МОСКОВСКИЙ ФИНАНСОВО-ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

МФЮА

КАФЕДРА ЗАЩИТЫ ИНФОРМАЦИИ

СПЕЦИАЛЬНОСТЬ ОРГАНИЗАЦИЯ И ТЕХНОЛОГИЯ ЗАЩИТЫ ИНФОРМАЦИИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

На тему: Разработка метода защиты сайтов от сканирования и хаотичных интенсивных запросов

Студента

Солодова Максима Вадимовича

Научный руководитель

Черепанов А.В

Задание на выполнение выпускной квалификационной работы

Студенту Солодову Максиму Вадимовичу

Тема: Разработка метода защиты сайтов от сканирования и хаотичных интенсивных запросов

Исходные данные к работе: Нормативно-правовая, учебная, периодическая литература, материалы преддипломной практики

Содержание пояснительной записки (перечень подлежащих разработке вопросов)

1. Анализ распространенных видов сканирования и хаотичных интенсивных запросов

1.1. Анализ распространенных видов сканирования

1.1.1. Средства копирования содержимого сайта и осуществления поиска в нем

1.1.2. Программы для поиска уязвимостей

1.1.3. Результаты анализа распространенных видов сканирования сайта

1.2. Анализ хаотичных интенсивных запросов

1.2.1. Средства подбора паролей и несанкционированный доступ

1.2.2. Средства, направленные на нарушение работы сайта

1.2.3. Результаты анализа хаотичных интенсивных запросов

2. Разработка методов защиты от сканирования и хаотичных интенсивных запросов

2.1. Методы защиты от сканирования

2.2. Методы защиты от хаотичных интенсивных запросов

2.3. Программные средства защиты сайта

2.4. Разработка методов защиты сайта от сканирования и хаотичных интенсивных запросов

3. Реализация системы защиты в виде PHP-скрипта

3.1. Данные о реализуемом php-скрипте

3.2. Листинг программы

4. Тестирование PHP-скрипта, внедренного в сайт

4.1. Тестирование вручную

4.2. Тестирование при помощи онлайн-сервисов

4.3. Результаты тестирования php-скрипта, внедренного в сайт

5. Экономическая часть

5.1. Введение

5.1. Разработка сетевой модели решения поставленной задачи, составление календарного плана-графика выполнения работ

5.2. Расчет затрат на создание скрипта для защиты сайта от сканирования и хаотичных интенсивных запросов

6. Охрана труда и окружающей среды

6.1. Воздушная среда

6.2. Шум, инфразвук, ультразвук и вибрация.

6.3. Электробезопасность

6.4. Воздействие электромагнитных полей

6.5. Производственное освещение

Срок предоставления студентом законченной работы 13.12.13

Научный руководитель _____ Черепанов А.В.

Задание принял к исполнению_____ Солодов Максим Вадимович

Введение

В Интернете существует проблема медленной работы сайтов, связанная с частыми обращениями к ним. Чаще всего это крупные порталы с высокой посещаемостью. Но эта проблема может коснуться и небольших сайтов, так как даже при малой посещаемости сайт может подвергаться высокой нагрузке. Высокая нагрузка создается различными роботами, постоянно сканирующими сайты. При этом работа сайта может сильно замедлиться, или он вообще может оказаться недоступным.

Так же свою лепту в снижение производительности сайта или вообще отказа в обслуживании вносят пользователи. Если у сайта достаточно высокая посещаемость, то большое значение играет количество страниц, запрошенных одним пользователем за малый промежуток времени.

Для того чтобы выбрать и разработать систему защиты, которая будет наиболее эффективно снижать нагрузку, нужно проанализировать виды сканирования и хаотичных интенсивных запросов.

Существует много различных программ и ресурсов в мировой сети осуществляющих сканирование сайтов с целью получения информации, но из всего их множества мы проанализируем самые распространенные. Так же по аналогии с анализом видов сканирования проанализируем самые распространенные виды хаотичных интенсивных запросов.

Сканирование сайта производится программами, сторонними сайтами или вручную. При этом создается большое количество запросов в короткий промежуток времени. Сканирование сайта чаще всего используют для поиска в нем уязвимостей или копирования содержимого сайта.

Хаотичные интенсивные запросы – это случайные или злонамеренные многочисленные запросы в короткий промежуток времени на страницы сайта со стороны пользователей или роботов. К примеру случайных интенсивных запросов относится частое обновление страницы. К злонамеренным многочисленным запросам относится спам на страницы сайта со стороны

пользователей или DoS атаки.

В работе будут разработаны рекомендации по защите сайта от сканирования и хаотичных интенсивных запросов, и создана система на основе php-сценария, использующая эти рекомендации.

1 . Анализ распространенных видов сканирования и хаотичных интенсивных запросов

1.1 Анализ распространенных видов сканирования

1.1.1 Средства копирования содержимого сайта и осуществления поиска в нем

Сканирование сайта осуществляется программами или сайтами. Целью такого сканирования может являться копирование содержимого сайта, изучение структуры сканируемого веб-ресурса, поиск определенного рода информации, а также анализ полученного содержимого.

Можно выделить два основополагающих вида сканирования: это сканирование сайта вручную и сканирование сайта при помощи программы-краулера. Остановимся на каждом из этих видов подробнее.

Сканирование узла вручную возможно лишь в случае, если ресурс содержит не слишком много страниц. В подобной ситуации, для получения полного перечня имеющихся на ресурсе гиперссылок, вполне можно использовать обычный браузер. В сравнении с использованием программы-краулера подобный подход значительно точнее. Дело в том, что программы-краулеры далеко не всегда могут правильно интерпретировать различные клиентские сценарии и, соответственно, содержащиеся в них гиперссылки. Но с другой стороны сканирование веб-ресурса вручную – это очень долгий и трудоемкий процесс [1, стр. 181].

Во время сканирования сайта вручную человек обращает внимание на наличие статических и динамических страниц на ресурсе, изучает структуру каталогов, получает доступ к вспомогательным файлам. Особое внимание уделяется используемым на страницах сайта Java-классам, а также апплетам. Кроме того, определенную пользу представляют собой строки запросов [2, стр. 116].

Наибольшая польза для атакующего будет получена именно при

«ручном» исследовании web-страниц целевого ресурса. В такой ситуации у злоумышленника получится извлечь наиболее полезную информацию – оценить уязвимость кода, наличие комментариев, а также разобраться с методами построения сайта. Нередко в этом деле необходимо задействовать интуицию.

В коде документа HTML содержится множество интересных сведений, которые порой недоступны при простом взгляде в окно браузера. К подобным сведениям можно отнести наличие комментариев разработчика в коде страницы, наличие адресов электронной почты, номеров телефонов и прочее [3, стр. 372].

Краулер (crawler) представляет собой программу, которая позволяет автоматизировать процесс сканирования сайта. Подобная программа может рассматриваться в качестве браузера, способного самостоятельно переходить по каждой ссылке, расположенной на текущей странице и, таким образом, перемещаться по всем доступным страницам веб-ресурса. Программа-краулер отправляет веб-серверу HTTP-запрос GET, после чего выполняет синтаксический анализ HTML-кода, который получает в качестве ответа. После получения ответа от сервера программа извлекает из него все доступные гиперссылки и рекурсивно продолжает выполнять те же действия касательно каждой из найденных гиперссылок.

В целом, программы-краулеры, вполне могут быть довольно сложными. Они могут не только переходить по гиперссылкам, но и создавать образ сканируемого веб-ресурса на локальном жестком диске. Из подобного образа программа вполне сможет извлечь разного рода элементы, к примеру, клиентские сценарии, комментарии и пр. [1, стр. 181].

Сегодня существует довольно много программных инструментов, которые помогают собирать информацию о веб-приложениях. В большинстве своем такие программные средства полностью копируют рассматриваемое веб-приложение на локальный диск, после чего анализируют полученные данные. Отдельно следует отметить, что подобным

инструментам не под силу создавать копии таких узлов, которые содержат сценарии ASP или же запросы к базам данных.

Далее рассмотрим инструменты, предназначенные для автоматизации сканирования сайта.

Утилита lynx до сих пор является прекрасным инструментом, отлично подходящим для работы с отдельными URL адресами. В основном данным инструментом пользуются в сценариях getit для того, чтоб обеспечивать HTTP-аутентификацию с применением параметра auth.

Утилита Wget представляет собой инструмент, который работает в командной строке под платформами Windows или UNIX. Этот инструмент дает возможность получить на локальный жесткий диск полное содержимое сканируемого Web-узла. Данная программа отлично справляется с получением файлов со сканируемого сайта. Правда, для анализа полученных результатов эта утилита не слишком пригодна.

Утилита WebSleuth представляет собой инструмент, дающий возможность автоматически исследовать веб-узлы. Кроме того, программа сочетает в себе возможности присущие персональному прокси-серверу, такому как Achilles. Утилита предоставляет пользователю возможность находить заданные данные на страницах веб-документа. После выполнения всех действий программа создает отчет, в котором содержатся ссылки на сценарии, html-комментарии, подробная информация об имеющихся файлах cookie и пр.

Утилита Black Widow дает возможность находить и копировать заданную информацию. Еще одним преимуществом данной утилиты является ее возможность сохранения файлов в каталог на локальный жесткий диск. С сохраненным каталогом можно работать при помощи таких инструментов, как findstr или grep.

Утилита Teleport Pro работает на платформе Windows. Программа по своим возможностям приблизительно равна уже рассмотренной утилите Wget. Также она предоставляет возможность анализа полученных данных.

В случае использования программы Teleport Pro можно задавать абсолютно любой начальный URL-адрес. Кроме того, программа дает возможность заранее задать типы искомых файлов. Именно такие файлы и будут загружаться и, соответственно, сохраняться в качестве копии на локальном жестком диске. Основным недостатком этой утилиты можно считать то, что копию сканируемого веб-узла она сохраняет в виде файла с расширением .tpp (это ее собственный недокументированный формат).

Правда, эта утилита не дает возможности использовать поиск в полученном исходном коде страниц веб-ресурса. Для того, чтоб выполнить подобную задачу можно задействовать команду findstr [1, стр. 176].

Утилита Funnel Web Profiler дает возможность выполнения всестороннего анализа сканируемого веб-узла. Программа отличается простым и понятным пользовательским интерфейсом, а также дает возможность получения разного рода информации. После того, как сканирование будет завершено, программный продукт Funnel Web Profiler даст возможность изучить собранную о сайте информацию в разных ракурсах.

Утилита Sam Spade работает на платформе Windows и дает возможность не только проводить сканирование веб-ресурса, но и отсеивать часть «непригодных» результатов [1, стр. 183-185].

Для анализа наиболее распространенных программных средств сканирования сайтов, приведенных выше, сведем информацию об их преимуществах и недостатках в таблицу 1.1.

Таблица 1.1 – Анализ программных средств сканирования сайтов

Название утилиты	Преимущества	Недостатки	Вывод
1	2	3	4
lynx	Работа в командой строке, гибкость, распространяется бесплатно, поддержка SSL	Копировать веб-ресурсы неудобно	Прекрасная программа для проверки из командной строки отдельных адресов URL
Wget	Работа в командой строке, гибкость, распространяется бесплатно, поддержка SSL	Слабые возможности поиска различных элементов (например, комментариев, email и т.д.)	Неплохой инструмент для создания копий веб-узлов работающий из командной строки
WebSleuth	Есть возможность создания отчетов о файлах cookie, формах, передаваемых параметрах и пр.	Необходимо «вручную» заходить на каждую страничку сканируемого сайта	Инструмент более ориентирован на взлом средств проверки данных, которые вводятся в веб-формы
Black Widow	Поддержка режима работы из командной строки, хорошие функции поиска	Это платный программный продукт, нет поддержки SSL	Утилита ориентирована на создание образа веб-узла оснащенного графическим интерфейсом пользователя
Teleport Pro	Утилита проста в использовании	Отсутствует поддержка SSL, программа расспрос-	Неплохая утилита для сканирования сайта состоящего из статичес-
		траняется на коммерческой основе, слабые возможности поиска различных элементов (например, комментариев, email и т.д.)	ких файлов
Funnel Web Profiler	Утилита проста в использовании и предоставляет огромное количество возможностей	Программа распространяется на коммерческой основе	Прекрасная утилита для проведения всестороннего анализа сканируемого веб-ресурса

	связанных с анализом полученных данных		
Sam Spade	Утилита проста в использовании	Это платный программный продукт, нет поддержки SSL	Неплохая утилита для сканирования сайта и отсеивания «излишней» информации

Кроме того, сканирование сайтов можно осуществлять при помощи пользовательских сценариев или же специально написанных скриптов. Конечно же, и скрипты, и сценарии, могут быть созданы на различных языках программирования. Разработанный скрипт или сценарий также сможет справиться с задачей сканирования заданного веб-ресурса.

К тому же, всегда можно найти готовые бесплатные или же недорогие скрипты (сценарии) [3, стр. 372].

Для сканирования веб-ресурса также может использоваться поисковый сервер Google. Этот сервер представляет собой поисковую машину, в базе данных которой содержится очень много информации. Именно поэтому, если робот Google посетил и проиндексировал сайт, значит, информация с этого сайта может быть получена от поисковой системы [2, стр. 118].

Одним из самых простых способов сканирования сайта является поиск страниц ресурса в поисковой системе. В частности, поисковая система Google вполне может проиндексировать страницы ресурсов, где защита основана на том, что при входе пользователь вводит пароль, а не на использовании SSL. В подобной ситуации поисковая система проиндексирует запрещенную веб-страницу и, таким образом, ее можно будет увидеть неавторизованному пользователю, у которого нет на это соответствующих прав. Для этого достаточно верно составить поисковый запрос [4, стр. 25].

Страницы становятся доступны благодаря ошибкам администраторам сайта. В частности страница может быть закрыта паролем, но информация будет находится в другом файле, доступном для индексации или когда-то

открытые страницы сайта, которые успел проиндексировать робот. Так же существует информация о том, что происходит индексация поисковой системой запароленных страниц с помощью браузера Google Chrome.

В целом, после того, как необходимые веб-страницы оказались скопированными на локальный жесткий диск, атакующему все равно придется изучить каждую из скопированных страниц, каждую форму или сценарий, каждый из графических файлов. Это, бесспорно, поможет разобраться в составе и структуре просканированного веб-сайта [3, стр. 375].

Не сложно заметить, что основной целью сканирования сайта является сбор информации о недочетах, оставленных программистами, желающими облегчить и максимально ускорить процесс просмотра и последующей отладки веб-ресурса. Вот так и получается, что у анонимного пользователя может появиться доступ к важным функциям и важной информации на сайте вследствие ошибок допущенных в конфигурации ресурса, а также вследствие непродуманности всей архитектуры узла [2, стр. 136-137].

1.1.2 Программы для поиска уязвимостей

Отдельно следует отметить, что существуют утилиты способные автоматически выявлять общеизвестные уязвимые места на сканируемых веб-ресурсах.

В случае использования готового сценария можно быстро выявить уязвимости, оставленные программистами в защите веб-ресурса. Отметим, что средства выявления такого рода уязвимостей постоянно публикуются на страницах специализированных Интернет-ресурсов.

Далее рассмотрим сценарии и утилиты, позволяющие выявить уязвимость в процессе сканирования сайта.

Сценарий Phfscan.c направлен на выявление уязвимости PHF, которая в свое время оказалась первой общеизвестной уязвимостью, оставляемой в защите скрипта веб-сервера. Такого рода уязвимость дает возможность атакующему исполнять любые команды от лица пользователя веб-сервера.

Сценарий Cgiscan.c направлен на сканирование систем и поиск в них старых уязвимостей (в том числе уязвимости PHF, оставленных test-cgi, count.cgi, webdist.cgi, handler, nph-test-cgi и пр.) [3, стр. 375-376].

Пакет whisker считается одним из лучших средств, предназначенных для сканирования веб-серверов. Данный пакет состоит из двух различных частей. Первая часть представляет собой саму программу сканирования, а во второй располагаются конфигурационные файлы, определяющие, что же будет сканироваться. Данные файлы носят название базы данных сценария. В составе пакета whisker имеется набор баз данных, которые обеспечивают высокую надежность работы утилиты.

Основным преимуществом утилиты whisker является наличие простого языка, на которой написаны сценарии, а также взаимодействия сценариев и базы данных. Данный язык описывается в текстовом документе whisker.txt, который входит в комплект поставки. При помощи этого языка могут быть разработаны собственные базы данных [5, стр. 607-608].

Приложений, предназначенных для проверки определенных веб-сайтов на наличие в них широко известных или же формируемых в качестве умолчания, уязвимых мест, также немало. Их основным отличием является то, что они работают последовательно в, так сказать, ручном режиме.

Утилита Grinder работает на платформе Windows. Программе под силу сканировать одновременно заданный диапазон IP-адресов и, в процессе сканирования, выявлять версии и имена сайтов. Программа способна одновременно создавать несколько параллельно работающих сокетов, что ускоряет ее работу.

Утилита SiteScan показывает лучшие результаты в поиске уязвимых точек (по сравнению с предыдущим приложением) на веб-серверах. Программа одновременно может сканировать лишь один веб-сайт. Утилита работает на платформе Windows.

Среди нестандартных сценариев можно выделить таковые, которые производят атаку подтверждения ввода (или input validation attack). Такие

сценарии действуют против Active Server Pages (ASP), а также Common Gateway Interface (CGI) и Cold Fusion Markup Language (CFML). Они дают возможность обнаружить ошибки допущенные веб-разработчиками. Основа проблемы заключается в неадекватности данных, которые вводятся сценарием в процессе сканирования сайта. В случае если отсутствует проверка ввода, у атакующего есть возможность введения определенного символа в качестве одного из параметров локальной команды. Таким образом, можно заставить сервер выполнить данную команду.

В сети существует множество сценариев и скриптов, использующих данную уязвимость [3, стр. 376-378].

1.1.3 Результаты анализа распространенных видов сканирования сайта

Подытоживая все вышесказанное можно сделать вывод о том, что информацию на сайте необходимо защитить от различных сканирующих программ, а также от тех программ, которые используют уже обнаруженные уязвимости, найденные в используемом программном обеспечении. Сканирование сайта при помощи программы-краулера вполне может быть выявлено. С учетом того, что сканирование сайта может быть направлено не на кражу информации, а представлять собой первый этап исследования ресурса, перед его взломом, следует относиться к этому очень серьезно и создать максимальное количество препятствий на пути автоматизированных средств, предназначенных для сканирования веб-ресурсов. Также необходимо продумать разного рода преграды на пути тех пользователей, которые действуют «вручную» и ставят перед собой цель досконально изучить структуру исследуемого сайта, а также же исследовать веб-ресурс на наличие разного рода уязвимостей.

Для того чтоб избежать взлома сайта и кражи информации с него, необходимо разработать методы защиты от сканирования сайта.

1.2 Анализ хаотичных интенсивных запросов

1.2.1 Средства подбора паролей и несанкционированный доступ

Хаотичные интенсивные запросы представляют собой случайные или же злонамеренные многочисленные запросы, создающиеся со стороны пользователей в короткий промежуток времени на различные страницы сайта. Например, случайным интенсивным запросом может считаться частое обновление страницы. А вот злонамеренным интенсивным запросом вполне можно считать флуд со стороны пользователей на страницы сайта.

В абсолютно любой компьютерной системе постоянно происходят события, которые ведут к изменению ее состояния, или же состояния ее компонентов, к примеру, информационных ресурсов. В данных событиях можно выделить две части – действие, к примеру, чтение, запись, изменение данных, а также адресат этого действия, к примеру, файл или процесс операционной системы. Система защиты ресурса ограничивает различные события, которые могут в ней произойти, не допуская, например, выполнение операции чтения закрытых данных для пользователей, не имеющих на это необходимых прав. Подобные ограничения носят название политики безопасности и, в случае, если в системе происходит множество событий, которые нарушают установленную на сервере политику безопасности, то подобная ситуация вполне может быть интерпретирована как один из признаков хакерской атаки.

