service\Provider\MVCFactory;
use Joomla\CMS\MVC\Factory\MVCFactoryInterface;
use Joomla\DI\Container;
use Joomla\DI\ServiceProviderInterface;
```

```
return new class implements ServiceProviderInterface {

    public function register(Container $container): void {
        $container->registerServiceProvider(new
MVCFactory('\\JohnSmith\\Component\\SimpleVirusScanner'));
        $container->registerServiceProvider(new
ComponentDispatcherFactory('\\JohnSmith\\Component\\SimpleVirusScanner'))
;
        $container->set(
            ComponentInterface::class,
            function (Container $container) {
                $component = new MVCComponent($container-
>get(ComponentDispatcherFactoryInterface::class));
                $component->setMVCFactory($container-
>get(MVCFactoryInterface::class));

                return $component;});};};
```

3 Листинг Д.3 – Функция отображения интерфейса в административной панели

```
<?php
namespace
JohnSmith\Component\SimpleVirusScanner\Administrator\Controller;

defined('_JEXEC') or die;

use Joomla\CMS\MVC\Controller\BaseController;

class DisplayController extends BaseController {
    protected $default_view = 'svs';
    public function display($cachable = false, $urlparams =
array()) {
        return parent::display($cachable, $urlparams);}}
```

4 Листинг Д.4 – Функция формирования Html-отчета

```
<?php
namespace
JohnSmith\Component\SimpleVirusScanner\Administrator\View\Svs;

defined('_JEXEC') or die;

use Joomla\CMS\MVC\View\HtmlView as BaseHtmlView;

class HtmlView extends BaseHtmlView {
```

```
    /**
     * Отображение основного вида "Site Scanner"
     *
     * @param   string  $tpl  Имя файла шаблона для анализа;
автоматический поиск путей к шаблону.
     * @return  void
     */
    function display($tpl = null) {
        parent::display($tpl);}}
```

5 Листинг Д.5 – Функция-сканер файлов сайта

```
<?php

defined('_JEXEC') or die('Нет прямого доступа к этому файлу');

define('SVS_FILE_EXTENSIONS', "php,inc");
define('SVS_FORBIDDEN_DIRS', "img,images,upload,tmp,assets");
define('SVS_MAX_FILESIZE', 2100000);
define('SVS_VERSION', "2.44");
?>
<h2>Результат сканирования:</h2>

<?
class RainbowCounter
{
    var $rainbowColors = ["#9b350c", "#009688", "#607d8b",
"#03a9f4", "#9c27b0", "#673ab7", "#e91e63", "#ff9800", "#795548",
"#cddc39", "#681676", "#4983dd", "#49ddd1", "#9b700c", "#e45023",
"#a80b63", "#db246a", "#dd7128", "#e95a2f"];
    var $rainbowCnt = 0;
    var $isWeb = true;

    public function __construct($isWeb)
    {
        $this->isWeb = $isWeb;}
    public function rainbow($text)
    {
        if (!$this->isWeb) return $text;

        $text = "<span style='color:" . $this-
>rainbowColors[$this->rainbowCnt] . ";'>$text</span>";

        return $text;
    }

    public function reset()
    {$this->rainbowCnt = 0;}
```

```
    public function inc()
    {
        if ($this->rainbowCnt < count($this->rainbowColors) - 1) {
            $this->rainbowCnt++;
        } else {
            $this->rainbowCnt = 0;}}}
class Scanner
{
    var $rootFolder = "";
    var $isWeb = true;
    var $scanList = [];
    var $forbiddenDirs = [];
    var $scanExtentions = [];
    private $selfSize = 0;
    private $selfCRC = "";
    private $totalFilesScanned = 0;
    private $totalFilesSuspicious = 0;
    private $totalFilesExcluded = 0;
    var $noSort = false;

    private $minScore = 6;

    public function __construct($rootFolder = __DIR__)
    {
        $this->rootFolder = $_SERVER['DOCUMENT_ROOT'];
        $this->isWeb = (php_sapi_name() !== 'cli');
        $options =  getopt('', ['html::', 'm::', 'root::',
'nosort::']);

        if (isset($options['html'])) $this->isWeb = true;
        if (isset($_GET['plaintext'])) $this->isWeb = false;
        if (isset($options['root'])) $this->rootFolder =
$options['root'];
        if (isset($options['nosort'])) $this->noSort = true;
        if (isset($_GET['nosort'])) $this->noSort = true;
        if (isset($options['m'])) $this->minScore = $options['m'];
        if (isset($_GET['m'])) $this->minScore = $_GET['m'];