Среди таких признаков можно выделить, к примеру, многократные неудачные попытки авторизации или же попытки обращения к файлу, закрытому для неавторизованных пользователей. Еще одним методом выявления различных признаков подозрительной активности является обнаружение подозрительных событий, происходящих в определенный временной промежуток. К примеру, поступление большого количества сетевых пакетов заданного типа за довольно короткий временной интервал вполне может говорить о том, что кто-то пытается просканировать веб-

сервер. Кроме того, подозрительные события, связанные с высокой интенсивностью разного рода действий пользователей, вполне могут быть выявлены при помощи изучения определенных шаблонов хакерских атак. Иначе говоря, речь идет об определенной, уже выявленной последовательности действий злоумышленников связанных со взломом веб-сервера. В частности, выявление множества сетевых пакетов, имеющих специально искаженную структуру (которая зафиксирована в определенном шаблоне) вполне может говорить о том, что предпринимается попытка определить тип операционной системы, установленной на веб-сервере.

Среди подозрительных признаков атаки также числятся непонятные события, связанные с перегрузкой сети, или же внезапное появление множества пакетов, имеющих небольшие размеры. Подобные пакеты называют фрагментированными.

Появление непонятных отказов во время запуска каких-либо служб на веб-сервере, или же возникновение внезапной перегрузки серверного процессора вполне могут свидетельствовать о злонамеренных интенсивных запросах от пользователей. В то же время, причиной возникновения перегрузки процессора на веб-сервере вполне может быть работа программы подбирающей пароли с использованием метода перебора. Еще одним признаком злонамеренных действий может быть внезапная попытка запроса к нетрадиционной службе сервера, например, к службе Telnet [6, стр. 26-28].

Злонамеренными интенсивными запросами можно считать также попытки злоумышленника подобрать пароли пользователей. Взлом системы при помощи подбора пароля никак нельзя назвать интересным или очень захватывающим. В то же время, данный метод является наиболее эффективным среди всех методов взлома разного рода средств веб-аутентификации. Дело в том, что если администратор веб-сервера правильно подобрал протокол аутентификации и в реализации данного протокола не существует каких-либо ошибок, то самым уязвимым аспектом такой системы аутентификации станут выбираемые разными пользователями пароли.

Подбор паролей может производиться как вручную, так и при помощи программных средств, призванных автоматизировать данный процесс. Подобрать пароль вручную довольно непросто, в то же время порой интуиция способна одержать победу над автоматизацией, особо в той ситуации, когда в ответ на неудачную попытку авторизации веб-узел начинает выдавать нестандартные страницы, содержащие сообщения об ошибках. Проверку обычно начинают с самых простых вариантов.

В случае использования автоматизированных средств на сайт обрушивается буквально шквал пар «логин-пароль». И это делается в таком темпе, который человеку никогда не повторить.

Подбор паролей может использоваться против почти всех схем веб-аутентификации.

Наиболее популярными программами для автоматизации процесса подбора паролей являются WebCracker и Brutus.

Утилита WebCracker является простым и удобным средством. Основой ее работы является считывание из файлов пользовательских имен и паролей, после чего утилита использует полученные данные для составления пары «логин-пароль». Данная программа дает возможность прохождения базовой аутентификации. Индикатором успешного прохождения процесса аутентификации для утилиты является сообщение HTTP – 302 Object Moved. Правда, в случае обнаружение подходящей пары «логин-пароль» утилита не прекратит свою работу, а продолжит перебирать все ранее указанные пользователем значения в списках имеющихся логинов и паролей.

Утилита Brutus является универсальным средством, дающим возможность подобрать пароли. Программа обеспечивает возможность взлома механизмов как базовой, так и формоориентированной аутентификации не только по протоколу HTTP, но и по другим протоколам – POP3 и SMTP. Утилита предоставляет пользователю возможность выполнить как взлом по словарю или dictionary attack (в этом случае используется заранее созданный список наиболее часто используемых логинов и паролей

пользователей), так и взлом перебором всех вариантов или brute-force attack (в этом случае пароли генерируются при помощи комбинирования символов из заранее заданного алфавита).

Также данная утилита снабжена подсистемой взлома так называемого формоориентированного механизма аутентификации. Программа в состоянии автоматически проанализировать введенный пользователем URL-адрес и выявить поля для ввода логина и пароля пользователя.

Кроме того, данная программа дает возможность пользователю указать какой код отклика необходимо ожидать во время прохождения аутентификации. Данный момент можно считать очень важным по той причине, что во время прохождения формоориентированной аутентификации бывают ситуации, когда на веб-узле используются нестандартные страницы для извещения об успешном или же неудачном окончании процесса. Утилита дает возможность настроить механизм подбора пары «логин-пароль» так, чтоб программа смогла обработать абсолютно любые (даже нестандартные) коды отклика, которые будут возвращаться веб-узлом.

Среди недостатков программы можно выделить то, что она отображает паролей, подобранных во время формоориентированной аутентификации. Также программа время от времени сообщает о том, что пара «логин-пароль» подобрана, в то время как подобрать ее утилите не удалось. Иначе говоря, происходит «ложное срабатывание» приложения. Но даже несмотря на такие досадные ошибки в работе программы, достоинств у этой утилиты существенно больше, нежели недостатков [2, стр. 161-164].

1.2.2 Средства, направленные на нарушение работы сайта

Нередко бывает так, что злоумышленники просто бессмысленно создают помехи в работе веб-сервера (ради удовольствия) во время его использования. Подобное поведение имеет что-то схожее с актами вандализма. Кроме того, такого рода действия могут быть и своеобразной проверкой наличия уязвимостей в системе, иначе говоря, прелюдией к

дальнейшим атакам или же средством для сокрытия следов оставшихся после несанкционированного доступа. Также интенсивные запросы, создающие нагрузку на веб-сервер, могут посылаться просто в отместку. Заблокировать или же сильно усложнить работу выбранного веб-сервера сравнительно несложно [7, стр. 51].

Одним из видов злонамеренных интенсивных запросов, направленных на нарушение работы сайта, можно считать флуд. Например, если на форуме или же в гостевой книге сайта, любой пользователь может оставить сообщение, то следует учесть то, что часть пользователей решит воспользоваться возможностью отправки сообщений и «пофлудить». В частности, злоумышленник вполне может попытаться просто таки засыпать базу данных сайта многочисленными однотипными и бессмысленными сообщениями, отправляемыми друг за другом и состоящими из повторяющихся фраз, символов, смайлов или различных графических файлов. Флудом может считаться не только отправка бессмысленных сообщений, но и, к примеру, накрутка голосования. Или же действия пользователя, направленные на порождение многочисленных бессмысленных потоков информации. Отметим, что чаще всего это делается с недобрым умыслом.

Такого рода действия пользователя не только создают бессмысленную нагрузку на ресурсы веб-сайта, но и наполняют базу данных ненужной информацией, что в свою очередь ведет к снижению скорости ее работы и, соответственно, увеличению времени, необходимого на выполнение какого-либо запроса к серверу базы данных [8, стр. 180].

Один из видов злонамеренных интенсивных запросов от одного пользователя носит название Denial of Service (DoS), что можно перевести как «отказ в облуживании». Такие атаки не совсем подходят под описание критериев хаотичных интенсивных запросов. Мы рассмотрим их в качестве примера.

Основная цель подобного рода действий, как можно догадаться из

названия, заключается в том, чтоб заставить веб-сервер не отвечать на те запросы, которые поступают от пользователей. Подобного результата злоумышленники добиваются при помощи зацикливания какой-либо работы. К примеру, если на сервере не проверяется корректность всех входящих пакетов, то злоумышленник вполне может создать такой запрос, на который серверу потребуется очень много времени на обработку, соответственно, на работу с другими соединениями у веб-сервера просто не хватит его процессорного времени. В такой ситуации остальные пользователи веб-сайта (клиенты) вынуждены будут получить отказ в обслуживании.

DoS-атаку можно производить несколькими способами, к примеру, используя ошибку в скрипте сайта и перегрузку пользовательского канала, или же мощности веб-сервера, который атакуется. В первом случае злоумышленнику необходимо знать об уязвимостях веб-сайта.

Далее рассмотрим, как же происходит отказ в обслуживании в случае переполнения буфера (эту ошибку злоумышленники используют чаще всего). К примеру, пользователь должен передать строку в 5 символов. Для этих целей в используемом скрипте выделен объем памяти для хранения именно такого размера данных.

Например, пользователь отправил сто символов. В случае если во время приема информации программой не будет проверен размер принимаемого блока, то во время записи полученных данных в буфер обмена сервера их размер выйдет за пределы отведенного объема памяти и будет записан поверх кода. Это означает, что, вероятнее всего, скрипт будет запорчен и не сможет выполнить возложенных на него обязанностей. Скорее всего, это приведет к зависанию его работы. Результатом станет то, что сервер не будет посылать ответы на запросы клиентов, а значит, будет совершена классическая DoS-атака выполненная при помощи переполнения буфера.

В итоге получается, что сайт не взломан, и информация на нем нетронута, вот только веб-сервер больше недоступен по сети.

В целом, для перегрузки всех ресурсов атакуемого сервера злоумышленнику ничего не надо знать. Ресурсы абсолютно любого компьютера вполне ограничены. В подобной ситуации злоумышленнику следует выбрать наиболее «слабое звено» веб-сервера (к примеру, процессор или канал связи) и там вызвать всплеск нагрузки [4, стр. 32-33].

Сейчас атаки DoS превратились в мощное оружие террористов в киберпространстве. Ведь сегодня значительно проще нарушить правильность функционирования сети или же системы, нежели получить доступ к ней. Существует множество программных средств, позволяющих реализовать DoS-атаку.

Выделяют четыре основных типа атак DoS.

1. Bandwidth consumption или насыщение полосы пропускания. Иначе говоря, злоумышленник заполняет всю доступную в определенной сети полосу пропускания. Конечно же, подобную атаку можно осуществить в локальной сети, но все же чаще злоумышленники производят захват ресурсов удаленно.

2. Recourse starvation или недостаток ресурсов. Данная атака отличается от вышеприведенной тем, что направлена на захват ресурсов системы, к примеру, центрального процессора, пространства жесткого диска, памяти и пр. системных процессов. Подобная атака ведет к тому, что система или же пользователи будут испытывать, как минимум, недостаток в тех ресурсах, которые используются совместно. Подобные атаки чаще всего приводят к продолжительной недоступности ресурса, переполнению файловой системы, зависанию различных процессов, а иногда и к краху системы в целом.

3. Programming flaw или ошибки программирования. Основой этой атаки является неспособность приложения, логической микросхемы или же операционной системы обрабатывать возникающие исключительные ситуации. Чаще всего подобные ситуации возникают во время передачи уязвимому элементу каких-либо несанкционированных данных. Во время такой атаки взломщик много раз пытается передать пакеты, в которых не

учтены рекомендации к документам RFC, для того, чтоб определить, может ли сетевой стек справиться с такого рода исключениями или же получение подобных пакетов приведет к панике ядра (называемой kernel panic) и последующему возможному краху системы.

4. Атака DNS и маршрутизация. Подобные атаки DoS основаны на манипуляциях с записями в таблице маршрутизации. Это вполне может привести к прекращению обслуживания различных легитимных сетей или систем. В большинстве своем протоколы маршрутизации вообще не имеют или же пользуются слабыми алгоритмами аутентификации. Собственно говоря, именно этот факт и дает возможность взломщикам изменять маршруты. Чаще всего они указывают ложный исходных IP-адрес, а это, в свою очередь, приводит к отказу в обслуживании. Результатом подобной атаки становится то, что целевой трафик маршрутизируется либо через сеть самого взломщика, либо в сеть, которой фактически не существует. Атаки, которые направлены на сервера DNS, также считаются довольно эффективными. В большинстве своем такого рода атаки приводят к кэшированию фиктивных адресов на целевом сервере. Атаки на DNS-сервера ведут к тому, что большие узлы довольно долго оказываются недоступными для своих многочисленных пользователей [5, стр. 513-515].

При помощи любого из приведенных видов DoS-атак очень непросто вывести из строя большие веб-сайты, по той причине, что они для своей работы обычно используют широкие каналы в паре со сверхмощными серверами. Практика показывает, что хакеры вполне могут найти выход из практически любой ситуации [4, стр. 35].

Информацию, которая необходима для выявления признаков появления интенсивных злонамеренных запросов к серверу, можно найти в специальных журналах. Эти журналы ведутся программами защиты, установленными на веб-сервере. Среди такого рода журналов, предоставляющих информацию об активности пользователей, можно выделить:

1. Журнал регистрации системных событий. К примеру, журнал безопасности, который ведется операционной системой. В таком журнале регистрируются события аудита. Также следует обратить внимание и на журнал действий пользователя, который создается программами, относящимися к разряду компьютерной полиции (например, к подобным программам можно отнести клавиатурного регистратора STARR);

2. Журнал сетевого трафика. В такого рода журналах ведется запись всего трафика, как входящего, так и исходящего. Подобного рода запись производится при помощи специализированных программ, к примеру, такой утилиты как TCPDump;

3. Сообщения от программы обнаружения вторжений, записывающиеся в режиме реального времени. Подобного рода программы носят название Систем обнаружения вторжений в режиме реального времени или Intrusion Detection System (IDS). IDS представляют собой специализированные программы, которые в состоянии, в реальном времени, отслеживать весь сетевой трафик. Целью работы подобной программы является обнаружение признаков вторжения. Программа руководствуется шаблонами хакерских атак. Чаще всего, системы IDS ведут журналы, в которых регистрируют события безопасности. Эти журналы необходимы для того, чтоб пользователи могли проанализировать ситуацию самостоятельно. Примером такого рода программы является довольно популярная утилита под названием Blacklce Defender [6, стр. 28].

1.2.3 Результаты анализа хаотичных интенсивных запросов

Подытоживая все вышесказанное можно сделать вывод о том, что наблюдать за активностью пользователей необходимо постоянно. Дело в том, что всплеск активности пользователей, вполне может вызвать большую нагрузку на оборудование веб-сервера и тем самым привести к тому, что сайт перестанет отвечать на запросы пользователей. Кроме того, многочисленные интенсивные запросы вполне могут создаваться разного рода программным

обеспечением, которое используется злоумышленниками во время сканирования сайта, поиска уязвимостей в его скрипте и т.д. Также интенсивная нагрузка, т.е. множество запросов к веб-сайту, может быть создана программой, подбирающей логины и пароли пользователей.

Для того чтоб избежать нарушения работоспособности веб-сервера необходимо разработать методы защиты от хаотичных интенсивных запросов.

Напоследок отметим, что нагрузку на веб-сервер также могут создавать поисковые роботы, посещающие сайт во время индексации. Конечно же, запросы, создаваемые поисковыми роботами, никак нельзя назвать злонамеренными, но все же они подчас, за счет интенсивности отправляемых запросов, также могут создать довольно высокую нагрузку на веб-сервер.

2 Разработка методов защиты от сканирования и хаотичных интенсивных запросов

В истории существования сети Интернет есть сведения о многочисленных разрушительных атаках произведенных на веб-сайты, когда нападающие смогли не только извлечь ценную информацию о структуре самого сайта, но и получить привилегированный доступ к нему. В то же время данные атаки почти не затронули огромный айсберг ошибок, допускаемых множеством разработчиков. Многие веб-разработчики до сих пор создают сайты, не обладающие должным уровнем защиты [3, стр. 390].

В целом, проблема защиты веб-ресурса не может ограничиваться одной лишь защитой сценария. Можно написать самый безопасный скрипт, но при этом расположить его на сервере, который будет иметь операционную систему с настройками по умолчанию. Дело в том, что настройки установленные по умолчанию практически всегда далеки от идеала. Именно за счет этой их особенности такой сервер, вполне может быть взломан без использования сложных программных средств.

Безопасным обязательно должен быть не только каждый из участков написанного кода, но и каждая из программ, установленных на веб-сервере, а также сама операционная система вместе со всем используемым оборудованием (данное утверждение, естественно, касается разного рода сетевых устройств, используемых в работе веб-сервера).

Программисту всегда нужно сотрудничать с администратором веб-сервера, а также со специалистами по обеспечению безопасности. К примеру, программист может принять решение, что для его удобства нужно создать определенную папку, которая будет открыта всем пользователям для чтения, а также записи. В данной папке скрипт будет сохранять определенные данные. Вот только если эту папку использует также администратор и хранит в ней важные данные или же конфигурационные файлы, то веб-сервер, естественно, окажется под угрозой.

Подытоживая все вышесказанное, отметим, что защищать нужно не только скрипт самого сайта, но и операционную систему всего веб-сервера, а также сервер баз данных и, конечно же, используемое веб-сервером сетевое оборудование [8, стр. 111-112].

Прежде всего, администратору сервера необходимо обратить свое внимание на информацию об уже обнаруженных уязвимостях. Для их устранения необходимо постоянно обновлять используемое программное обеспечение, а также изменить конфигурацию используемого сервера.

Кроме того, в работе веб-сервера не следует использовать уязвимые сценарии. Для нахождения различных уязвимостей можно использовать коммерческие системы IDS. В случае если в системе используется NFR можно применить сценарий ncode, которому под силу выявить атаку вида PHF на сервер.

Кроме того, существует несколько видов сценариев, которые используют так называемый метод «ловли на живца». Такого рода сценарии сами выступают в роли наживки или обманного сценария PHF. Они отвечают атакующему компьютеру, в процессе его вторжения, и одновременно собирают о нем разного рода сведения. Правда такой метод подойдет только самым бесстрашным администраторам веб-серверов.

В целом, можно выделить два метода борьбы с обнаруженными уязвимостями: это либо удаление обнаруженного уязвимого сценария (замена его более новой версией), либо установка отдельного сценария-заплатки. Каждый администратор сервера сам решает, какой из этих способов обеспечения безопасности ему ближе [3, стр. 382-390].

В этой главе нам необходимо разработать методы защиты содержимого сайта от сканирования, а также защиты сайта от разного рода хаотичных интенсивных запросов.

2.1 Методы защиты от сканирования

Исходя из приведенного в предыдущей главе анализа процесса сканирования сайта, а также программного обеспечения, используемого для этих целей, несложно сделать вывод о том, что сам по себе процесс сбора необходимой злоумышленнику информации в основном основывается на недочетах, оставленных программистами. Все дело в желании облегчить и максимально ускорить процесс, как просмотра, так и отладки работы сайта. Именно благодаря таким недочетам, у анонимного пользователя вполне может появиться шанс получить доступ к какой-либо важной информации или же к важным функциям сайта. К такому исходу могут привести ошибки в конфигурации веб-ресурса, а также непродуманность общего построения узла [2, стр. 136-137]. Прежде всего, следует позаботиться о защите от сканирования сайта при помощи программы-краулера. Начнем, пожалуй, с того, что нередко такого рода программы оказываются в тупике, если встречают необычный для себя метод перенаправления или же связывания ресурсов. В то же время, некоторая часть программ-краулеров может верно распознать такого рода аномалии и предоставить неплохие результаты.

К примеру, проблема плохого распознавания перенаправления программой-краулером вполне может возникнуть в случае, если функции данного перенаправления возложены на клиентский сценарий, написанный на JavaScript или же на VBScript.

Ниже приведен фрагмент кода на JavaScript в котором используется директива перенаправления или же вызова метода объекта location. Именно эту директиву сначала интерпретирует, а после, соответственно, выполняет веб-браузер клиента.

```
<script language=”JavaScript”>
location.replace(“./index.php3”);
</script>
```

Результатом обработки данного фрагмента кода станет то, что браузер обратиться к ресурсу с именем index.php3. Конечно же, подобное перенаправление сработает лишь в том случае, если в самом браузере не запрещено выполнение клиентских сценариев, написанных на JavaScript. Одновременно программа-краулер не сможет правильно интерпретировать, а значит и выполнить, такой оператор как location.replace(). Именно по этой причине во время сканирования сайта при помощи программы-краулера, файл с именем index.php3 окажется пропущенным.

В то же время, если такого рода перенаправление будет выполняться с использованием дескриптора <meta> языка гипертекстовой разметки или же с применением заголовка ответа Content-Location, то программа-краулер сумеет это обнаружить, а, значит, просканирует и проведет анализ страницы на которую установлено перенаправление.

Отметим, что некоторое количество из существующих программ-краулеров все же снабжено необходимыми функциями, для проведения столь нужного синтаксического анализа, обеспечивающего корректную обработку всех директив перенаправления. Но все же часть такого рода программ, к примеру, утилита под названием wget не может обрабатывать включенные в документ HTML-дескрипторы [1, стр. 186-187].

Для того чтоб обезопасить сайт от сканирования необходимо создать максимально стойкую защиту для веб-узла. В большинстве своем утечки информации могут быть предотвращены благодаря поддержанию жесткой политики безопасности, а также благодаря устранению ошибок в конфигурировании сайта. О безопасности ресурса должен заботиться не только администратор веб-сервера, но и программист. В частности программисту необходимо позаботиться о защите каталогов. Дело в том, что возможность получить список каталогов, а также просмотреть файлы, содержащиеся в каталоге или же определить внутренний IP-адрес, благодаря полю Location находящемуся в заголовке HTTP, даст злоумышленнику массу информации, помогающей исследовать веб-приложение. Именно поэтому, во

время разработки скриптов необходимо придерживаться принципа минимизации объемов информации, которая может оказаться доступной злоумышленнику. Подобный подход позволит существенно повысить общую защищенность всего веб-ресурса.

Во время сканирования сайта злоумышленник особое внимание уделяет файлу robots.txt, из которого он может почерпнуть информацию о расположении каталогов на сайте.

Наличие данного файла существенно снижает для злоумышленника трудоемкость полного исследования структуры каталогов сайта. Ведь в файле robots.txt содержится список множества каталогов, который предназначается для поисковых машин. Эти каталоги либо разрешены, либо запрещены к индексированию. Относительно разрешения или запрета на индексирование, априори можно сделать вывод о полезности определенного каталога для проведения исследования сайта. Ведь файл robots.txt, который может быть получен либо с сервера поисковой машины, либо непосредственно с веб-узла, обеспечивает отличное представление о внутренней структуре каталогов всего веб-узла. Собственно говоря, это одна из целей сканирования сайта.

Единственным способом избавиться от данной «подсказки» для злоумышленника является отказ от использования файла ограничения доступа к информации для различных поисковых роботов. Использовать файл robots.txt не обязательно. Иначе говоря, помещение файла robots.txt в корневой каталог сайта дело добровольное [2, стр. 120-138].

Для того чтоб сделать невозможным сканирование сайта при помощи поисковой системы необходимо запретить для поисковых роботов сканирование закрытых частей сайта. Для этого можно использовать не только файл robots.txt, но и основывать защиту закрытых областей сайта на криптографическом протоколе SSL, а не только на простой проверке пароля доступа, вводимого пользователем сайта при входе в защищенную область. Ведь в последнем случае поисковая система проиндексирует «закрытые» для

простых пользователей страницы. В таком случае в базе данных поисковой системы может оказаться важная информация, скрытая от простых пользователей, но по ошибке администратора ставшая доступной роботу индексирующей машины [4, стр. 25].

Для того чтоб усложнить работу приложений, предназначенных для проверки определенных веб-сайтов на наличие в них широко известных или же формируемых в качестве умолчания, уязвимых мест, администратору веб-сервера необходимо не только изменить конфигурационные параметры веб-сервера по умолчанию, но и удалить (или же обновить) файлы, в которых была обнаружена уязвимость.

В целом, лучше удалить все сценарии, которые не задействованы в работе веб-сервера. На сервере должны работать лишь те сценарии, которые необходимы для его работоспособности.

Иначе говоря, все лишнее должно быть удалено. В то же время, если удалить или же обновить сценарии нет возможности, то необходимо установить так называемую «заплатку». Подобный подход, конечно же, нельзя назвать панацеей. Именно поэтому администратору также необходимо ограничить доступ к исходным файлам сервера. В частности, нужно отменить право на чтение для большинства групп пользователей. В целом, для исходного кода сценария вполне достаточно будет разрешения на исполнение файла. Право на чтение файла в такой ситуации необязательно.

Следующие советы дадут возможность администратору ресурса защитить сайт от различных технологий его исследования:

1. Необходимо помещать все клиентские сценарии, написанные на языке JavaScript в определенный отдельный каталог. При этом администратору следует убедиться в том, что у всех скриптов, которые размещены в данном каталоге, отсутствует разрешение на выполнение. Иначе говоря, данные файлы могут быть лишь прочитаны сервером, но не могут быть им выполнены в качестве сценариев.

2. В исходном коде не должно оставаться комментариев, выводимых в

браузер клиента. Дело в том, что в исходном коде программы могут иметься переменные, невидимые во время обыкновенного просмотра Интернет-страницы. Эти переменные предназначаются для упрощения процесса отладки работы скрипта.

3. В случае если скрипт должен вызывать какой-либо другой файл, расположенный на веб-сервере, необходимо использовать имена путей относительно текущего каталога или же корневого каталога веб-сервера. Нельзя использовать в строках имен путей существующие названия каталогов (которые находятся вне корневого каталога сервера) или логических дисков веб-сервера. К тому же, необходимо учесть то, что написанный сценарий должен самостоятельно отбрасывать те символы, которые предназначаются для обхода каталогов, в частности ../, ./ и пр.

4. В случае если необходимо использовать аутентификацию, то программисту нужно обеспечить возможность добавочной аутентификации для получения доступа к каталогам и подкаталогам. Иначе говоря, если анонимный пользователь не имеет доступ к PHP-файлам, то значит, он не сможет получить доступ и к XSL-файлам [2, стр. 138].

В целом, контрмеры, которые помогут предотвратить кражу данных с веб-сайта, сформулированы ниже.

Необходим мониторинг всех действий пользователя. Он поможет выявить быстрые повторные запросы GET исходящие из одного источника. Такого рода запросы, естественно, необходимо блокировать на некоторое время. В подобной ситуации сканирование сайта окажется не только затрудненным, но и продолжительным.

Для анализа и разработки наиболее эффективных мер, противодействующих исследованию веб-ресурса, сведем все вышеприведенные методы защиты сайта от сканирования в таблицу 2.1.

Таблица 2.1 – Виды сканирования сайта и методы защиты от проведения сканирования

Вид исследования	Метод противодействия	Эффект от использования метода
1	2	3
Ручное сканирование содержимого сайта	1. Установка временной задержки между запросами, исходящими от одного пользователя 2. Использование на веб-сервере мощного модуля под названием mod_rewrite, дающего возможность создания «дружественных адресов» или ЧПУ 3. Отказ от использования robots.txt	1. Возможность существенно замедлить проведение сканирования сайта 2. Возможность скрыть структуру сайта, а также оставить в тайне от злоумышленника принцип формирования URL-адресов на ресурсе 3. Усложнит изучение структуры веб-ресурса, с одной стороны, правда, может привести к некоторым проблемам с индексацией содержимого сайта
Сканирование сайта при помощи программы-краулера	1. Установка временной задержки между запросами, исходящими от одного пользователя 2. Использование на веб-сервере мощного модуля под названием mod_rewrite, дающего возможность создания «дружественных адресов» или ЧПУ 3. Отказ от использования robots.txt	1. Возможность существенно замедлить проведение сканирования сайта 2. Возможность скрыть структуру сайта, а также оставить в тайне от злоумышленника принцип формирования URL-адресов на ресурсе 3. Усложнит изучение структуры веб-ресурса, с одной стороны, правда, может привести к некоторым проблемам с индексацией содержимого сайта
	5. Использование нестандартных способов перенаправления	5. Часть программ-краулеров не сможет обработать такие перенаправления, а значит не справится с поставленной задачей.
Исследование сайта при помощи поисковой системы	1. Для закрытых частей сайта необходимо использовать криптографический протокол SSL 2. Правильно настроить разрешенные для сканирования поисковым роботом страницы в файле robots.txt	1. Роботу поисковой системы не удастся обойти такого рода защиту, а значит «закрытая» страница не попадет в индексную базу поисковой системы 2. Робот поисковой системы будет игнорировать те каталоги, которые запрещены к индексированию в файле robots.txt
Исследование сайта при помощи утилит для автоматического поиска известных	1. Установка временной задержки между запросами, исходящими от одного пользователя 2. Регулярное обновление (замена или же удаление) используемых на	1. Возможность существенно замедлить проведение сканирования сайта 2. Регулярное обновление используемых скриптов, даст возможность устранить найденные в этих скриптах уязвимости

уязвимостей	сайте скриптов, в которых была	
	обнаружена уязвимость 3. Проведение регулярного аудита сайта при помощи подобных программ	3. Эта мера даст возможность выявить уязвимости сайта «заблаговременно» и, соответственно, принять меры, связанные с их ликвидацией

Анализируя информацию, приведенную в таблице 2.1 можно отметить, что наиболее эффективным методом противодействия всем видам сканирования и исследования сайта является установка некоторой временной задержки между различными запросами, исходящими от одного пользователя (следует делать привязку к IP-адресу пользователя). Иначе говоря, в определенный промежуток времени, одному пользователю удастся отправить небольшое ограниченное количество запросов к страницам сайта. Это, с одной стороны, позволит существенно замедлить (или даже сделать невозможным) процесс сканирования и исследования сайта, а с другой стороны, данный метод не будет создавать серьезных неудобств для простых пользователей, не ставящих перед собой цель исследовать структуру сайта.

2.2 Методы защиты от хаотичных интенсивных запросов

Исходя из приведенного в предыдущей главе анализа различных видов хаотичных интенсивных запросов, проведем исследование существующих методов защиты от таких действий пользователей и от создания такого рода нагрузки на веб-сервер.

Защитить сайт от случайных интенсивных запросов, исходящих от пользователей можно установив небольшую временную задержку между запросами, исходящими от одного пользователя. При этом привязка к пользователю должна быть сделана с учетом его IP-адреса. В то же время, следует учесть тот факт, что такого рода ограничение не должно действовать на администратора сайта или же на модераторов ресурса. Правда, подобная сдерживающая мера, вполне возможно, создаст определенные неудобства

для некоторой, вероятнее всего, небольшой части пользователей сайта.

Рассмотрим методы защиты от различных видов хаотичных интенсивных запросов.

Метод подбора паролей используется злоумышленниками довольно часто. Данный процесс не только занимает массу времени, но и сильно нагружает веб-ресурс. В целом, взлом шифров при помощи перебора различных вариантов очень утомительное занятие, особо, если учесть тот факт, что для того, чтоб получить один зашифрованный текст нужно потратить массу процессорного времени компьютера.

Подбор пароля довольно долгий процесс. В то же время, если выбран на самом деле сложный и длинный пароль, то даже наличие самого лучшего словаря не поможет злоумышленнику. Правда, в такой ситуации также не стоит расслабляться, ведь злоумышленник может воспользоваться способом полного перебора всех символов. Данный процесс займет значительно больше времени, но все же в итоге, метод даст возможность получения положительного результата. Администратору веб-сервера, для того, чтоб этого избежать, необходимо установить систему обнаружения различных хакерских атак. В целом, хороший сетевой экран способен выявить попытки подбора пароля пользователей. После чего данная программа сообщит администратору веб-сервера об этом событии. Правда, следует учесть тот факт, что в большинстве своем администратор веб-сервера не слишком интересуется обеспечением безопасности сайта, расположенного на его сервере.

Во время полного перебора сложного пароля злоумышленнику может понадобиться очень много времени. Подобные временные затраты, вполне возможно, окажутся несопоставимыми с получаемым результатом.

Кроме того, за время подбора такого пароля пользователь вполне может его изменить, если, к примеру, меняет свои пароли через каждые пару недель. Вот так и получается, что одним из способов защиты от подбора пароля методом полного перебора может стать регулярная смена кода

доступа, например, раз в месяц. Отметим, что часть систем предусматривает установку определенных политик безопасности, которые принуждают пользователей время от времени менять свои пароли. В случае если программным обеспечением веб-сервера предусмотрена подобная возможность, администратору сервера следует ею воспользоваться. Эта мера обязательно повысит общий уровень обеспечения безопасности всего веб-сервера.

Подбор пользовательского пароля будет существенно упрощен для злоумышленника, в случае, если ему станет известно имя учтенной записи пользователя [4, стр. 28-29].

Самой эффективной контрмерой, направленной на борьбу против подбора паролей считается комбинация хорошо продуманной политики по помощи в выборе паролей пользователей, а также взвешенной политики блокировки учетных записей. Для того чтоб риск такого рода взлома был сведен к минимуму, приложение должно производить блокировку учетной записи после того, как было совершено несколько неудачных попыток входа. В то же время не следует впадать в другую крайность – администратору следует помнить о том, что параноидальная политика блокировки учетных записей пользователей вполне может привести к такому состоянию, как отказ в обслуживании. К тому же, воспользовавшись подобным изъяном, злоумышленник сможет просто заблокировать все имеющиеся учетные записи пользователей. Большинство разработчиков отдают предпочтение золотой середине – либо блокируют учетные записи на некоторое время, к примеру, на десять минут, либо не дают отправлять пользователю подряд множество запросов на авторизацию, к примеру, устанавливая между попытками авторизоваться небольшую временную задержку. Подобный подход дает возможность довольно эффективно противостоять попыткам подобрать логин и пароль пользователя, а также существенно снижает общую производительность и, соответственно, эффективность работы злоумышленника. Использование строгой политики по помощи

пользователям в выборе пароля обеспечит существенное снижение вероятности случайно угадать регистрационные данные пользователя. Установка длины пароля не менее шести символов в комбинации с хорошей и продуманной политикой блокировки учетных записей пользователей дают возможность успешно противостоять разного рода попыткам взлома учетных записей пользователей путем перебора всевозможных вариантов.

Кроме того, как ранее упоминалось, множество программ подбора пользовательских паролей вполне может быть нейтрализовано путем использования нестандартных сообщений, в процессе произведения формоориентированной аутентификации. Такого рода прием легко сможет существенно затруднить использование программных инструментов разного рода для подбора пользовательских паролей [2, стр. 164].

Еще одним видом интенсивных запросов является флуд. Данная атака состоит в том, что злоумышленник отправляет на веб-сайт огромное количество различной бессмысленной информации. Флуд может производиться против тех скриптов, которые в состоянии получать различную текстовую или графическую информацию, а также отображать ее на веб-страницах.

Такими скриптами считаются:

- форумы;
- формы обратной связи;
- чаты;
- гостевые книги;
- формы, на которых пользователь может посмотреть, а также оставить собственные комментарии.

Для осуществления флуда злоумышленник может использовать программу или же сценарий, целью которых будет направление бесконечного потока информации на веб-сервер. Чаще всего, задача флуда – это заполнение базы данных или же окна чата многочисленными бесполезными сообщениями. Отметим, что проблема борьбы с флудом

существенно усложняется в случае, если скрипт веб-сайта дает возможность отправлять анонимные сообщения всем гостям и пользователям сайта.

Существует довольно много методов защиты сайта от флуда. Ниже приведем некоторые из них:

1. Сообщения разрешено оставлять лишь зарегистрированным пользователям. В такой ситуации защититься от флуда будет значительно проще, ведь злоумышленнику нужно будет постоянно регистрироваться.

2. Запрет на получение сообщений от одного и того же пользователя с определенным IP-адресом на протяжении определенного промежутка времени. В таком случае, у злоумышленника будет возможность «пофлудить» (отправить какое-либо сообщение) не чаще, нежели один раз в заранее заданный промежуток времени.

3. В некоторых ситуациях можно разрешить пользователю отправлять одновременно несколько сообщений кряду, но при этом необходимо предусмотреть ограничение такого типа: сообщений должно быть не больше 5 штук в минуту. При этом промежуток времени между сообщениями учитываться не будет. В целом, среднестатистическому пользователю написать и отправить более 5 сообщений на сайт, за одну минуту, не представляется возможным, а значит данная мера предосторожности «усложнит жизнь» лишь злоумышленникам создающим бессмысленные сообщения.

4. Если веб-сайт отличается высоким уровнем посещаемости и привязка к уникальному IP-адресу просто не представляется возможной, то программисту приходится привязываться к файлам Cookies, в которых сохранять информацию о последнем отправленном пользователем сообщении.

5. Запрет на создание одинаковых сообщений. Имеется ввиду то, что от одного и того же пользователя сервер не должен принимать два абсолютно одинаковых сообщения, вне зависимости от промежутка времени между ними. Дело в том, что создание абсолютно одинаковых сообщений можно

считать бессмысленным.

Вышеприведенные методы, дадут возможность выстроить приемлемый уровень обороны сайта от флуда со стороны пользователей. Правда, будет ли этот уровень достаточным чтобы решать поставленные задачи. Это зависит, прежде всего, от того, как именно будут реализованные настройки. К примеру, в случае, если создать запрет на отправку более чем 5 сообщений от одного пользователя за 10 секунд, то флудер сможет направить не более 30 сообщений за одну минуту. Правда, администратору, стоит учесть тот факт, что злоумышленник может быть не один, или же он будет использовать анонимные прокси-сервера, для достижения поставленной цели.

В случае привязки к IP-адресу пользователя не следует забывать о том, что существуют обширные сети, пользователи которых в Интернете видны под одним и тем же IP-адресом (NAT с перегрузкой). В то время как реально в подобной сети может иметься тысяча или даже десять тысяч различных компьютеров. В подобном случае работа множества пользователей, такого рода сетей, на защищаемом веб-сайте окажется затрудненной. В то же время злоумышленник запросто обойдет такого рода препятствие, для этого он воспользуется помощью нескольких различных анонимных прокси-серверов.

Конечно же, всегда можно использовать пару, созданную IP-адресом и файлами Cookies. Правда, последние могут быть легко удалены с компьютера-клиента, а значит, их нельзя считать надежным решением. В целом, самым простым вариантом будет разрешение оставлять сообщения лишь зарегистрированным пользователям. В противном случае необходимо усложнить систему безопасности сайта, что в свою очередь усложнит использование ресурса рядовыми пользователями [4, стр. 62-66].

В целом, самым лучшим вариантом защиты от флуда сообщениями является запрет на отправку нескольких сообщений подряд с одного и того же IP-адреса. Для этого программисту вполне можно реализовать следующую логику в скриптах защиты сайта:

1. После того, как пользователь отправил сообщение адрес посетителя,

также как и текущее время сохраняются в таблице базы данных. Сохранять IP-адрес отправителя нужно именно на сервере, по той причине, что все то, что хранится на компьютере клиента, вполне может быть уничтожено буквально за пять секунд, а порой и быстрее. Для того чтоб определить адрес пользователя отправившего сообщение или запрос, вполне можно использовать переменную окружения REMOTE_ADDR:

$_SERVER[“REMOTE_ADDR”]

2. В процессе принятия сообщения, исходящего от пользователя, нужно удалять из базы данных все IP-адреса пользователей, у которых время хранения превышает некоторое, заранее определенное, количество минут. Обычно для этих целей вполне достаточно установить интервал равный двум минутам серверного времени.

3. После отправки сообщения скрипту необходимо проверить, остался ли внесенный в базу данных IP-адрес. Если остался, значит, обрабатывать сообщение не следует. Вместо этого нужно выдать пользователю сообщение об ошибке, с текстом вроде «Нельзя отправлять несколько сообщений за интервал меньше, чем две минуты».

Практика показывает, что злоумышленники не станут «флудить» на сайте, который оснащен такого рода защитой. Дело в том, что в такой ситуации флуд займет слишком много времени, в то время как эффект от него будет минимален, ведь много сообщений подряд одному пользователю просто напросто не удастся отправить [8, стр. 180-181].

Одной из разновидностей злонамеренного флуда считается атака на отказ в обслуживании.