        $this->selfSize = filesize(__FILE__);
        $this->selfCRC = crc32(file_get_contents(__FILE__));

        $this->forbiddenDirs = explode(",", SVS_FORBIDDEN_DIRS);

        $exts = explode(",", SVS_FILE_EXTENSIONS);
        $this->scanExtentions = "~\.(" . join("|", $exts) .
")$~i";}
    public function error($errorText)
    {
        if ($this->isWeb) {
            echo "<span style='color:red'>ERROR: </span><br>" .
$errorText . "<br>";
```

```
        } else {
            echo ">> ERROR: " . $errorText . "\n";}}
    private function out($message)
    {
        if ($this->isWeb) {
            echo $message . "<br>";
        } else {
            echo $message . "\n";}}
    public function buildScanList($startDir = "", $level = 0)
    {
        $dirToScan = ($startDir != "") ? $this->rootFolder . "/" .
$startDir : $this->rootFolder;
        if ($handle = opendir($dirToScan)) {
            $exts = $this->scanExtentions;

            $arrDirs  = [];
            $arrFiles = [];

            while (false !== ($entry = readdir($handle))) {
                if ($entry != "." && $entry != "..") {
                    if (is_dir($dirToScan . "/" . $entry) &&
!is_link($dirToScan . "/" . $entry)) {
                        $arrDirs[] = $entry;
                    } else {
                        //extension is in the list and the file is
not self
                        if (preg_match($exts, $entry) > 0 &&
$dirToScan . "/" . $entry !== __FILE__) {
                            $arrFiles[] = $entry;}}}}

            closedir($handle);
            asort($arrDirs);
            asort($arrFiles);

            foreach ($arrFiles as $entry) {
                $dirPath = ($startDir != "") ? $startDir : "/";
                $this->totalFilesScanned++;
                $id = $this->totalFilesScanned;
                $this->scanList[$id]['dir'] = $dirPath;
                $this->scanList[$id]['name'] = $entry;
                $this->scanList[$id]['score'] = 0;
                $this->scanList[$id]['level'] = $level;
                $this->scanList[$id]['diag'] = [];
            }

            unset($arrFiles);

            foreach ($arrDirs as $entry) {
                $this->buildScanList($startDir . "/" . $entry,
$level + 1);
            }
```

```
        } else {
            $this->error("Cannot open root folder '" . $this-
>rootFolder . "'");
        }
    }

    function isExclusion($dirPath, $fileName)
    {
        $fullPath = ($dirPath == "/") ? $this->rootFolder : $this-
>rootFolder . $dirPath;
        $file = $fullPath . "/" . $fileName;
        $arrWhitelist = ['snusminer.php',
        '/administrator/components/com_akeeba/BackupEngine/Archive
r/Jpa.php',
        '/administrator/components/com_akeeba/BackupEngine/Archive
r/Jps.php',
        '/administrator/components/com_akeeba/BackupEngine/Archive
r/Zip.php',
        '/administrator/components/com_akeeba/BackupEngine/Postpro
c/Sugarsync.php',
        '/administrator/components/com_akeeba/restore.php',
        '/administrator/components/com_akeeba/View/Upload/tmpl/def
ault.php',
        '/administrator/components/com_akeeba/View/Upload/tmpl/don
e.php',
        '/administrator/components/com_akeeba/View/Upload/tmpl/err
or.php',
        '/administrator/components/com_akeeba/View/Upload/tmpl/upl
oading.php',
        '/administrator/components/com_finder/helpers/indexer/stem
mer/fr.php',
        '/administrator/components/com_joomlaupdate/views/upload/t
mpl/captive.php',
        '/administrator/components/com_media/views/images/tmpl/def
ault.php',
        '/image_uploader/localization.php',
        '/libraries/idna_convert/idna_convert.class.php',
        '/libraries/vendor/joomla/string/src/phputf8/native/core.p
hp',
        '/modules/fileman/classes/general/sticker.php',
        '/modules/iblock/admin/iblock_edit.php',
        '/modules/iblock/admin/iblock_element_admin.php',
        '/modules/iblock/admin/iblock_list_admin.php',
        '/modules/iblock/admin/templates/iblock_subelement_list.ph
p',
        '/modules/iblock/classes/general/comp_pricetools.php',
        '/modules/main/admin/group_edit.php',
        '/modules/main/admin/restore.php',
        '/modules/main/admin/site_checker.php',
        '/modules/main/classes/general/backup.php',
        '/modules/main/classes/general/cache_files.php',
```