Большинство атак на отказ в обслуживании может быть отражено довольно просто. Программист должен предусмотреть возможность того, чтоб скрипт сайта автоматически контролировал и ограничивал количество запросов, исходящих от одного IP-адреса в определенный промежуток

времени. Правда, подобный подход сможет защитить лишь от начинающих хакеров. Дело в том, что опытный взломщик без проблем сможет подделать собственный IP-адрес, после чего засыпать сервер многочисленными запросами, в которых указывается поддельный адрес их отправителя.

Наиболее сложным считается установка защиты от перегрузки пользовательского канала. Вся проблема в том, что когда на веб-сервер идет множество пакетов, то нет возможности отфильтровать их без получения. Иначе говоря, защититься от перегрузки канала можно лишь расширением канала. При этом канал должен быть настолько широк, чтоб его, одному компьютеру, просто невозможно было заполнить [4, стр. 35].

Интенсивные запросы к страницам сайта могут также создаваться поисковыми роботами.

С одной стороны, индексация сайта необходима, а с другой – порой интенсивность различных запросов, исходящих от посещающих сайт поисковых роботов становится очень высокой и может довольно сильно повлиять на работу всего веб-сервера.

В такой ситуации администратору сайта необходимо предусмотреть методы защиты сайта от интенсивных запросов, исходящих от различных поисковых роботов.

Для этих целей можно использовать файл robots.txt, расположенный в корневой директории веб-ресурса. Файл robots.txt необходим для частичного управления индексированием всего сайта. В частности, с его помощью можно дать указание поисковому роботу загружать страницы сайта с определенной интенсивностью.

К примеру, синтаксис

```
User-agent: *
Crawl-delay: 10
```

говорит о том, что поисковый робот (принадлежащий любой поисковой

системе) должен загружать страницы сайта с интервалом в 10 секунд. В такой ситуации нагрузка на веб-сервер, создаваемая поисковым роботом, окажется минимальной [2, стр. 138].

Кроме того, администратор веб-ресурса может создать файл Sitemap.xml. Ссылку на карту сайта необходимо поместить в файл robots.txt при помощи такой строки:

Sitemap: <sitemap_location>

В созданном, согласно XML-стандарту, файле Sitemap.xml может быть указана дата последней модификации определенной страницы, а также частота обновления информации для каждой из страниц сайта. Пример синтаксиса файла Sitemap.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
<loc>http://example.com/</loc>
<lastmod>2011-01-01</lastmod>
<changefreq>monthly</changefreq>
<priority>0.8</priority>
</url>
</urlset>
```

Использование файла Sitemap.xml помогает поисковому роботу лучше ориентироваться в структуре сайта, а также не загружать в базу данных страницы, содержимое которых не изменилось. А это, в свою очередь, ведет к уменьшению количества запросов, исходящих от поисковых систем и, соответственно снижению уровня нагрузки на сайт [9, стр. 89-90].

2.3 Программные средства защиты сайта

Множество программ выявления вторжений (Instrusion Detection System или IDS) способно обнаружить процесс исследования сайта. Такого рода программы прослушивают весь сетевой трафик и, в случае, обнаружения чрезмерной активности, сигнализируют о ней администратору веб-сервера. Отдельно следует отметить, что системы IDS далеко не всегда выявляют атаки. Дело в том, что в большинстве своем системы IDS по умолчанию конфигурируются на обнаружение шумного и бурного сканирования сайта. Если чувствительность IDS не повысить, то огромный пласт атак останется незамеченным.

Большинство систем IDS легко определяют атаки направленные на выявление наличия различных уязвимостей в системе. Кроме того, установка подобной системы поможет в борьбе с атаками на отказ в обслуживании. Ведь система будет блокировать повторяющиеся запросы, если их частота превысит допустимую норму [3, стр. 291-300].

Системы обнаружения вторжений могут распознавать сигнатуры различных враждебных действий непосредственно в процессе их выполнения. После обнаружения сигнатуры враждебных действий, система либо будет использовать соответствующие контрмеры, которые либо устранят точку уязвимости, либо заблокируют выполнение враждебных действий в этой точке. [7, стр. 41].

Системы выявления вторжений предназначаются для выявления атак, а также уведомления о них в режиме реального времени. Чаще всего системы IDS состоят из трех частей: модуля мониторинга, модуля логического вывода, а также модуля оповещения. На модуль мониторинга возложена задача сбора данных о текущей деятельности (чаще всего подобные данные характеризуют различный сетевой трафик). Те данные, которые были собраны модулей мониторинга, после передаются модулю занимающемуся логическим выводом. Данный модуль анализирует полученную информацию

и определяет, порождена ли сетевая активность нормальными или же злонамеренными действиями. Отметим, что с каждой атакой связана некая, характерная именно для нее сигнатура, т.е. шаблон, по которому система данную атаку распознает и классифицирует. Для выявления хакерских атак в большинстве существующих систем IDS используются сигнатуры. После того, как атака обнаружена, модуль оповещения генерирует некий ответ, который основывается на внесенных заранее конфигурационных параметрах системы. реакция IDS на атаку может быть либо пассивная, либо активная. Если система IDS в качестве ответного действия направляет сообщение администратору или же вносит запись в свой системный журнал, то подобную реакцию называют пассивной. А вот активная реакция системы представляет собой передачу сообщению брандмауэру о необходимости заблокировать сетевой трафик, который инициирован взломщиком.

Одной из ключевых проблем, связанных с системами IDS, можно считать необходимость обеспечить высокую точность результатов данной системы. Дело в том, что системы выявления вторжений далеко не всегда хорошо справляются с возложенными на них функциями. Выделяют два вида ошибок систем IDS: несрабатывание системы в случае атаки (false negative) или же ложное срабатывание (fake positive). Ложное срабатывание системы происходит в случае, если IDS расценивает обычные действий пользователей как атаку, хотя это не так. А вот несрабатывание системы в случае атаки происходит тогда, когда реальная атака остается неклассифицированной.

Идеальной системы IDS, которая смогла бы обеспечить отсутствие ложных срабатываний, а также идентификацию абсолютно всех атак, просто не существует. Если хакеру необходимо обойти систему IDS он либо маскирует атаку, делая ее незаметной для системы, либо при помощи автоматизированных средств генерирует массу ложных срабатываний, тем самым, заставляя администратора сервера принять меры по снижению чувствительности системы [1, стр. 334-336].

На системы обнаружения вторжений (IDS) возложена функция

предупреждения администратора о возможной атаке или же попытке проникновения злоумышленника в сеть. Системы защиты от вторжений (Intrusion Prevention Systems или IPS) вполне могут быть настроены для противодействия большинству атак, например, для осуществления блокировки связи с источником атаки.

В целом, рекомендации по использованию как IDS, так и IPS систем одинаковы:

– оба типа систем могут стать эффективными дополнительными средствами защиты веб-сервера как от атак снаружи, так и от злонамеренных действий изнутри;

– IDS и IPS могут представлять собой программы, компьютерные системы или же специализированные устройства. В любом процессе обнаружения и противодействия вторжениям, системам необходимо производить анализ журналов процессов и безопасности [10, стр. 128].

Одной из самых популярных бесплатных систем обнаружения и защиты от вторжений является система Snort. Принцип работы этой системы основан на использовании сигнатур, а также комбинации правил и препроцессоров для анализа трафика. В целом, эти правила предоставляют пользователю возможность простого создания сигнатур предназначенных для исследования отдельных пакетов. Препроцессоры системы Snort дают возможность выполнять глубокое исследование входящей и исходящей информации, а также проводить обработку данных, которая не может быть осуществлена при помощи одних только правил. При помощи препроцессоров можно выполнить массу задач, к примеру, дефрагментацию IP-дейтограмм, сборку протоколов TCP-данных, обнаружение сканирования портов сервера и пр. Препроцессоры дают возможность системе просматривать и обрабатывать потоки данных и они имеют существенное отличие от используемых правил, ведь последние предназначены для проведения анализа одиночных пакетов. Недостатком Snort является пассивность системы. Единственное, что может эта IDS – это записать

событие в лог-файл. Т.е. придется ежедневно просматривать логи.

Стоимость коммерческих систем IDS и IPS довольно высока. Кроме того, такого рода системы сложны в настройке и последующем использовании. Это очень громоздкие системы, потребляющие массу системных ресурсов. К тому же, как уже упоминалось, ни одна такая система не дает гарантии абсолютной защищенности, а значит высокий уровень затрат на приобретение и настройку подобной системы может просто не окупиться. В дополнение ко всему сказанному, отметим, что не следует забывать о множестве хакерских атак, направленных на ложное срабатывание подобных систем.

2.4 Разработка методов защиты сайта от сканирования и хаотичных интенсивных запросов

Для анализа и разработки наиболее эффективных мер, которые будут противодействовать хаотичным интенсивным запросам к страницам сайта, сведем все вышеприведенные методы защиты сайта от хаотичных интенсивных запросов в таблицу 2.2.

Таблица 2.2 – Виды хаотичных интенсивных запросов и методы защиты сайта от них

Виды хаотичных интенсивных запросов	Метод противодействия	Эффект от использования метода
Случайные интенсивные запросы	1. Установка временной задержки между запросами, исходящими от одного пользователя	1. Данная мера существенно снизит нагрузку на веб-сервер, правда, вполне может создать определенные неудобства для пользователей ресурса
Подбор пользовательских паролей	1. Установка временной задержки между запросами, исходящими от одного пользователя 2. Установка временной блокировки учетной записи пользователя в случае	1. Возможность существенно замедлить процесс подбора паролей (или даже сделать его невозможным, в случае, если пользователь время от времени меняет свой пароль)

	обнаружения нескольких подряд неудачных попыток авторизации 3. Принуждение пользователей менять свои пароли время от времени, а также предоставление им рекомендаций касательно длины нового пароля и, конечно же, его сложности	2. Эта мера еще больше замедлит процесс подбора пароля, правда, может создать некоторое неудобство для пользователя 3. Вкупе с предыдущими мерами сделает процесс подбора паролей невозможным, правда, может создать некоторое неудобство для пользователей
Флуд со стороны пользователей	1. Установка некоторой временной задержки между отправкой сообщений, исходящих от одного и того же пользователя 2. Запрет на отправку абсолютно одинаковых сообщений от одного пользователя 3. Предоставление возмож-	1. Существенно снизит эффективность флуда, а также очень замедлит данный процесс, что в свою очередь заставит злоумышленника отказаться от поставленных целей 2. Данная мера существенно снизит эффективность флуда
	ности отправки сообщений на страницы ресурса лишь зарегистрированным пользователям	сообщениями, а также замедлит этот процесс, ведь злоумышленнику придется переписывать сообщения для их отправки. Правда, такая мера может создать неудобства для простых пользователей любящих отправлять короткие сообщения, к примеру, вида «+1», или же сообщения, в которых используются лишь смайлы 3. Существенно снизит количество флуда на страницах ресурса, правда, сообщений в целом будет меньше, по той причине, что не все пользователи желают проходить процесс регистрации
Атака на отказ в обслуживании	1. Установка временной задержки между запросами, исходящими от одного пользователя 2. IP-адреса пользователей, постоянно отправляющих запросы к сайту необходимо блокировать	1. Данная мера существенно снизит нагрузку на веб-сервер, а это, в свою очередь, сделает проводимую пользователем атаку на отказ в обслуживании как минимум несостоявшейся 2. Подобная мера снизит нагрузку на сервер, а также запретит доступ к страницам сайта пользователям, имеющим IP-адреса, которые внесены в «черный список». В то же время, злоумышленник сможет воспользоваться

		помощью проки-серве-ра, в то время как простые пользователи, имеющие такой же IP-адрес, лишатся возможности получения доступа к содержимому веб-сайта
Индексация сайта поисковыми роботами	1. Установка в файле robots.txt временной задержки загрузки любой страницы	1. Данная мера поможет существенно снизить уровень нагрузки, создаваемой поис-
	сайта 2. Внесение в файл Sitemap.xml информации о частоте обновления страницы, а также о дате ее последней модификации	ковыми роботами на веб-сервер 2. Эта мера поможет лучше управлять индексацией сайта и снизить количество запросов к страницам от поисковых роботов

Анализируя информацию, приведенную в таблице 2.2 можно отметить, что наиболее эффективным методом противодействия всем видам хаотичных интенсивных запросов является установка некоторой временной задержки между различными запросами, исходящими от одного пользователя (следует делать привязку к IP-адресу пользователя). Иначе говоря, в определенный промежуток времени, одному пользователю удастся отправить небольшое ограниченное количество запросов, на создание сообщений или просмотр содержимого, к страницам сайта. Это, с одной стороны, позволит существенно замедлить (или даже сделать невозможным) создание хаотичных интенсивных запросов, к страницам сайта, а с другой стороны, данный метод не будет создавать существенных неудобств для многочисленных простых пользователей. Если, конечно, эти пользователи не ставят перед собой цель отправить максимальное количество сообщений в небольшой промежуток времени или же создать максимально высокий уровень нагрузки на веб-сервер.

В то же время следует учесть и тот момент, что на часть IP-адресов не должны накладываться какие либо ограничения (например, на IP-адрес администратора сайта). Одновременно, IP-адреса тех пользователей, действия которых можно считать злонамеренными, должны быть внесены в «черный список», иначе говоря – заблокированы. Примером таких

злонамеренных действий может быть постоянная отправка различных запросов к страницам сайта (отправка множества одинаковых сообщений или же сообщений не несущих смысловой нагрузки). Кроме того, злонамеренными могут считаться действия, которые явно направлены на исследование структуры сайта, а также попытки отправить огромное количество бесполезных пакетов информации и пр.

Использовать сложные и дорогие системы IDS и IPS, для защиты сайта от сканирования и хаотичных интенсивных запросов, нецелесообразно ввиду их дороговизны, сложности настройки и высокой ресурсоемкости. А потому вышеприведенный метод защиты следует реализовать путем написания php-скрипта. Преимущества написания подобного скрипта очевидны, ведь такой способ решения проблем, связанных с защитой сайта, будет недорогим, простым в использовании и, конечно же, результативным.

3 Реализация системы защиты в виде php-скрипта

3.1 Данные о реализуемом php-скрипте

В настоящее время на практике используются различные подходы к защите любой компьютерной информации, в том числе и информации, располагаемой на Интернет-ресурсе. Данные подходы могут быть определены следующими характеристиками:

– наличием формализованных требований, как к набору, так и к параметрам всех механизмов защиты, которые регламентируются большинством современных требований к обеспечению общей компьютерной безопасности (иначе говоря, требованиями, определяющими, что именно должно быть предусмотрено разработчиками);

– наличием реальных механизмов защиты, которые реализуются в процессе защиты любой компьютерной информации. К таким механизмам защиты, прежде всего, относя встроенные в операционную систему средства защиты, по той простой причине, что в большинстве своем используемые на веб-сервере скрипты пользуются встроенными в операционную систему механизмами защиты или же наследуют используемые в них методы. Именно на базе этих механизмов и используемых в них методов определяется общий уровень защиты всего веб-сервера;

– существующей статистикой различных угроз безопасности компьютерной информации. В частности, это данные об уже осуществленных успешных атаках на какие-либо информационные ресурсы. Ведение данной статистики призвано помочь определить, насколько эффективны предпринятые меры защиты, а также дать оценку уровню выдвигаемых требований к созданию защиты на веб-сервере [12, стр. 14].

Как уже упоминалось, наиболее эффективным методом для противодействия сканированию сайта, а также всем видам хаотичных интенсивных запросов, является установка некоторой временной задержки

между запросами, исходящими от одного и того же пользователя. При идентификации пользователя основной упор следует делать на его IP-адрес, по той причине, что файлы cookie могут быть легко удалены. Конечно же, IP-адрес пользователя также может быть изменен, к примеру, при помощи прокси-сервера или же с помощью переподключения, в случае, если IP-адрес у пользователя динамический, правда, такая операция может занять довольно много времени, что в свою очередь сведет на нет все старания злоумышленника.

Данный метод защиты сайта следует реализовать путем написания php-скрипта. Использование такого рода скрипта поможет защитить содержимое сайта от сканирования проводимого при помощи программ-краулеров и, одновременно, поможет существенно замедлить проведение сканирования сайта «вручную». Кроме того, использование подобного скрипта обеспечит прекрасную защищенность абсолютно всех страниц сайта от различных видов хаотичных интенсивных запросов, что в свою очередь даст возможность снизить нагрузку на оборудование веб-сервера.

Разрабатываемый скрипт должен иметь возможность настройки. В частности необходимо заранее предусмотреть возможность изменения части параметров скрипта. Иначе говоря, в скрипте должно быть:

– наличие возможности настройки времени блокировки IP-адреса пользователя;

– наличие возможности задать интервал времени, в который будет проверяться активность пользователя, иначе говоря, время, в течение которого будет вестись учет количества запросов, поступивших от одного определенного пользователя;

– наличие возможности установки количества запросов, которые один пользователь сможет отправить на страницы сайта в течение заданного временного интервала;

– наличие возможности создания списка «всегда разрешенных IP-адресов». IP-адреса, внесенные в данный список, никогда не будут

заблокированы;

– наличие возможности создания списка «всегда запрещенных IP-адресов», т.е. скрипт всегда будет блокировать IP-адреса, которые внесены в данный список.

Преимуществами создания и использования подобного php-скрипта можно считать:

– существенное снижение количества запросов, отправляемых к серверу баз данных;

– существенную экономию входящего и исходящего трафика на веб-сервере;

– наличие удобной и гибкой настройки наиболее важных параметров работы скрипта;

– возможность существенного снижения нагрузки на веб-сервер со стороны пользователей;

– копирование всей информации, размещенной на сайте, будет сильно затруднено или даже невозможно в случае, если страниц на данном ресурсе достаточно много.

Логика работы разрабатываемой системы защиты сайта, создаваемой в виде php-скрипта, приведена на рисунке 3.1.

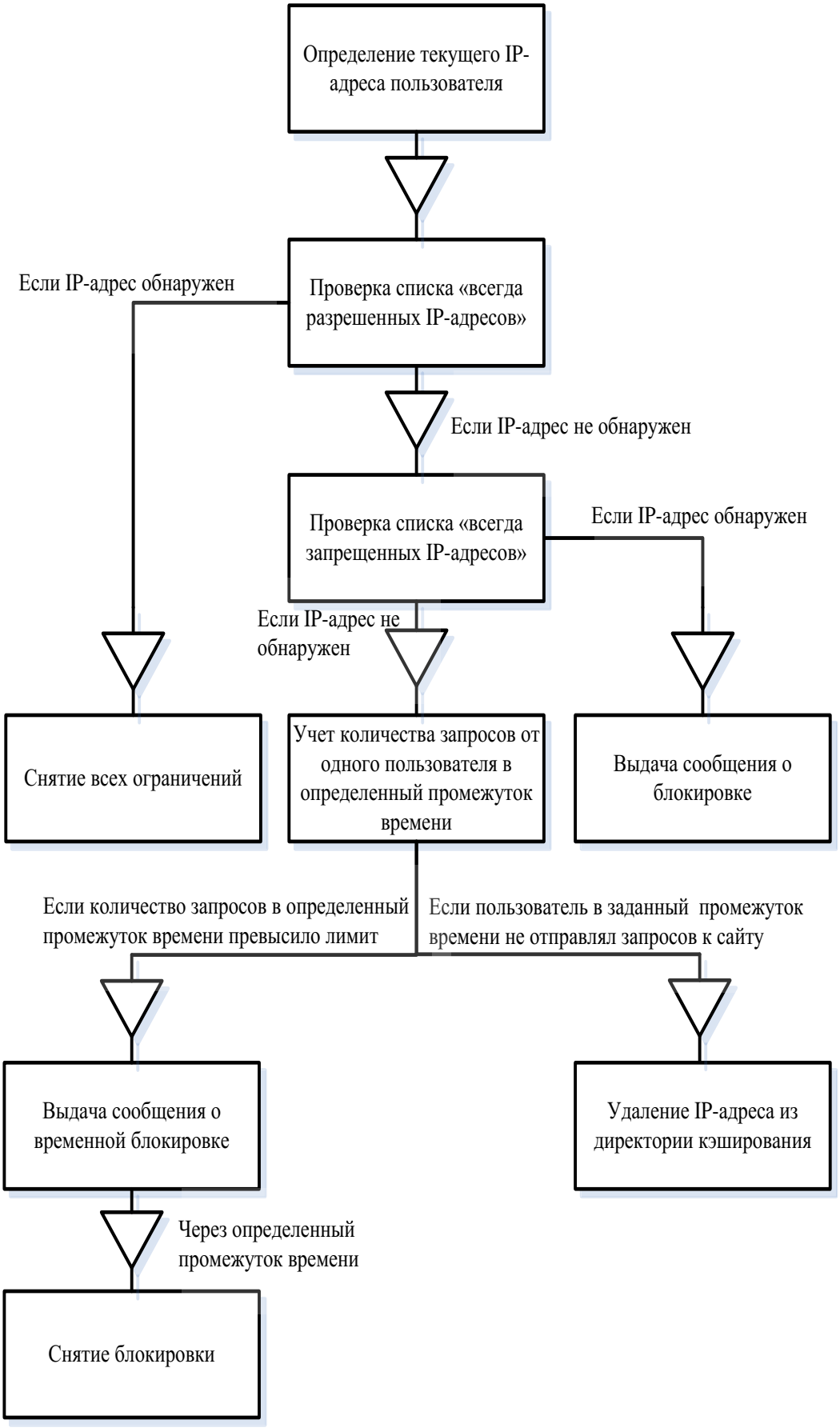

Рисунок 3.1 – Логика работы скрипта

В процессе разработки скрипта важным моментом является выбор способа хранения получаемой информации. В нашем случае, это IP-адреса текущих пользователей. Данная информация может храниться либо в базе данных, либо на жестком диске. Для того чтоб ускорить работу разрабатываемого скрипта, а также сделать его более устойчивым, для хранения IP-адресов текущих пользователей мы будем использовать директории кэширования.

В одну из таких директорий скрипт будет помещать на хранение IP-адреса активных, на данный момент пользователей, эта директория будет носить название active. А в другую директорию мы будем вносить файлы, название которых будет включать IP-адреса временно заблокированных пользователей (директория block).

Кроме IP-адреса, в процессе работы скрипта, также понадобиться информация об активности пользователя. Для этого в название файла, содержащего IP-адрес пользователя, мы также будем вносить точное системное время, когда он проявил «первую», в заданный промежуток времени, активность, т.е. первый раз за определенный (заранее установленный) промежуток времени отправил запрос к странице сайта.

В случае, если пользователь в заданный интервал времени превысил заранее определенное в скрипте количество отправленных запросов к страницам сайта, скрипт удалит файл, содержащий его IP-адрес из директории активных пользователей. После чего запишет новый файл (название которого будет содержать IP-адрес только что заблокированного пользователя) в директорию, содержащую заблокированные IP-адреса.

Таким образом, в директориях кэширования (active и block) будут временно сохраняться файлы с такими названиями как: 127.0.0.1_1302615293, 195.80.91.151_1302615389, 95.30.17.60_1302615457, 77.39.54.104_ 1302615504 и тому подобные.

В имени файла, до нижнего слеша, будет содержаться IP-адрес активного пользователя, а после нижнего слеша в имя файла будет вноситься

точное системное время, в которое он проявил активность (т.е. отправил запрос к страницам ресурса или же был заблокирован).

Отдельно следует отметить тот момент, что для директорий кэширования, т.е. папок active и block, обязательно нужно выставить права доступа владельца файла или 777. Иначе говоря, атрибуты данных папок, указанные на сервере, должны давать скрипту право на запись, право на чтение, а также право на выполнение. Права доступа 777 обязательно должны быть установлены для всех групп пользователей.

К тому же следует предусмотреть отдельное исключение на случай, если администратор сайта этого не сделает. Ведь в подобной ситуации работа скрипта будет невозможна, а значит, защита сайта от сканирования и хаотичных интенсивных запросов окажется, как минимум, несостоятельной. Иначе говоря, в процессе работы скрипта обязательно должно проверяться наличие или же отсутствие возможности произведения чтения, а также записи в какую-либо из директорий кэширования.

Информацию о текущих пользователях (в частности, об их IP-адресах) мы будем брать из суперглобального массива $_SERVER[]. Данный массив создается веб-сервером. В нем содержатся значения разнообразных переменных окружения.

Для работы с IP-адресами пользователей нам понадобиться использовать такие переменные окружения, содержащиеся в суперглобальном массиве $_SERVER[]:

- $_SERVER['HTTP_X_FORWARDED_FOR'];
- $_SERVER['HTTP_CLIENT_IP'];
- $_SERVER['HTTP_X_CLUSTER_CLIENT_IP'];
- $_SERVER['HTTP_PROXY_USER'];
- $_SERVER['REMOTE_ADDR'].

Переменная окружения $_SERVER['HTTP_X_FORWARDED_FOR']

дает нам возможность определить IP-адрес клиента, если он использует для работы прокси-сервер.

Переменная окружения $_SERVER['HTTP_CLIENT_IP'] дает возможность получить IP-адрес клиента, если он не использует прокси-сервер, для работы в Интернете.

Переменная окружения $_SERVER['HTTP_X_CLUSTER_CLIENT_IP'] дает возможность получить IP-адрес клиента, в случае, если на сайте не используется криптографический протокол SSL, обеспечивающий безопасное соединение между сервером и клиентом.

Переменная окружения $_SERVER['HTTP_PROXY_USER'] дает возможность определить IP-адрес клиента, который использует в работе прокси-сервер.

Переменная окружения $_SERVER['REMOTE_ADDR'] дает возможность получить IP-адрес удаленного пользователя. Во время тестирования на локальной машине данный IP-адрес будет равен 127.0.0.1. В то же время в сети данная переменная вернет либо IP-адрес клиента, либо IP-адрес последнего используемого пользователем прокси-сервера (при помощи которого данный клиент попал на веб-сервер).

Таким образом, используя одновременно множество различных переменных окружения из массива $_SERVER[] (все используемые переменные приведены выше), у нас будет возможность определить реальный IP-адрес пользователя, даже в том случае, если он попытается его «замаскировать» при помощи какого-либо прокси-сервера.

Для устойчивой работы скрипта на любой из страниц сайта, вне зависимости от ее уровня вложенности, будем использовать возможность приведения к абсолютному виду путей к директориям кэширования (active и block). Использование такого подхода даст нам возможность запускать скрипт с любой страницы сайта и при этом не опасаться того, что изначально указанные в скрипте относительные пути к директориям кэширования на какой-то из страниц окажутся неверными.

Как уже упоминалось, разрабатываемый скрипт должен иметь возможность настройки. В частности необходимо заранее предусмотреть возможность изменения части параметров скрипта (времени блокировки IP-адреса пользователя, интервала времени, за который будет учитываться количество запросов отправляемых к страницам ресурса, а также количества разрешенных запросов в данный интервал времени).

Данные параметры изначально в скрипте будут задаваться при помощи констант.

В частности, подобными константами в скрипте будут указаны такие параметры:

– время блокировки IP-адреса пользователя указываемое в секундах (const blockSeconds);

– интервал времени, в который будут учитываться запросы от одного пользователя к страницам сайта. Данный интервал также будет указываться в секундах (const intervalSeconds);

– количество запросов к страницам веб-сайта, которые сможет отправить один пользователь за заданный временной промежуток (const intervalTimes).

Отдельно в скрипте следует определить такие массива, содержащие строчные данные:

– массив значений тех IP-адресов, которые внесены в «список всегда разрешенных IP-адресов» (объявление массива public static $alwaysActive = array(‘’));

– массив значений тех IP-адресов, которые внесены в «список всегда запрещенных IP-адресов» (объявление массива public static $alwaysBlock = array(‘’)).

Для правильной работы скрипта, а также для его отладки необходимо предусмотреть наличие в скрипте нескольких флагов. Отметим, что использование подобных флагов также поможет в процессе защиты содержимого сайта от сканирования и различных хаотичных интенсивных

запросов. К таким флагам можно отнести:

– флаг возможности подключения всегда активных пользователей (const isAlwaysActive);

– флаг возможности подключения всегда заблокированных пользователей (const isAlwaysBlock).

Разработанный скрипт содержит один класс TBlockIp. Данный класс включает такие методы:

– checkIp(). Данный метод реализовывает возможность произведения проверки IP-адреса пользователя на его блокировку или же на активность. При этом пропускаются IP-адреса, внесенные в список «всегда разрешенных IP-адресов», а IP-адреса внесенные в список «всегда запрещенных IP-адресов» наоборот блокируются. В случае, если пользовательский IP-адрес не найден в массиве возможных IP-адресов – скрипт создаст идентификатор нового активного IP-адреса;

– _getIp(). Данный метод дает возможность получить текущий IP-адрес пользователя, выбираемый из всех возможных IP-адресов (фильтрация производиться до выявления нужного IP-адреса клиента). Метод возвращает полученный IP-адрес.

В конечном счете разработанный скрипт может считаться эффективным инструментом, который поможет оказывать противодействие процессу сканирования сайта, а также станет мерой пресечения для всех видов хаотичных интенсивных запросов. В разработанном скрипте имеется установка некоторой временной задержки между запросами, исходящими от одного и того же пользователя. При идентификации пользователя основной упор делается на его IP-адрес. Причина этого проста – дело в том, что файлы cookie могут быть легко удалены с компьютера злоумышленника, а значит строить защиту ресурса с их использованием нельзя.

Конечно же, IP-адрес пользователя также может быть изменен, к примеру, при помощи прокси-сервера. Для того чтоб исключить подобную возможность для злоумышленника, в скрипте используются переменные

окружения, которые в свою очередь помогают выявить реальный IP-адрес пользователя. Именно этот IP-адрес, в случае превышения пользователем установленного лимита запросов к страницам сайта в определенный промежуток времени, окажется заблокированным.

Разработанный скрипт выполняет все возложенные на него функции, связанные с защитой сайта от сканирования, а также от хаотичных интенсивных запросов.

3.2 Листинг программы