```
        '/modules/main/classes/general/cache_html.php',
        '/modules/main/classes/general/punycode.php',
        '/modules/main/classes/general/usertypedbl.php',
        '/modules/main/classes/general/virtual_io_filesystem.php',
        '/modules/main/classes/general/zip.php',
        '/modules/main/lib/io/directory.php',
        '/modules/main/tools.php',
        'administrator/components/com_joomlaupdate/restore.php',
        'modules/defa.socialmediaposter/classes/general/idna_conve
rt.class.php',
        'libraries/vendor/algo26-matthias/idna-
convert/src/NamePrep/NamePrepData2008.php',
        'administrator/components/com_simplevirusscanner/tmpl/svs/
default.php',
        ];

        $isEx =  false;

        if ($fileName == __FILE__) {
            $isEx = true;
        }

        foreach($arrWhitelist as $wl) {
            if (strstr($file, $wl) !== false) {
                $isEx = true;
                break;}}
        return $isEx;}

    function scanFile($id)
    {
        $rbc = new RainbowCounter($this->isWeb);
        $entry = &$this->scanList[$id];
        $dirPath = $entry['dir'];
        $fileName = $entry['name'];

        foreach ($this->forbiddenDirs as $fd) {
            if (preg_match("~\/" . $fd . "[/$]~i", $dirPath) > 0)
{
                $entry['score'] = $entry['score'] + 50;
                $entry['diag']['susp_dir'] = $rbc-
>rainbow('suspicious dir \''.$fd.'\'');
                break;
            }
        }
        $rbc->inc();

        $contents = "";
        $fullPath = ($dirPath == "/") ? $this->rootFolder : $this-
>rootFolder . $dirPath;
        if (filesize($fullPath . "/" . $fileName) <
SVS_MAX_FILESIZE) {
```

```
            $contents = file_get_contents($fullPath . "/" .
$fileName);
        } else {
            $this->error("File too big: " . $fullPath . "/" .
$fileName. " (". filesize($fullPath . "/" . $fileName). ")");
            $entry['score'] = $entry['score'] + 50;
            $entry['diag']['too_big'] = $rbc->rainbow('file too
big');
            return true;
        }
        $rbc->inc();

        if ($contents === false) {
            $this->error("Cannot read " . $fullPath . "/" .
$fileName);
            $entry['score'] = $entry['score'] + 50;
            $entry['diag']['cant_read'] = $rbc->rainbow('can\'t
read');
            return true;
        }
        $rbc->inc();

        if (preg_match('~(\\\x[a-z0-9]{2,3}){3,}~', $contents) >
0) {
            $entry['score'] = $entry['score'] + 10;
            $entry['diag']['esc_seq'] = $rbc->rainbow('escape
sequence');
        }
        $rbc->inc();

        if (preg_match('~@\$\{\"~', $contents) > 0) {
            $entry['score'] = $entry['score'] + 30;
            $entry['diag']['weird_constr'] = $rbc->rainbow('weird
construction');
        }
        $rbc->inc();

        $matches = [];
        $cnt = preg_match_all('~[a-zA-Z0-9]{35,}~', $contents);
        if ($cnt > 1) {
            $entry['score'] = $entry['score'] + 5 + floor($cnt /
5)*5;
            $entry['diag']['long_id'] = $rbc->rainbow('long
identifier (x' . $cnt . ')');
        }
        $rbc->inc();

        if (preg_match('~base64_decode~', $contents) > 0) {
            $entry['score'] = $entry['score'] + 10;
            $entry['diag']['base64'] = $rbc-
>rainbow('base64_decode()');
```