```
<?php
/**
* Класс проверки и блокировки ip-адреса.
*/
class TBlockIp {
/**
* Время блокировки в секундах.
*/
const blockSeconds = 60;
/**
* Интервал времени запросов страниц.
*/
const intervalSeconds = 15;
/**
* Количество запросов страницы в интервал времени.
*/
const intervalTimes = 3;
/**
* Флаг подключения всегда активных пользователей.
*/
```

```
const isAlwaysActive = true;
/**
* Флаг подключения всегда заблокированных пользователей.
*/
const isAlwaysBlock = true;
/**
* Путь к директории кэширования активных пользователей.
*/
const pathActive = 'active';
/**
* Путь к директории кэширования заблокированных пользователей.
*/
const pathBlock = 'block';
/**
* Флаг абсолютных путей к директориям.
*/
const pathIsAbsolute = false;
/**
* Список всегда активных пользователей.
*/
public static $alwaysActive = array(
'172.16.1.1',
);

/**
* Список всегда заблокированных пользователей.
*/
public static $alwaysBlock = array(
'172.16.1.1',
);
```

```
        /**
        * Метод проверки ip-адреса на активность и блокировку.
        */
        public static function checkIp() {

        // Получение ip-адреса
        $ip_address = self::_getIp();

        // Пропускаем всегда активных пользователей
        if (in_array($ip_address, self::$alwaysActive) && self::isAlwaysActive) {
        return;
        }

        // Блокируем всегда заблокированных пользователей
        if (in_array($ip_address, self::$alwaysBlock) && self::isAlwaysBlock) {
        echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">';
        echo '<html xmlns="http://www.w3.org/1999/xhtml">';
        echo '<head>';
        echo '<title>Вы заблокированы</title>';
        echo '<meta http-equiv="content-type" content="text/html; charset=utf-8"
/>';
        echo '</head>';
        echo '<body>';
        echo '<p style="background:#ccc;border:solid 1px #aaa;margin:30px
auto;padding:20px;text-align:center;width:700px">';
        echo 'Вы заблокированы администрацией ресурса.<br />';
        echo '</p>';
        echo '</body>';
        echo '</html>';
```

```
exit;
}

// Установка путей к директориям
$path_active = self::pathActive;
$path_block = self::pathBlock;

// Приведение путей к директориям к абсолютному виду
if (!self::pathIsAbsolute) {
$path_active = str_replace('\\' , '/', dirname(__FILE__) . '/' . $path_active . '/');
$path_block = str_replace('\\' , '/', dirname(__FILE__) . '/' . $path_block . '/');
}

// Проверка возможности записи в директории
if (!is_writable($path_active)) {
die('Директория кэширования активных пользователей не создана или закрыта для записи.');
}
if (!is_writable($path_block)) {
die('Директория кэширования заблокированных пользователей не создана или закрыта для записи.');
}

// Проверка активных ip-адресов
$is_active = false;
if ($dir = opendir($path_active)) {
while (false !== ($filename = readdir($dir))) {
// Выбирается ip + время активации этого ip
if (preg_match('#^(\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})_(\d+)$#', $filename,
```

```
$matches)) {
        if ($matches[2] >= time() - self::intervalSeconds) {
        if ($matches[1] == $ip_address) {
        $times = intval(trim(file_get_contents($path_active . $filename)));
        if ($times >= self::intervalTimes - 1) {
        touch($path_block . $filename);
        unlink($path_active . $filename);
        } else {
        file_put_contents($path_active . $filename, $times + 1);
        }
        $is_active = true;
        }
        } else {
        unlink($path_active . $filename);
        }
        }
        }
        closedir($dir);
        }

        // Проверка заблокированных ip-адресов
        $is_block = false;
        if ($dir = opendir($path_block)) {
        while (false !== ($filename = readdir($dir))) {
        // Выбирается ip + время блокировки этого ip
        if    (preg_match('#^(\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})_(\d+)$#',    $filename,
$matches)) {
        if ($matches[2] >= time() - self::blockSeconds) {
        if ($matches[1] == $ip_address) {
        $is_block = true;
```

```
$time_block = $matches[2] - (time() - self::blockSeconds) + 1;
}
} else {
unlink($path_block . $filename);
}
}
}
closedir($dir);
}

// ip-адрес заблокирован
if ($is_block) {
header('HTTP/1.0 502 Bad Gateway');
echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">';
echo '<html xmlns="http://www.w3.org/1999/xhtml">';
echo '<head>';
echo '<title>502 Bad Gateway</title>';
echo '<meta http-equiv="content-type" content="text/html; charset=utf-8" />';
echo '</head>';
echo '<body>';
echo '<h1 style="text-align:center">502 Bad Gateway</h1>';
echo '<p style="background:#ccc;border:solid 1px #aaa;margin:30px auto;padding:20px;text-align:center;width:700px">';
echo 'К сожалению, Вы временно заблокированы, из-за частого запроса страниц сайта.<br />';
echo 'Вам придется подождать. Через ' . $time_block . ' секунд(ы) Вы будете автоматически разблокированы.';
echo '</p>';
```

```
echo '</body>';
echo '</html>';
exit;
}

// Создание идентификатора активного ip-адреса
if (!$is_active) {
touch($path_active . $ip_address . '_' . time());
}
}


/**
* Метод получения текущего ip-адреса из переменных сервера.
*/
private static function _getIp() {

// ip-адрес по умолчанию
$ip_address = '127.0.0.1';

// Массив возможных ip-адресов
$addrs = array();

// Сбор данных возможных ip-адресов
if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
// Проверяется массив ip-клиента установленных прозрачными прокси-серверами
foreach (array_reverse(explode(',', $_SERVER['HTTP_X_FORWARDED_FOR'])) as $value) {
$value = trim($value);
```

```
        // Собирается ip-клиента
        if (preg_match('#^\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}$#', $value)) {
        $addrs[] = $value;
        }
        }
        }
        // Собирается ip-клиента
        if (isset($_SERVER['HTTP_CLIENT_IP'])) {
        $addrs[] = $_SERVER['HTTP_CLIENT_IP'];
        }
        // Собирается ip-клиента
        if (isset($_SERVER['HTTP_X_CLUSTER_CLIENT_IP'])) {
        $addrs[] = $_SERVER['HTTP_X_CLUSTER_CLIENT_IP'];
        }
        // Собирается ip-клиента
        if (isset($_SERVER['HTTP_PROXY_USER'])) {
        $addrs[] = $_SERVER['HTTP_PROXY_USER'];
        }
        // Собирается ip-клиента
        if (isset($_SERVER['REMOTE_ADDR'])) {
        $addrs[] = $_SERVER['REMOTE_ADDR'];
        }

        // Фильтрация возможных ip-адресов, для выявление нужного
        foreach ($addrs as $value) {
        // Выбирается ip-клиента
        if (preg_match('#^(\d{1,3}).(\d{1,3}).(\d{1,3}).(\d{1,3})$#', $value,
$matches)) {
        $value = $matches[1] . '.' . $matches[2] . '.' . $matches[3] . '.' . $matches[4];
        if ('...' != $value) {
```

```
$ip_address = $value;
break;
}
}
}

// Возврат полученного ip-адреса
return $ip_address;
}

}

// Проверка текущего ip-адреса
TBlockIp::checkIp();
```

4 Тестирование php-скрипта, внедренного в сайт

Просмотр и последующее тестирование написанного кода являются одним из важнейших этапов разработки php-скриптов. Данный этап разработчики, занимающиеся программированием для Web, нередко упускают. Дело в том, что очень просто можно два или три раза запустить написанный скрипт и отметить, что он работает нормально. В то же время это является распространенной ошибкой. Ведь созданный скрипт необходимо подвергнуть тщательному анализу или же тестированию.

Главным отличием разрабатываемых php-скриптов от прикладных программ является то, что с Web-приложениями будет работать чрезвычайно широкий круг людей. В подобной ситуации не следует рассчитывать на то, что пользователи будут в совершенстве разбираться в использовании различных Web-приложений. Пользователей нельзя снабдить огромным руководством или же кратким справочником. Именно поэтому разрабатываемые Web-приложения обязательно должны быть не только самодокументируемыми, но и самоочевидными. При внедрении разработанного php-скрипта необходимо учесть всевозможные способы его использования, а также протестировать его поведение в различных ситуациях.

Опытному пользователю или программисту нередко сложно понять проблемы, которые возникают у неискушенных пользователей. Одним из способов решения проблемы будет найти тестеров, которые и будут представлять действия типовых пользователей.

В прошлом использовался следующий подход – сначала на рынок выпускалась бета-версия разработанного Web-приложения. После того, как большинство ошибок, предположительно, было исправлено, данное приложение публиковали для небольшого количества пользователей, которые, соответственно, создадут небольшую интенсивность входящего трафика. После проведения такого рода теста разработчики получают массу

различных комбинаций данных и вариантов использования разработанной системы, о которых ее разработчики даже и не догадывались [13, стр. 394].

Созданный php-скрипт должен делать именно то, для чего, собственно говоря, он и был создан. Применительно к разработанному и созданному в данной работе php-скрипту можно сказать так: если пользователь сможет отправить за определенный промежуток времени (по умолчанию 15 секунд) больше заданного количества запросов (по умолчанию – 3 запроса) к страницам сайта и скрипт не заблокирует его IP-адрес, значит, в работу написанного Web-приложения закралась ошибка.

В целом, тестирование представляет собой деятельность, которая направлена на выявление подобного рода несоответствий, существующих между ожидаемым поведением написанного php-скрипта и действительным его поведением. За счет выявления несоответствия в поведении скрипта, а также его некорректного поведения еще во время разработки, у разработчика есть возможность существенно снизить вероятность того, что с такого рода поведением скрипта столкнется и пользователь.

Проводить тестирование программы (в нашем случае, написанного php-скрипта) можно как вручную, так и при помощи специализированных онлайн-сервисов.

4.1 Тестирование вручную

При тестировании написанного php-скрипта вручную, нам необходимо проверить выполняет ли скрипт возложенные на него задачи. В частности, ограничивает ли количество запросов к страницам сайта, исходящим от одного и того же пользователя, в случае, если их частота превышает допустимый предел в определенный промежуток времени.

Для проверки работоспособности скрипта нам, прежде всего, необходимо внедрить его в код сайта. Для этого можно воспользоваться такой языковой конструкцией:

include "block/index.php";

При таком включении скрипт должен быть расположен в папке block. Следует учесть, что в случае, если система управления содержимым сайта использует для страниц создание, так называемых Friendly URL, т.е. веб-адресов, которые удобны для восприятия человеком (в данном случае имеется ввиду создание каких-либо многоуровневых структур, например, /news/sport/2003/10/), или же загружает страницы с адресом отличным от корня папки (например, /news/sport.php), возникнет необходимость правильного указания адреса к скрипту. В подобной ситуации вполне может быть указан абсолютный путь к скрипту.

Само по себе тестирование написанного php-скрипта будет заключаться в отправке запросов к страницам сайта. В процессе тестирования нам нужно проверить: пропускает ли созданный скрипт заданное максимальное количество запросов (по умолчанию – 3 запроса) в определенный промежуток времени (по умолчанию – 15 секунд).

Прежде всего, нам необходимо проверить как скрипт справляется со своей основной функцией, т.е. непосредственной блокировкой интенсивных запросов на страницы сайта, исходящих от одного и того же пользователя.

Проверка количества запросов, которые могут быть отправлены к страницам сайта, показала, что на протяжении отведенных пятнадцати секунд отправить к страницам сайта более трех запросов нет возможности. В частности, если пользователь отправляет больше трех запросов, то получает сообщение о временной блокировке, приведенное на рисунке 4.1.

Рисунок 4.1 – Сообщение о временной блокировке

Таким образом, тестирование показало, что скрипт работает именно так, как предполагалось: блокирует пользователя на определенное время (по умолчанию – на одну минуту) в случае, если в определенный интервал времени количество запросов от одного пользователя превышает заданный показатель (по умолчанию – 3 запроса).

Следующим этапом тестирования будет проверка, работоспособности отдельных функций написанного php-скрипта. В частности, нужно проверить, как будет работать скрипт если IP-адрес тестирующего пользователя будет добавлен в список «всегда разрешенных IP-адресов». Также нужно проверить, как будет вести себя скрипт, если IP-адрес тестирующего пользователя будет добавлен в список «всегда запрещенных IP-адресов».

После добавления IP-адреса тестирующего пользователя в список «всегда разрешенных IP-адресов» ограничение на количество отправляемых к страницам сайта запросов оказалось снятым, и переходить по ссылкам (или обновлять страницу) можно было неограниченное количество раз. Иначе говоря, функция, которую должен выполнять список «всегда разрешенных

IP-адресов» правильно работает.

После добавления IP-адреса тестирующего пользователя в список «всегда запрещенных IP-адресов» отправить хотя бы один запрос к странице сайта не удалось. Вместо содержимого веб-ресурса в браузере появляется сообщение о блокировке, приведенное на рисунке 4.2.

Рисунок 4.2 – Сообщение о блокировке

Другими словами, функция, которую должен выполнять список «всегда запрещенных IP-адресов» работает правильно. Данная функция полностью блокирует пользователя, IP-адрес которого внесен в список «всегда запрещенных IP-адресов».

Таким образом, функции разграничения доступа, которые реализованы при помощи наличия списка «всегда разрешенных IP-адресов», а также наличия списка «всегда запрещенных IP-адресов» работают верно и скрипт ведет себя именно так, как изначально предполагалось. Иначе говоря, скрипт предоставляет неограниченный доступ к страницам сайта для тех пользователей, IP-адреса которых внесены в список «всегда разрешенных IP-адресов» и полностью запрещает доступ тем пользователям, чьи IP-адреса

обнаруживаются в списке «всегда запрещенных IP-адресов».

Тестирование написанного php-скрипта вручную показало, что все функции скрипта, а значит и сам скрипт, работают правильно, т.е. именно так как изначально задумано.

4.2 Тестирование при помощи онлайн-сервисов

Тестирование написанного php-скрипта, при помощи онлайн-сервисов, даст возможность узнать, точное количество трафика необходимого для загрузки страниц сайта, а также поможет точно установить «пропускную способность» написанного php-скрипта для запросов, которые будут отправляться различными автоматическими сервисами или же программами. Последний параметр очень важен, ведь одной из задач скрипта является защита сайта от сканирования при помощи программ-краулеров.

Такого рода тестирование не только поможет сделать вывод об необходимом количестве трафика и процессорного времени, но и о том, доступна ли будет информация представленная на сайте для программных средств, запрашивающих страницы в автоматическом режиме.

Для начала оценим среднее время загрузки страниц сайта, при разных скоростях соединения. Также оценим и размер загружаемых пользователем страниц. Для проведения такого рода тестирования воспользуемся онлайн-сервисом, расположенным по адресу http://analyze.websiteoptimization.com.

Размеры загружаемых страниц, определенные сервисом, приведены на рисунке 4.3.

Рисунок 4.3 – Размер загружаемых страниц

Из данной таблицы видно что, размер страницы составляет 3824 байта, она загрузиться за 0,96 секунды (при скорости соединения 56 К) или за 0,22 секунды (при скорости соединения пользователя 1,44 Mbps).

Скорость загрузки страниц при разной скорости соединения пользователя приведена на рисунке 4.4.

Рисунок 4.4 – Среднее время загрузки страниц

Как видно из приведенной на рисунке информации, даже при самой низкой скорости соединения (14,4 К) страница сайта грузится довольно быстро – за 13,60 секунд. В то же время, если скорость соединения пользователя довольно высока (в частности, 1,44 Mbps), то страница будет загружена буквально моментально (за 0,89 секунды).

Отметим, что низкая скорость загрузки на данном ресурсе корректируется на коэффициент потери пакетов (этот коэффициент равен 0,7). Также учитывается и время задержки равное в среднем 0,2 секунды на загрузку одной страницы. Именно эти параметры и объясняют, почему существенный рост скорости соединения не приводит к столь же существенному сокращению времени загрузки страницы.

Отметим, что данный сервис кроме оценки основных параметров загрузки страниц сайта также предоставляет общие результаты анализа сайта и, конечно же, рекомендации направленные на улучшение всех параметров загрузки страниц сайта.

Результаты анализа и рекомендации сервиса приведены на рисунке 4.5.

Рисунок 4.5 – Результаты анализа и рекомендации

Как видно из рисунка, все показатели тестируемого сайта по всем анализируемым параметрам находятся в пределах нормы. В частности, сервис констатирует, что количество страниц сайта, объектов на них, а также размеры этих страниц и объектов находятся в допустимых пределах. Исходя из полученных данных оптимизация тестируемому сайту не требуется, ведь сайт соответствует нормам по всем анализируемым параметрам.

Иначе говоря, объемы загружаемой с сайта информации (или страниц сайта) довольно малы и позволяют экономить ресурсы как веб-сервера, так и пользователя. Т.е. разработанный скрипт не нагружает сайт и, фактически, его присутствие не заметно. Что подчеркивает проведенное тестирование.

Проверку скорости загрузки страниц сайта, а также их доступности для других скриптов можно выполнить при помощи такого онлайн-сервиса.

Результат теста, сделанного при помощи данного онлайн-сервиса, приведен на рисунке 4.6.

Рисунок 4.6 – Проверка скорости загрузки сайта и его доступности

Как видно, из полученных данных при средней скорости загрузки 9,34 Кб/сек, время загрузки страницы составляет 0,4 секунды. В то же время при средней скорости 7,05 Кб/сек, время загрузки страницы составляет уже 0,53 секунды. В то же время, тестирование показало, что разработанный в данной работе скрипт блокирует больше трех запросов к страницам сайта, исходящих от одного IP-адреса и тем самым снижает излишнюю нагрузку и весьма усложняет попытки или вовсе делает невозможным подбор пароля к сайту.

4.3 Результаты тестирования php-скрипта, внедренного в сайт

Проведенные тесты показали, что разработанный php-скрипт отлично справляется с изначально возложенными на него функциями по защите сайта от сканирования, а также от хаотичных интенсивных запросов и попыток брутфорса, исходящих от одного пользователя.

В частности, тестирование сайта вручную показало, что скрипт блокирует запросы к страницам сайта, исходящие с одного IP-адреса, в случае, если количество этих запросов в заданный интервал времени превышает заранее отведенный лимит.

Кроме того, данное тестирование показало, что все функции скрипта работают отлично, в частности функции по работе со списками «всегда разрешенных IP-адресов» и «всегда запрещенных IP-адресов».

Иначе говоря, разработанный скрипт предоставляет неограниченный доступ ко всем страницам сайта для тех пользователей, IP-адреса которых внесены в список «всегда разрешенных IP-адресов» и полностью запрещает доступ тем пользователям, чьи IP-адреса обнаруживаются в списке «всегда запрещенных IP-адресов».

Тестирование сайта, в который внедрен разработанный php-скрипт, показало, что все показатели тестируемого сайта по всем анализируемым параметрам находятся в пределах нормы. В частности, сервис констатирует, что количество страниц сайта, объектов на них, а также размеры этих страниц и объектов находятся в допустимых пределах. Исходя из полученных данных оптимизация тестируемому сайту не требуется, ведь сайт соответствует нормам по всем анализируемым параметрам.

Иначе говоря, объемы загружаемой с сайта информации (или страниц сайта) довольно малы и позволяют экономить ресурсы как веб-сервера, так и пользователя.

Кроме того, тестирование сайта при помощи онлайн-сервиса, расположенного по адресу http://www.pr-cy.ru, показало, что скорость

загрузки страниц тестируемого сайта довольно высока. Также, скрипт заблокировал проведение тестирования сайта, когда количество запросов отправленных с IP-адреса данного онлайн-сервиса превысило допустимый в разработанном скрипте предел (максимальное количество запросов с одного IP-адреса, установленное в скрипте, равно трем запросам за 15 секунд).

Другими словами, защита сайта от сканирования и хаотичных интенсивных запросов работает именно так, как и предусматривалось изначально. Другими словами, если количество запросов, исходящих с одного IP-адреса (в случае, если IP-адрес пользователя не находится в списке «всегда разрешенных IP-адресов» или же в списке «всегда запрещенных IP-адресов»), превышает допустимый лимит в определенный интервал времени, то IP-адрес пользователя будет временно заблокирован вне зависимости от того, отправлялись запросы при помощи браузера, программы и использовался ли при отправке запросов прокси-сервер (для маскировки реального IP-адреса пользователя).

5 Экономическая часть

На сегодняшний день достаточно отчетливо ясно, что для конкурентоспособности сайтов необходимо, помимо различных экономических расчетов и обоснований, планомерных действий со стороны управленческого аппарата, тщательного стратегического планирования, осуществлять и непосредственную защиту данных ресурсов. Поэтому защите сайта на сегодняшний день в особой рекламе не нуждается. Конечно же, средств защиты сайта существует довольно много, но не все они могут быть использованы по причине их сложности или же дороговизне. В целом, решение о выбираемых конкретных способах и, соответственно, возможностях защиты сайта принимается его владельцем или же специалистом по защите информации. В любой случае, человек, принимающий решение относительно защиты сайта должен уметь обосновать технические возможности каждого из предлагаемых вариантов. Однако не следует забывать, что любой способ защиты несет определенные затраты, которые отражаются в соответствующих отчетных документах и являются составляющей частью себестоимости предоставляемых услуг.

5.1 Разработка сетевой модели решения поставленной задачи, составление календарного плана-графика выполнения работ

Работы необходимые для решения поставленной задачи:

1-2) Постановка технического задания.

2-3) Установка и настройка рабочего места и программного обеспечения.

3-4) Анализ методов решения задачи.

4-5) Выбор наиболее эффективного метода защиты сайта от сканирования и хаотичных интенсивных запросов.

5-6) Уточнение и доработка выбранного метода защиты сайта от

сканирования и хаотичных интенсивных запросов.

6-7) Написание php-скрипта.

7-8) Отладка написанного php-скрипта.

8-9) Составление технической документации, распечатка.

События:

1) Поступил заказ на разработку метода защиты сайта от сканирования и хаотичных интенсивных запросов.

2) Завершен анализ цели проектирования.

3) Закончена установка и настройка рабочего места и программного обеспечения.

4) Закончен обзор методов решения задачи.

5) Закончен выбор наиболее эффективного метода защиты сайта от сканирования и хаотичных интенсивных запросов.

6) Закончены уточнения и доработки выбранного метода защиты сайта от сканирования и хаотичных интенсивных запросов.

7) Закончено написание php-скрипта.

8) Закончена отладка php-скрипта.

9) Составлена техническая документация.

Календарный план-график выполняемых работ приведен в таблице 5.1.

Таблица 5.1 – Календарный план-график выполняемых работ

Сроки выполнения работ	Шифр работы	Наименование работы	Tmin (дни)	Тнв (дни)	Tmax (дни)	Тож (дни)	Категория исполнителя	Начисленная ЗП, руб
04.03.10-09.03.10	1-2	Постановка технического задания	4	5	6	5	Системный аналитик	10000
10.03.10-13.03.10	2-3	Установка и настройка рабочего места и программного обеспечения	2	4	6	4	Системный администратор	6000
14.03.10-24.03.10	3-4	Анализ методов решения задачи	8	11	14	11	Системный аналитик	22000

25.03.10-31.03.10	4-5	Выбор наиболее эффективного метода защиты сайта от сканирования и хаотичных интенсивных запросов	5	7	9	7	Программист	10500
1.04.10-10.04.10	5-6	Уточнение и доработка выбранного метода защиты сайта от сканирования и хаотичных интенсивных запросов	6	10	14	10	Программист	15000
11.04.10-27.04.10	6-7	Написание php-скрипта	15	17	19	17	Программист	25500
28.04.10-08.05.10	7-8	Отладка написанного php-скрипта	6	10	14	10	Программист	15000
09.05.10-15.05.10	8-9	Составление технической документации, распечатка	5	7	9	7	Консультант-аналитик в области ИБ	10500

На рисунке 5.1 построим сетевой график.

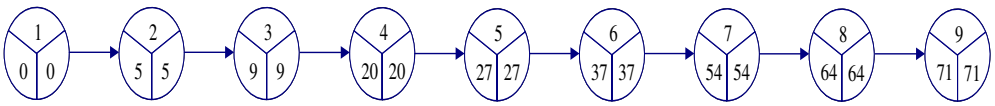

Рисунок 5.1 – Сетевой график

Рассчитаем критический путь (t(кр)):

L12345678 = 5 + 4 + 11 + 7 + 10 + 17 + 10 + 7= 71 день

5.2 Расчет затрат на создание скрипта для защиты сайта от сканирования и хаотичных интенсивных запросов

В связи с тем, что все работы, производимые в ходе разработок по своему составу и степени законченности можно отнести к ПБУ «Учет

расходов на научно-исследовательские, опытно-конструкторские и технологические работы», рассмотрим состав затрат согласно этому положению.

Согласно данному положению к расходам по научно-исследовательским, опытно-конструкторским и технологическим работам относятся все фактические расходы, связанные с выполнением указанных работ. В состав расходов при выполнении научно-исследовательских, опытно-конструкторских и технологических работ включаются:

– стоимость материально-производственных запасов и услуг сторонних организаций и лиц, используемых при выполнении указанных работ;

– затраты на заработную плату и другие выплаты работникам, непосредственно занятым при выполнении указанных работ по трудовому договору;

– страховые взносы в Пенсионный фонд РФ, Фонд социального страхования РФ, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования, стоимость спецоборудования и специальной оснастки, предназначенных для использования в качестве объектов испытаний и исследований;

– амортизация объектов основных средств и нематериальных активов, используемых при выполнении указанных работ;

– затраты на содержание и эксплуатацию научно-исследовательского оборудования, установок и сооружений, других объектов основных средств и иного имущества;

– общехозяйственные расходы, в случае если они непосредственно связаны с выполнением данных работ;

– прочие расходы, непосредственно связанные с выполнением научно-исследовательских, опытно-конструкторских и технологических работ, включая расходы по проведению испытаний.

1) Стоимость материально-производственных запасов и услуг сторонних организаций и лиц, используемых при выполнении указанных работ.

К данной статье принадлежат расходы на доступ в сеть Интернет с помощью сторонних организаций, а именно, Интернет-провайдера «Net-by-net». Первоначальная стоимость подключения составляет 2000 рублей, а ежемесячный взнос за услуги провайдера – 500 рублей.

Сумма расходов по данной статье на срок 2 месяца составит:

$2000 + 2 \cdot 500 = 3000$ рублей.

2) Затраты на заработную плату и другие выплаты работникам, непосредственно занятым при выполнении указанных работ по трудовому договору.

Для расчёта затрат по данной статье нам необходимо воспользоваться сетевым графиком приведенным на рисунке 5.1.

В таблице 5.2 приведены сотрудники, которые задействованы в разработке php-скрипта для защиты сайта от сканирования и интенсивных хаотичных запросов.

Таблица 5.2 – Сотрудники предприятия

Исполнитель	Количество дней, дн.	Зарплата в месяц, руб.	Зарплата, руб.
Системный аналитик	16	40000	32000
Системный администратор	4	30000	6000
Программист	44	30000	66000
Консультант-аналитик в области ИБ	7	30000	10500
Итого			114500

Таким образом, затраты на заработную плату сотрудникам составят 114500 рублей.

3) Страховые взносы в Пенсионный фонд РФ, Фонд социального страхования РФ, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования.

В 2010 согласно закону «О страховых взносах в Пенсионный фонд РФ, Фонд социального страхования РФ, Федеральный фонд обязательного медицинского страхования и территориальные фонды обязательного медицинского страхования» ставка страховых взносов устанавливается в размере 26%, из них 20% идут Пенсионный фонд, 2,9% – в Фонд соцстрахования, 3,1% – в фонды обязательного медицинского страхования.