```
        }
        $rbc->inc();

        if (preg_match('~php_uname~', $contents) > 0) {
            $entry['score'] = $entry['score'] + 20;
            $entry['diag']['php_uname'] = $rbc-
>rainbow('php_uname()');
        }
        $rbc->inc();

        $matches = [];
        $cnt = preg_match_all('~chr\(\d{2,3}\)~', $contents,
$matches);
        if ($cnt > 0) {
            $entry['score'] = $entry['score'] + 5 + floor($cnt /
10)*5;
            $entry['diag']['chr'] = $rbc->rainbow('chr() (x' .
$cnt . ')');
        }
        $rbc->inc();

        if (preg_match('~shell_exec\s*\(~', $contents) > 0) {
            $entry['score'] = $entry['score'] + 50;
            $entry['diag']['shell_exec'] = $rbc-
>rainbow('shell_exec()');
        }
        $rbc->inc();

        $matches = [];
        $cnt = preg_match_all('~chmod\s*\(~', $contents,
$matches);
        if ($cnt > 0) {
            $entry['score'] = $entry['score'] + 10;
            $entry['diag']['chmod'] = $rbc->rainbow('chmod()');
        }
        $rbc->inc();

        if (preg_match('~\$[a-zA-Z0-9]{5,}\{\d{2}}~', $contents) >
0) {
            $entry['score'] = $entry['score'] + 30;
            $entry['diag']['weird_var'] = $rbc->rainbow('weird
variables');
        }
        $rbc->inc();

        $matches = [];
        $cnt = preg_match_all('~0x[0-9A-F]{4}~', $contents,
$matches);
        if ($cnt > 3) {
            $entry['score'] = $entry['score'] + 5 + floor($cnt /
20);
```

```
            $entry['diag']['hex'] = $rbc->rainbow('hex codes (x' .
$cnt . ")");
        }
        $rbc->inc();

        $matches = [];
        $cnt = preg_match_all('~eval\(~', $contents, $matches);
        if ($cnt > 0) {
            $entry['score'] = $entry['score'] + 10 + floor($cnt /
5)*5;
            $entry['diag']['eval'] = $rbc->rainbow('eval()');
        }
        $rbc->inc();

        $matches = [];
        $cnt = preg_match_all('~fileperms\(~', $contents,
$matches);
        if ($cnt > 0) {
            $entry['score'] = $entry['score'] + 10 + floor($cnt /
5)*5;
            $entry['diag']['fileperms'] = $rbc-
>rainbow('fileperms()');
        }
        $rbc->inc();

        if (preg_match('~gzinflate\(~', $contents) > 0) {
            $entry['score'] = $entry['score'] + 10;
            $entry['diag']['gzinflate'] = $rbc-
>rainbow('gzinflate()');
        }
        $rbc->inc();

        if (preg_match('~register_shutdown_function\(~',
$contents) > 0) {
            $entry['score'] = $entry['score'] + 5;
            $entry['diag']['register_shutdown_function'] = $rbc-
>rainbow('register_shutdown_function()');
        }
        $rbc->inc();

        if (preg_match('~[^"a-zA-Z0-9_$]goto\s~', $contents) > 0)
{
            $entry['score'] = $entry['score'] + 20;
            $entry['diag']['goto'] = $rbc->rainbow('goto');
        }
        $rbc->inc();

        if (preg_match('~enctype\s*=\s*[\'"]multipart/form-
data[\'"]~i', $contents) > 0) {
            $entry['score'] = $entry['score'] + 5;
```

```
            $entry['diag']['upload'] = $rbc->rainbow('upload
form');
        }
        $rbc->inc();

        if (preg_match('~include\s*\([^;]*\$_GET~Uis', $contents)
> 0) {
            $entry['score'] = $entry['score'] + 50;
            $entry['diag']['include_get'] = $rbc->rainbow('include
from $_GET');
        }
        $rbc->inc();

        foreach ($this->forbiddenDirs as $fd) {
            if (preg_match("~include\s*\([^;]*\/".$fd."[^a-zA-Z0-
9]~is", $contents) > 0) {
                $entry['score'] = $entry['score'] + 50;
                $entry['diag']['susp_include'] = $rbc-
>rainbow('include in \''.$fd.'\'');
                break;}}
        $rbc->inc();

        if (preg_match('~\$USER->Authorize\([^\$]~', $contents) >
0) {
            $entry['score'] = $entry['score'] + 50;
            $entry['diag']['authorize'] = $rbc-
>rainbow('authorize()');}
        $rbc->inc();
        if (count($entry['diag']) > 2) {
            $entry['score'] = $entry['score'] + 20;
        }