Расчет страховых взносов приведен в таблице 5.3.

Таблица 5.3 – Расчет страховых взносов

Исполнитель	Зарплата, руб.	Процентная ставка, %	Страховой взнос, руб.
Системный аналитик	32000	26	8320
Системный администратор	6000	26	1560
Программист	66000	26	17160
Консультант-аналитик в области ИБ	10500	26	2730
Итого			29770

Таким образом, общая сумма страховых взносов составит 29770 рублей.

4) Стоимость специального оборудования и специальной оснастки.

В таблицу 5.4 внесем список оборудования необходимого для выполнения задач по разработке методов защиты сайта от сканирования, а также от хаотичных интенсивных запросов.

Таблица 5.4 – Необходимое оборудование

Оборудование	Количество	Стоимость, руб.
Компьютер	2	16000
Принтер	1	8000
Итого		40000

Стоимость оборудования составит 40000 рублей.

В качестве специальной оснастки в данном случае будет выступать программное обеспечение.

Все установленное программное обеспечение является полностью бесплатным, или свободно предоставленным разработчиками на ограниченный срок.

5) Амортизация объектов основных средств и нематериальных активов, используемых при выполнении указанных работ.

Амортизация – в широком смысле – бухгалтерская и налоговая концепции, используемые для оценки потери величины стоимости активов с течением времени.

Амортизация объектов основных средств:

Принтер. Первоначальная стоимость: 8000 руб.

Полный срок его полезного использования (4 года) – это 48 месяцев. Учитывая относительно медленный темп устаревания, но достаточно интенсивную эксплуатацию, установлен срок эксплуатации в 4 года и выбран линейный метод начисления амортизации.

Ежемесячную норму амортизации определяем так (п. 4 ст. 259 НК РФ):

$$K = \left[\frac{1}{48}\right] \cdot 100\% = 2{,}08\ \%.$$

$$A_{мес} = 8000 \cdot 2{,}08\% = 166{,}4\ руб.$$

Компьютер. Первоначальная стоимость: 16000 руб.

Полный срок его полезного использования (3 года) – это 36 месяцев.

$$K = \left[\frac{1}{36}\right] \cdot 100\% = 2{,}78\ \%.$$

$$A_{мес} = 16000 \cdot 2{,}78\% = 444{,}8\ руб.$$

Так как будут использоваться 2 компьютера, ежемесячные амортизационные отчисления утроятся, т.е.:

$$A_{мес\_полн} = 444{,}8 \cdot 2 = 889{,}6 \ руб.$$

Полные амортизационные отчисления в период выполнения работ (2 месяца) составят:

$$A_{полн} = 2 \cdot (166{,}4 + 889{,}6) = 2112 \ руб.$$

6) Затраты на содержание и эксплуатацию научно-исследовательского оборудования, установок и сооружений, других объектов основных средств и иного имущества.

Затрат по данной категории не требуется.

7) Общехозяйственные расходы, в случае если они непосредственно связаны с выполнением данных работ.

К данным расходам относятся расходы на аренду помещений, электроэнергию, коммунальные услуги.

При ставке Мосэнергосбыт 3,45 руб. за 1 кВт/час и ежемесячном потреблении электроэнергии порядка 180 кВт/час, затраты за 2 месяца составят:

$$\frac{71}{22} \cdot 3{,}45 \cdot 180 = 2004 \ руб.$$

Аренда помещения площадью 20 м2 сроком на 2 месяца при стоимости 11000 рублей за квадратный метр в год включая коммунальные услуги (кроме электричества) составит:

$$\frac{11000 \cdot 20}{12} \cdot \frac{71}{22} = 59166 \ руб.$$

Общие расходы составят:

2004 + 59166 = 61170 руб.

8) Прочие расходы, непосредственно связанные с выполнением научно-исследовательских, опытно-конструкторских и технологических работ, включая расходы по проведению испытаний.

К данным расходам отнесем расходы на покупку носителей

информации:

- бумага для принтера формата А4 (500 листов) – 180 руб.
- канцелярские принадлежности – 320 руб.

Прочие расходы составят – 500 руб.

9) Итог.

В таблице 5.5 подсчитаем общие затраты на защиту сайта от сканирования и хаотичных интенсивных запросов.

Таблица 5.5 – Общие затраты

Вид затрат	Стоимость (руб.)
1. Стоимость материально-производственных запасов и услуг сторонних организаций и лиц	3000
2. Затраты на заработную плату работникам	114500
3. Отчисления на социальные нужды	29770
4. Стоимость спец. оборудования и спец. оснастки	40000
5. Амортизация объектов основных средств и нематериальных активов	2112
6. Затраты на содержание и эксплуатацию научно-исследовательского оборудования	0
7. Общехозяйственные затраты	61170
8. Прочие расходы	500
Итого:	251052

Таким образом, общие затраты на защиту сайта от сканирования и хаотичных интенсивных запросов составят 251052 руб.

Списание расходов по научно-исследовательским, опытно-конструкторским и технологическим работам проводится один раз в год 1-го июня.

6 Охрана труда и окружающей среды

В процессе работы программист воздействует на орудия труда (компьютер, оргтехнику) в определенных условиях среды. В результате такого взаимодействия он может подвергаться воздействию различных факторов: механических, химических, термических и т.д.

Совокупность факторов, воздействующих на программиста в процессе труда, формируют условия труда (условия производственной среды).

Условия труда – это совокупность факторов производственной среды, оказывающих влияние на программиста, его работоспособность и здоровье.

Человек может переносить те или иные воздействия пока они не превысят определенного уровня и/или продолжительности.

Предельно допустимая величина (ПДВ) фактора – это такая минимальная (максимальная) величина при которой человек может работать ежедневно нормальную рабочую смену весь трудовой период до выхода на пенсию, при этом у него не возникает отклонение здоровья вызванного этим фактором. Значение ПДВ устанавливается государственным документами:

1) ГОСТ
2) Нормы
3) СН
4) СниП
5) СанПиН

В зависимости от количественных значений факторов условия труда подразделяют на четыре класса: оптимальные (I), допустимые (II), вредные (III) и опасные или экстремальные (IV). Для первых двух классов факторы не превышают ПДВ, т.е. условия труда являются безопасными. Для III класса факторы превышают ПДВ, возможно профессиональное заболевание. В случае IV класса риск профессионального заболевания очень высок.

Оценка класса условий труда производится на основе инструментальных измерений факторов производственной среды и

сравнении их с ПДВ. Превышение допустимых значений является нарушением правил охраны труда и требует принятия мер по их снижению, либо доплат за риск.

6.1 Воздушная среда

Воздушную среду с точки зрения охраны труда и безопасности жизнедеятельности можно разделить на наружную, внутреннюю (производственную) и внутреннюю (жилую).

Наружная воздушная среда характеризуется метеорологическими факторами и вредными факторами загрязнения. К метеорологическим факторам относятся: температура, влажность, скорость движения воздуха и атмосферное давление.

Воздушную среду внутренней производственной среды иначе называют микроклиматом. Микроклимат оказывает влияние на теплообмен человека с окружающей его средой.

Микроклимат подразделяется на следующие группы параметров микроклимата: оптимальные, допустимые и недопустимые.

Оптимальные параметры - это условия, которые обеспечивают человеку наивысшую работоспособность, то есть человек испытывает состояние теплового комфорта, а система терморегуляции организма работает с наименьшей нагрузкой.

Допустимые параметры – условия, при которых работоспособность в течение дня может снижаться, может ощущаться дискомфорт, однако после отдыха работоспособность возвращается.

Недопустимые параметры – условия, вызывающие дискомфорт и заболевания.

При резком изменении микроклимата человеку нужно время адаптации. При длительном нахождении в условиях повышенной/пониженной температуры необходима акклиматизация.

Рассмотрим составляющие микроклимата производственной среды: температуру, влажность и скорость движения воздуха.

Оптимальные значения температуры: t = +17°...+23°С.

Допустимые значения температуры: от +12°...+18°С до +24°...+28°С. При повышении температуры выше, чем +30°С может привести к перегреву и тепловому удару. Если же температура снижается ниже +12°С следует переохлаждение организма.

Оптимальные значения относительной влажности: 40%...60%.

Допустимые значения относительной влажности: 60%...75%.

При превышении относительной влажностью значения 80% затрудняется испарение влаги с поверхности тела, что приводит к перегреву и тепловому удару. При снижении значения относительной влажности ниже 20% испарение будет происходить интенсивно, что приводит к переохлаждению организма.

Скорость движения воздуха ощущается человеком в том случае, если она выше 0,1 м/с.

Эффективный отвод тепла с поверхности тела человека обеспечиваемся при: оптимальных значениях: 0,2...0,3 м/с;

допустимых значениях: 0,3...0,5 м/с.

При скорости движения воздуха свыше 0,7 м/с происходит чрезмерный отвод тепла с поверхности тела человека, что приводит к охлаждению и переохлаждению. Для оценки состояния человека используются понятия: эффективная температура, учитывающая влияние на человека температуры и относительной влажности, и эффективно-эквивалентная температура, учитывающая влияние температуры, относительной влажности и скорости движения воздуха.

Нормирование параметров микроклимата происходит в соответствии со следующими разделами, входящими в состав ГОСТ 12.1.005.88*: Система стандартов безопасности труда (ССБТ); Система стандартов охраны природы (ССОП); Безопасность в чрезвычайных ситуациях (БЧС).

ГОСТ 12.1.005.88* устанавливает оптимальные и допустимые параметры микроклимата, принципы нормирования, значения параметров микроклимата с учетом времени года и категории работ. ГОСТ 12.1.005.88* различает два периода года: теплый (среднесуточная температура выше +10°С) и холодный (среднесуточная температура ниже +10°С).

Следует сказать несколько слов о чистоте воздушной среды и вредных веществах, образующихся в воздушной среде в результате хозяйственной деятельности человека, таких как газы, пыль и т. п.

Вредными веществами называют вещества, вызывающие несчастные случаи и профессиональные заболевания при контакте с человеком (такие как свинец, ртуть, углекислый газ, продукты горения (дым), хлор, аммиак, ацетон, лаки, краски и тому подобные вещества и газы).

Пыль имеет особенность оседать на поверхности и, под воздействием воздушных потоков, снова подниматься в воздух.

Чистый воздух имеет в своем составе: азот (78,08%), кислород (20,95%), углекислый газ (0,03%) и прочие газы (в том числе инертные).

Качественное значение для человека имеет кислород. При содержании в воздушной среде кислорода меньше 17% снижается работоспособность, если содержание меньше 12% возникает угроза для жизни и возможна потеря сознания. Содержание кислорода в воздухе менее 6% ведет к прекращению дыхания и смерти.

Реальный состав воздуха редко соответствует приведенному составу чистого воздуха.

Вредные вещества классифицируют по следующим признакам: агрегатному состоянию, характеру воздействия на человека, степени воздействия, направленности воздействия.

Контроль содержания вредных веществ в воздушной среде во внутренней производственной среде для веществ I класса вредности производится постоянно, для II, III, IV классов периодически (периодичность зависит от характера и объема вредных выделений). Для контроля

применяют два основных метода: экспрессивный (простой и быстрый, но неточный) и лабораторный (высокоточный, но медленный).

Важным фактором в работе программиста является вентиляция. Вентиляция - это система мероприятий и устройств, обеспечивающих необходимый воздухообмен в помещении. В зависимости от способа перемещения воздуха вентиляция подразделяется на: естественную, механическую и комбинированную Интенсивность вентиляции характеризуется параметром, называющимся кратностью воздухообмена. Кратность воздухообмена есть отношение количества воздухообмена (м /час) к объему помещения (м^3).

Кратность воздухообмена показывает сколько раз в час происходит полная смена объемов воздуха в помещении.

Величина воздухообмена определяется в зависимости от характера вредных выделений, выделений избыточного тепла или избыточной влаги.

В случае вредных выделений воздухообмен вычисляется следующим образом:

$$L_{\text{вв}} = \frac{K_B}{\text{ПДК - К}_{\text{п}}},$$

где Кв – концентрация вредных веществ в рабочей зоне (м /час);

Кп – количество вредных веществ (мг/м);

Кп < 0,3 ПДК.

ПДК – предельно допустимая концентрация, устанавливается нормативными документами.

Рабочей зоной называется пространство помещений до 2 метров высотой над уровнем пола, где расположено рабочее место программиста.

Для случая избыточного тепла величина воздухообмена вычисляется следующим образом

$$L_Q = \frac{3600 * Q_{\text{изб}}}{C_p * \rho * \Delta t},$$

где Qизб – избыточная теплота (Дж/сек);

Ср – удельная теплоемкость воздуха (Дж/кг • град);

р – плотность воздуха;

Δ t – разность температур удаляемого и подаваемого воздуха.

Для случая избыточной влаги:

$$L_\delta = \frac{10^3 * \text{G}}{\text{A}_\text{у} - A_\text{П}},$$

где G – количество избыточной влаги, выделяемой в воздушную среду (г/час);

Ау – абсолютная влажность удаляемого воздуха;

Ап – абсолютная влажность подаваемого воздуха.

Если в воздушную среду выделяется несколько вредных веществ однонаправленного действия, то величина воздухообмена определяется как сумма воздухообменов, необходимых для удаления этих веществ.

Если вещества имеют разнонаправленный характер, то общий воздухообмен будет считаться как максимальный необходимый для удаления максимально концентрированного вещества.

При естественной вентиляции воздухообмен обеспечиваемся за счет разности температур и гравитационного обмена наружного воздуха с внутренним. Преимущество естественной вентиляции в том, что она не требует технических устройств и ухищрений.

Естественная вентиляция подразделяется на неорганизованную (проветривание) и организованную (аэрация).

Механическая вентиляция обеспечивает подачу воздуха в помещения с помощью системы воздуховодов и вентиляторов. Существуют следующие

виды механической вентиляции: приточная, вытяжная и комбинированная.

К системам вентиляции предъявляются следующие требования:

1. Количество удаляемого воздуха должно соответствовать количеству подаваемого;
2. Если в одном из смежных помещений выделяются вредные вещества, то количество удаляемого воздуха должно быть больше подаваемого;
3. Если в производственном помещении на одного работающего приходится объем воздуха менее 20 м , то количество подаваемого воздуха должно быть не менее 30 м /час на одно рабочее место. Если объем более 20 м , то количество подаваемого воздуха на одно рабочее место должно быть не менее 20 м3/час на человека;
4. Система вентиляции не должна превышать уровня шума допустимого нормативами значения;
5. Система вентиляции должна обладать следующими свойствами: надежностью, безопасностью и т.д.

6.2 Шум, инфразвук, ультразвук и вибрация

Шум – это сочетание звуков различной частоты и интенсивности, иначе звук не несущий информацию. С физической точки зрения - механическое колебание в пространстве (акустическом поле).

Шум характеризуется звуковым давлением, интенсивностью, частотой и акустической мощностью. Интенсивность шума - это поток энергии, проходящий через единицу площади в единицу времени перпендикулярно направлению распространения. Наименьшее значение интенсивности и давления звука называют порогом слышимости. Напротив, самые большие значения интенсивности и звукового давления (без повреждений слуховых органов) принято называть болевым порогом.

Так как слух реагирует на среднеквадратическое изменение давления звука, то чаще используют понятие уровня звукового давления.

Допустимые значения уровня звукового давления лежат на интервале 0...80 дБ по логарифмической шкале $L = 20 \cdot \lg (P/P0)$, где P0 - это пороговое значение звукового давления, P – наблюдаемое значение звукового давления.

Диапазон звуковых (слышимых человеку) частот лежит на интервале 20...20000 Гц. Выше и ниже этого диапазона частоты ультразвука и инфразвука соответственно.

Так как восприятие звука субъективно, вводится понятие громкости. Громкость измеряется в фонах. Фон - это разность уровня звуковой интенсивности в 10 дБ эталонного звука частотой 1000 Гц. Измерение громкости проводится при помощи аудиометра.

Шум классифицируется по источнику возникновения (механический, аэродинамический, гидродинамический и электромагнитный), по частоте, по спектру (широкополосный, тональный), по временным характеристикам (постоянный, непостоянный).

При проведении акустических расчетов используют понятие октавы. Октава - это удвоение частоты. Таким образом, весь диапазон звуковых частот разбивают на ряд октав (стандартный ряд: 31,5 Гц; 63 Гц; 125 Гц; 250 Гц; 500 Гц; 1 кГц и т.д.). Например, шум считается широкополосным, если его спектр шире одной октавы.

Степень воздействия шума на человека зависит от уровня, характера шума, продолжительности воздействия и индивидуальных особенностей человека. В определенных условиях шум может влиять на все органы человека. Неблагоприятное воздействие начинается при 50...60 дБ, особенно при умственной работе. При 70...80 дБ - начинается физиологическое воздействие (на сердечно сосудистую систему и т.д.). При 85...90 дБ -влияние на органы слуха вызывает снижение восприимчивости, а затем и «тугоухость». Влияние шума высокого уровня на нервную систему вызывает близорукость и дальтонизм. Более высокие уровни шума воздействуют на кору головного мозга, вызывают психические расстройства. Далее следует потеря слуха и вероятность возникновения несчастных случаев.

Нормирование шума состоит из гигиенического нормирования и технического нормирования. В производственных помещениях шум регулируется согласно ГОСТ 12.1.003.89*. На территории жилой застройки и в жилых помещениях ГОСТ 12.1.036.81*.

Принципиально существует два метода нормирования шума: нормирование по предельному спектру и нормирование по общему уровню.

В первом случае при нормировании ориентируются на предельный спектр. Предельным спектром (ПС) называется совокупность нормативных значений звукового давления в 8-октавной системе. Например: ПС-80, ПС-45 и т.д.

Во втором случае для ориентировочной оценки постоянных и непостоянных шумов применяют так называемый общий уровень, измеряемый в дБА, где А - это признак субъективности. Уровень в дБА (LA) привязан к предельному спектру следующим образом: LA = ПС + 5.

Далее приведена классификация методов и средств и защиты от шума:

1) Коллективные методы и средства, к ним относятся:

– организационные;

– строительно-архитектурные;

– технические, к ним относятся:

• снижение в источнике распространения;

• снижение на пути распространения, следующие методы:

– звукоизоляция;

– звукопоглощение;

– активный метод (использование антизвука);

2) Индивидуальные методы и средства.

Организационные методы направлены на снижение шума транспорта путем упорядочения транспортных потоков (т.е. уменьшения интенсивности движения в ночное время, использования обводных дорог и транспортных колец).

Строительно-архитектурные методы направлены на рациональную

планировку территорий, удаление жилых зданий внутрь кварталов, планирование зеленых насаждений.

Техническая группа является самой многочисленной. Методы снижения шума в источнике зависят от природы шума. Так механический шум уменьшают повышением точности обработки деталей и узлов, заменой металлических элементов неметаллическими в системах передачи движения.

Аэродинамический шум снижают используя глушители. Электромагнитный - применением композитных материалов в трансформаторах, дросселях; применением ферромагнитных материалов.

Снижение шума на пути распространения осуществляется звукоизоляцией, звукопоглощением и активным методом.

Звукоизоляция – метод снижения шума, основанный на отражении звуковой волны от преград, установленных на пути распространения. Звукоизоляция осуществляется веществами высокой плотности и большой удельной массы, таких как кирпич, железобетон и т.п.

Коэффициент звукопроницаемости рассчитывается по следующей формуле:

T = I2помещения / I1помещения,

I2помещения – интенсивность шума в помещении, где находится источник шума

I1помещения – интенсивность шума в пустом помещении

Звукоизоляция рассчитывается следующим образом

$$R = 10\lg(\frac{1}{\tau}) = 20\lg(\frac{P_{1\text{помещения}}}{P_{2\text{помещения}}}),$$

Звукоизоляция на практике рассчитывается экспериментально следующим образом (при условии, что звуковая волна распространяется

прямолинейно, без отражения):

$$R_0 = 20\lg f + 20\lg Q - 54,$$

где f – расчетная частота; Q - поверхностная плотность материала или вес одного квадратного метра данной толщины.

Звукоизоляция зависит от частоты. При повышении частоты увеличивается звукоизоляция. Увеличение массы в 2 раза ведет к увеличению звукоизоляции на 6 дБ.

Практически звукоизоляция применяется следующими способами:

- источник шума помещается в кожух;
- оператора помещают в изолирующую кабину;
- устанавливают звукоизолирующий экран.

Звукопоглощение - метод, основанный на снижении шума за счет потерь звуковой энергии в порах звукопоглощающего материала и переходе звуковой энергии в тепловую.

Для звукопоглощения применяются материалы, имеющие пористую структуру и обладающие небольшой массой.

Звукопоглощающие свойства определяются по следующей формуле:

$$\alpha = \frac{E_{\text{поглощения}}}{\text{E}_{\text{падающая}}}$$

где Е – энергия шума.

Ультразвук (УЗ) подразделяют на УЗ низкой частоты (до 100 кГц, распространяется воздушным и контактным путем) и УЗ высокой частоты (от 10 Гц до 10 Гц, распространение только контактным путем).