        $entry['exclusion'] =  $this->isExclusion($dirPath,
$fileName);
    }

    public function scanFiles()
    {
        foreach ($this->scanList as $id => $entry) {
                $this->scanFile($id);}}

    private function pre($var, $export =  false)
    {
        if ($this->isWeb) echo ("<pre>");

        if ($export) {
            var_export($var);
        } else {
            $this->out($var);
        }
        if ($this->isWeb) echo ("</pre>");
```

```
    }
    public function sortScanList(){
        global $noSort;
        $noSort = $this->noSort;

        function comp1($a, $b){
            global $noSort;
            if ($a["exclusion"] || $b["exclusion"] ) {
                if ($a["exclusion"] && $b["exclusion"]) {
                    if ($noSort) return 0;
                    return ($a["score"] > $b["score"]) ? -1 : 1;
                } else {
                    return 1;
                }
            } else {
                if ($noSort) return 0;
                if ($a["score"] == $b["score"]) return 0;
                return ($a["score"] > $b["score"]) ? -1 : 1;}}
        usort($this->scanList, "comp1");}
    public function printScanList()
    {
        $marker = $this->isWeb ? "■" : "+";

        $this->out('<div class="black">Минимальная оценка
безопасности: '.$this->minScore.'</div>');

        if ($this->isWeb) {
            ob_start();
            echo ("<html>
            <head>
            <title>Сканнирование сайта</title>
            <style>
<!-- стили -->
            </style>
            </head>
            <body>
            <table border='0' cellpadding='2' cellspacing='0'>
            <tr>
                <th class='file'>Файл</th>
                <th>Уровень опасности</th>
                <th class='alerts'>Обнаружены</th>
            </tr>
            ");
        } else {
            $this->out("-- Запуск сканнера --");
        }

        foreach ($this->scanList as $entry) {
            $dir = $entry['dir'];
            $dirname = ($dir == "/" || $dir == "") ? "/" :
substr($dir, 1) . "/";
```

```
            $dirname = $this->isWeb ? "<span class='directory'>" .
$dirname . "</span>" : "" . $dirname . "";

            if ($entry['score'] > $this->minScore) {
                $markerCount = floor($entry['score'] / 10);
                if ($markerCount > 10) $markerCount = 10;

                if ($this->isWeb) {
                    echo ("
                        <tr>
                            <td ".(($entry['exclusion']) ? "
class='exclusion'" :   "").">$dirname<b>" . $entry['name'] . "</b>
</td>
                            <td>" . (($entry['exclusion']) ?
                                        " <span
class='exclusion'>X</span>" :
                                        "<span
class='marker'>".  str_repeat("$marker",  $markerCount) .
"</span>") .
                            " [".$entry['score']."]
                            </td>
                            <td>" . join(", ", $entry['diag']) .
"</td>
                        </tr>
                        ");
                        ob_flush();
                } else {
                    $name =  $entry['name'];
                    $this->out((($entry['exclusion']) ? "X " :
str_repeat("$marker",  $markerCount))  . " [".$entry['score']."] "
. $dirname . $name .
                            " [" . join(", ", $entry['diag']) . "]");
                }

                if ($entry['exclusion']) {
                    $this->totalFilesExcluded++;
                } else {
                    $this->totalFilesSuspicious++;}}}
        if ($this->isWeb) {
            echo ("</table>");
            ob_end_flush();
        }
        $this->out("<div class='lastmes'>Всего просканировано
файлов " . $this->totalFilesScanned);
        $this->out("Из них потенциально опасных: " . $this-
>totalFilesSuspicious." + ".$this->totalFilesExcluded."
исключения(-ий).");
        $this->out("Отчет предоставлен 'Site Scanner' " . date("Y-
m-d H:i")."</div>");
    }
}
```

```
$scanner = new Scanner();
$scanner->buildScanList();
$scanner->scanFiles();
$scanner->sortScanList();
$scanner->printScanList();
```