Ультразвук слабо поглощается в упругой среде и может распространяться на большие расстояния. Причем распространение может быть узконаправленным и иметь высокое звуковое давление.

Ультразвук оказывает влияние на вестибулярный аппарат, на сердечнососудистую систему и на нервные окончания.

Ультразвук регламентируется ГОСТ 10.1001-89*.

Методы защиты от ультразвука схожи с методами защиты от шума:

– снижение в источнике (увеличение рабочих частот);

– применение звукоизоляции и звукопоглощения;

– средства индивидуальной защиты (отдельно для органов слуха, отдельно для рук и ног).

Инфразвук определяется теми же параметрами, что и шум. Диапазон инфразвука лежит ниже 20 Гц. Основные источники инфразвука: двигатели, компрессором и т.д.

Особенность инфразвука в том, что он также как ультразвук может распространяться на большие расстояния и слабо поглощается в упругой среде. Воздействие на человека зависит от уровня, диапазона частот и времени воздействия. Инфразвук воздействуем на вестибулярный аппарат (иногда воспринимается как качка), па центральную нервную систему, вызывает чувство страха.

Опасный диапазон инфразвука 2... 15 Гц. Особо опасна частота 7 Гц. Для нее уровень в 150 дБ является смертельным для человека.

Нормирование инфразвука происходит в соответствии со СН-22-74-80.

Метод защиты от инфразвука только один: снижение в источнике. Под этим понимается: повышение быстроходности агрегата-источника инфразвука, повышение жесткости конструкции, балансировка и др.

Вибрация - это колебательный процесс, при периодическом смешении центра тяжести от положения равновесия при изменении формы.

Степень воздействия на человека зависит от уровня, вида вибрации и направления воздействия. По виду вибрация делится на:

– общую (сотрясение всего организма);

– локальную;

– комбинированную.

Источники вибрации делятся на три категории: транспортную, технологическую, комбинированную.

Общая вибрация может привести к вибрационной болезни, к смещению отдельных органов. Локальная вибрация может привести к спазмам сосудов, нарушению кровообращения. Диапазоны воздействия: f1=0,7.. 1,4 Гц; f2 = 6...9 Гц; f3 = 25...30 Гц. Нормирование вибрации осуществляется ГОСТ 12.1012-90*. Методы снижения вибрации:

1) Снижение в источнике посредством ликвидации возбуждающих сил. Балансировка отдельных узлов, повышение точности изготовления.

2) Отстройка от режима резонанса. Достигается рациональным выбором массы системы, режимов работы, повышением жесткости.

3) Вибродемпфирование осуществляется путем нанесения вязких покрытий (материалов), например различные мастики. Материалы такого плана делятся на две категории: жесткие (для высоких частот) и мягкие (для средних частот).

4) Вибропоглощение осуществляется путем установки агрегатов на специальные фундаменты или опорные плиты.

Виброизоляция. Уменьшение вибрации в этом случае обусловлено присоединением к объекту дополнительной упругой связи, при этом вибрация переходит в механические колебания в виброизоляторе.

Эффективность виброизоляции определяется коэффициентом передачи:

$$КП = \frac{V_{основния}}{V_{агрегата}},$$

где V – виброскорость.

В системах, где можно пренебречь трением, коэффициент передачи может быть определен по формуле:

$$КП = \frac{1}{(\frac{f_В}{f_0})^2 - 1},$$

где fВ – частота возмущающей силы; f0 - собственная частота системы, установлена а амортизаторы. Эффективная работа виброизолятора достигается при fв>f0*1.4.

Величина ослабления при виброизоляции: L = 20 lg (1 / КП).

6.3 Электробезопасность

Основные причины поражения электрическим током:

1) Нарушение изоляции или потеря изолирующих свойств изоляционного материала;

2) Опасное приближение к токоведущей части;

3) Несогласованное включение;

Поражение электрическим током имеет следующие воздействия на организм человека: термическое (нагрев и ожоги тканей): электролитическое; биологическое (возбуждение живых тканей); динамическое (разрыв тканей).

Поражения делятся на две группы: местных и общих поражений.

Местные, в соответствии с воздействием подразделяются на:

- электрические ожоги (термическое воздействие);
- электрические знаки;
- металлизация;
- механические повреждения.

Степени исхода поражений электрическим током:

1) судорожное сокращение мышц без потери сознания;

2) судорожное сокращение мышц потеря сознания, сохранение сердечной деятельности;

3) потеря сознания, нарушение сердечной деятельности;

Факторы, влияющие на степень опасности поражения электротоком:

1) Сила тока;
2) Длительность воздействия;
3) Сопротивление тела человека;
4) Путь тока через организм;
5) Род тока (переменный / постоянный);
6) Частота тока;
7) Индивидуальные особенности человека;
8) Окру жающая среда.

Установлены следующие пороговые знамения силы тока:

1) пороговый ощутимый ток:
   - для переменного (50 Гц): 0,6... 1,5 мА;
   - для постоянного: 5...7 мА;
2) пороговый неотпуекающий гок:
   - для переменного (50 Гц): 0,6... 1,5 мА;
   - для постоянного: 5...7 мА;
3) пороговый фпбрилдяцпонный ток (то есть нарушающий сердечную деятельность):
   - для переменного (50 Гц): 50...350 мА;
   - для постоянною: ~ 300 мА;

Сопротивление тела человека имеет сноп особенности. Так внутренние органы имеют низкое сопротивление. Основное сопротивление дает сухая кожа (от 2 • 10° Ом до 2 • 10° Ом). В наихудшем случае 1000 Ом.

С увеличением времени воздействия тока опасность возрастает. Переменный ток при значении напряжения до 300 В опаснее, чем постоянный. При значении напряжения выше 300 В, опасность поражения и постоянным и переменным током приблизительно одинакова. Для переменного тока наиболее опасный частотный диапазон: 20... 100 1 Гц.

Как уже было сказано, на исход поражения влияет путь тока через организм. В зависимости от точек прикосновения с токоведущимп частями,

ток пройдет (коротким путем) через руки, ноги, голову и т.д.

В зависимости от опасности поражения электрическим током, помещения подразделяются на: нормальные;

- влажные (влажность выше 70%);
- особо сырые (влажность от 90% и выше);
- жаркие (температуры выше 35 °С);
- пыльные;
- помещения с наличием химикатов.

Помещения также делятся по группам:

1) Без повышенной опасности электропоражения (сухие, беспыльные, с непроводящими полами и нормальной температурой);

2) Помещения с повышенной опасностью, то есть такие, у которых присутствует хотя бы один из следующих признаков:

- высокая температура (выше 35°С);
- высокая влажность (выше 70%);
- токопроводящие полы;
- токопроводящая пыль;
- возможность одновременного прикосновения человека к имеющим соприкосновение с землей металлическим конструкциям зданий сооружений, оборудования и корпусам электроустановок;

3) Особо опасные помещения (при наличии следующих признаков):

- высокая сырость (влажность около 100%);
- наличие органической или агрессивной среды;
- одновременное наличие двух или более признаков помещений с повышенной опасностью.

Электроустановки делятся на установки с напряжением до 1000 В и установки с напряжением свыше 1000 В. По характеру тока электрические сети делят на сети с постоянным током и сети с переменным током. Сети переменного тока делятся на однофазные и многофазные (наиболее

распространены трехфазные схемы).

Далее приведены существующие трехфазные сети:

1) Трехфазная трехпроводная сеть с изолированной нейтралью;
2) Трехфазная четырехпроводная сеть с изолированной нейтралью;
3) Трехфазная трехпроводная сеть с заземленной нейтралью;
4) Трехфазная четырехпроводная сеть с заземленной нейтралью.

В России, в случае, если напряжение меньше 1000 В. используются схемы 1) и 2).

Человек может соприкоснуться с трехфазной схемой по двум сценариям: однофазное и двухфазное включение. Двухфазное включение человека в электрическую цепь наиболее опасно.

Для защиты от электропоражения применяются:

1) Изоляция токоведущих частей - все токоведущие части имеют рабочую изоляцию. Изоляция должна выдерживать перегрузки в 3...4 раза превышающие возможные. Возможно применение дополнительной изоляции, это называется двойная изоляция. Нередко применяется одна, но усиленная изоляция, соответствующая двойной;

2) Оградительные устройства – экраны, кожухи и т.д. используются, чтобы избежать опасности прикосновения к токоведущим частям;

3) Электрическое разделение сетей необходимо для уменьшения опасной емкостной составляющей. Иногда используют переходные трансформаторы;

4) Применение малых напряжений (до 42 В) для питания ручных инструментов, местного освещения и др.;

5) Применение средств электрозащиты (подразделяются на основные, вспомогательные и дополнительные). Основные например, монтажный инструмент с диэлектрическими ручками или другими местами возможного соприкосновения. Вспомогательные резиновые перчатки, резиновые коврики, пояса, шлемы и т.д.;

6) Блокировка (существуют механические или 'электронные методы).

Например, блокировка в распределительном шкафу. Когда открываются дверцы шкафа, происходит размыкание токоведуших проводов;

7) Сигнализация (зуммеры, звонки, лампочки-оповестители и т.д.) и знаки безопасности (предупреждающие, запрещающие, указательные и предписывающие);

8) Защитное заземление - преднамеренное электрическое соединение с землей (или ее эквивалентом) металлических нетоковедущих частей, которые могут оказаться под напряжением. Защитное заземление устраняет опасность поражения персонала при замыкании (пробое) на корпус. В нормальном состоянии никаких функций не несет. Принцип действия: снижение до безопасного уровня напряжения прикосновения уменьшением потенциала. Применяется в трехфазной трехпроводной сети с изолированной нейтралью при напряжении до 1000 В;

9) Зануление – преднамеренное электрическое заземление с нулевым проводником металлических нетоковедущих частей, которые могут оказаться под напряжением. Принцип действия: прекращение пробоя на корпус в однофазное короткое замыкание, при котором срабатывает защитный элемент. Ток срабатывания защитного элемента зависит от его типа: это может быть плавкая вставка, либо автоматический размыкатель (электронный или электромеханический). При заземлении качество защиты будет зависеть от сопротивления земли, предохранитель может не сработать. Различают два нулевых проводника. Используется же нулевой защитный проводник, через который в обычном состоянии токи не проходят;

10) Защитное отключение быстродействующая защита, обеспечивающая автоматическое отключение электроустановки при возникновении опасности электропоражения. Применяется, когда все вышеперечисленные средства либо малоэффективны, либо трудновыполнимы, либо когда предъявляются повышенные требования к электробезопасности. Примером устройства защитного отключения может служить электронное устройство (время срабатывания - меньше 0,2 с),

состоящее из датчика и электронного размыкателя. Быстродействие такого устройства выше, чем быстродействие зануления.

В зависимости от электрической величины устройства защитного отключения могут быть:

- реагирующие на напряжение на корпусе;
- реагирующие на токи замыкания;
- напряжение и токи нулевой последовательности – такое устройство обладает очень высокой чувствительностью и защищает человека даже в случае прикосновения к фазному проводу в исправном рабочем режиме.

Допустимые значения токов и напряжений прикосновения регламентируются ГОСТ 12.1.083-82 . В зависимости от рода тока, частоты тока и времени воздействия устанавливаются допустимые напряжения:

- для нормального режима: 2 В и 0,3 мА;
- для бытовых установок: 12 В и 2 мА.

Действия при попадании человека под напряжение:

1) обесточить электроустановку;
2) отсоединить человека от сети, действуя одной рукой и держа человека за одежду;
3) при потере сознания необходимо сделать искусственное дыхание.

6.4 Воздействие электромагнитных полей

Как естественный фактор, воздействующий на человека, существует геомагнитное поле Земли. Область около земного пространства называется магнитосферой. Отдельные возмущения магнитосферы вызываются вспышками на Солнце. Это, так называемые, магнитные бури, они оказывают влияние на человека. Максимальное значение магнитное поле имеет на экваторе.

Искусственные источники электромагнитных полей - линии электропередачи, электрифицированный транспорт, технологическое

оборудование и другое.

Электромагнитные поля промышленных частот:

1) оказывают биологическое воздействие (воздействие на центральную нервную систему, головной мозг и т.д.)

2) могут вызывать электризацию больших металлических сооружений, изолированных от земли.

Электромагнитные поля излучают также радиостанции, средства связи, управляющее оборудование. В зависимости от мощности излучения радиопередающие станции подразделяются на группы:

1) малой мощности (до 5 кВт);
2) средней мощности (от 5 до 25 кВт);
3) большой мощности (от 25 до 100 кВт);
4) сверхмощные (свыше 100 kBt);

Степень воздействия электромагнитных полей на человека зависит от:

- интенсивности излучения;
- диапазона частот (НЧ, ВЧ, УВЧ и УКВ);
- продолжительности воздействия;
- характера излучения;
- размеров излучаемой поверхности;
- индивидуальных особенностей человека.

Электромагнитные поля оказывают два вида воздействия на человека:

1) тепловое - возникают ионные токи в организме человека и токи проводимости в диэлектриках – костная ткань, скелет и т.д. Может также возникать перегрев, ожоги, обугливание.

2) биологическое - возникает опасность для глаз и т.п.

Нормирование электромагнитных полей регламентирует ГОСТ 12.1.006.84*.

Методы защиты от электромагнитных полей:

1) Снижение в источнике - аттенюаторы применяются для изменения мощности. Если выходов много, то используют так называемые поглотители

мощности.

2) Экранирование – прозрачные или непрозрачные экраны могут быть поглощающие (из радиопоглощающих компонентов) и отражающие (из хорошо проводящих металлов).

3) Повышение сопротивления.

4) Уменьшение времени воздействия.

5) СИЗ.

6.5 Производственное освещение

Свет является электромагнитным излучением. Видимый диапазон света лежит на интервале длин волн от 380 нм до 750 нм. Диапазон длин волн от 1 нм до 380 нм – ультрафиолетовое излучение. От 750 нм до 1000 нм - инфракрасное излучение.

Свет определяется световым потоком, силой света, яркостью. Длительность восприятия определяется яркостью.

Помимо основных понятий, необходимо знать следующие:

Объект различения – наименьший размер рассматриваемого предмета или его часть, которая рассматривается в ходе выполнения работы.

Фон – поверхность, на которой рассматривается объект различения. Фон определяется коэффициентом отражения поверхности, обозначаемым буквой р. Фон может быть светлый ($p > 0.4$). средний ($p = 0,2...0,4$) и темный ($p < 0,2$).

Контраст объекта с фоном - отношение яркостей объекта и фона к яркости фона, обозначается буквой К. Контраст, в зависимости от соотношения яркостей, может быть: большой ($К > 0,5$), средний ($К = 0,2...0,5$). малый ($К < 0,2$). При $К = 0$ объект и фон различаются только по цвету.

Производственное освещение классифицируется следующим образом:

1) Естественное, подразделяется на:

– боковое (через проемы в окнах), бывает:

• одностороннее;

• двустороннее;

– верхнее;

– комбинированное;

2) Искусственное:

– общее:

• равномерное;

• локализованное (с учетом расположения рабочих мест);

- комбинированное (общее и местное). Использование только местного искусственного освещения запрещено.

3) совмещенное;

Искусственное освещение по функциональному назначению подразделяется на:

1) Рабочее - во всех помещениях с постоянным пребыванием людей:

2) Аварийное:

– освещение безопасности (в тех случаях, когда нельзя остановить работу);

– эвакуационное (для эвакуации персонала в случае отключения рабочего освещения);

3) Охранное - для освещения охранных предприятий;

4) Дежурное - используется в непроизводственное время.

Основные гигиенические требования, предъявляемые к производственному освещению:

1) Освещение рабочей поверхности должно соответствовать характеру выполняемых зрительных работ, согласно нормам.

2) Освещенность рабочей поверхности должна быть одинаковой

3) Отношение освещений рабочей поверхности к окружающей не должно быть более, чем 10:1.

4) Освещенность поверхности должна быть постоянной во времени.

5) Источники света должны иметь правильную светопередачу.

6) Осветительная установка должна быть простой и эстетичной.

Освещенность изменяется в очень широких пределах и зависит от многих факторов. Поэтому для оценки естественного освещения вводится показатель, который называется коэффициентом естественной освещенности (КЕО) и обозначается буквой е:

$$e = \frac{E_{\text{внутр}}}{\text{Е}_{\text{наружн}}}$$

где Е = Ф/S, Ф – световой поток, S - площадь поверхности.

КЕО зависит от размеров световых проемов (окон) и от расстояния от них.

Естественное освещение и КЕО регламентируется СНиП 23-05-95*. КЕО нормируется с учетом разряда работ, вида освещения и номера группы административного района местности.

Существуют восемь разрядов зрительных работ от работы наивысшей точности (с элементами разрешения до 0,15 мм) до работы с неопределенной зрительной точностью.

Нормирование искусственного освещения производится тоже СниП 23-05-95. При этом нормируемым параметром считается величина минимальной освещенности.

Величина освещенности устанавливается с учетом разряда зрительных работ, подразряда, системы освещения (общая или комбинационная) и типа источника света.

В зависимости от характера фона и контраста объекта с фоном первые пять разрядов подразделяются на четыре подразряда.

В нормах представлены значения освещенности для газоразрядных ламп. Для ламп накаливания нормированное значение снижается на разряд.

В нормах регламентируются и качественные показатели:

1) Показатель ослепленности рн = 20.. .60%;

2) Коэффициент пульсации для газоразрядных ламп:

Кп = (Emax - Emin)/2Есредн = 10.. .20%;

3) Спектральный состав источника света.

Источники света – основные составные части осветительной установки. Чтобы выбирать и сравнивать источники света их определяют следующими характеристиками:

1) электрические:

напряжение питания;

– род тока;

– потребляемая мощность;

2) светотехнические:

– сила света;

– световой поток;

3) эксплуатационные:

– световая отдача;

– срок службы (время, в течение которого световой поток уменьшается не более чем на 20%);

4) конструктивные:

– форма колбы;

– наличие газа;

– состав газа;

– давление газа.

Лампы можно разделить по принципу действия на:

1) Лампы накаливания;

2) Газоразрядные лампы.

У ламп накаливания низкая себестоимость, компактные размеры, световой поток к концу службы уменьшается менее чем на 15%. Однако

низкая светоотдача, малый срок службы, спектр света отличается от естественного (в основном желтые и красные составляющие).

Довольно приличные характеристики имеют кварцево-галогенные лампы: световая отдача 30 Лм/Вт, время службы » 3000 часов, спектр ближе к естественному.

Газоразрядные лампы отличаются от ламп накаливания тем, что не имеют раскаленной спирали. Основные преимущества газоразрядных ламп:

1) Светоотдача 70... 110 Лм/Вт;
2) Больший срок службы (8... 12 тыс. часов);
3) Возможно получение любого спектра.

Недостатки:

– более дорогостоящие лампы;

– условия эксплуатации зависят от питающего напряжения и температуры;

– лампы могут создавать радиопомехи;

– значительный коэффициент пульсации светового потока, что может привести к стробоскопическому эффекту.

Виды газоразрядных ламп:

1) Люминесцентные лампы;
2) Дуговые ртутные люминесцентные лампы;
3) Металлогенные лампы;
4) Дуговые ксеноновые трубчатые лампы;
5) Дуговые натриевые лампы.

Светильники основная часть осветительной установки. Состоит из источника света или лампы, узкорегулирующей аппаратуры и осветительной аппаратуры. Светильники различаются по конструктивному исполнению и светотехническим характеристикам.

Производственное освещение классифицируется следующим образом:

1) Естественное, подразделяется на:

– боковое (через проемы в окнах), бывает:

• одностороннее;

• двустороннее;

– верхнее;

– комбинированное;

2) Искусственное:

– общее:

• равномерное;

• локализованное (с учетом расположения рабочих мест);

– комбинированное (общее и местное). Использование только местного искусственного освещения запрещено.

3) совмещенное;

Искусственное освещение по функциональному назначению подразделяется на:

1) Рабочее – во всех помещениях с постоянным пребыванием людей;

2) Аварийное:

– освещение безопасности (в тех случаях, когда нельзя остановить работу);

– эвакуационное (для эвакуации персонала в случае отключения рабочего освещения);

1) Охранное - для освещения охранных предприятий;

2) Дежурное - используется в непроизводственное время.

Основные гигиенические требования, предъявляемые к производственному освещению:

1) Освещение рабочей поверхности должно соответствовать характеру выполняемых зрительных работ, согласно нормам.

2) Освещенность рабочей поверхности должна быть одинаковой.

3) Отношение освещений рабочей поверхности к окружающей не должно быть более, чем 10:1.

Лампы можно разделить по принципу действия на:

1) Лампы накаливания;

2) Газоразрядные лампы.

У ламп накаливания низкая себестоимость, компактные размеры, световой поток к концу службы уменьшается менее чем на 15%. Однако низкая светоотдача, малый срок службы, спектр света отличается от естественного (в основном желтые и красные составляющие).

Довольно приличные характеристики имеют кварцево-галогенные лампы: световая отдача 30 Лм/Вт, время службы ~ 3000 часов, спектр ближе к естественному.

Газоразрядные лампы отличаются от ламп накаливания тем, что не имеют раскаленной спирали. Основные преимущества газоразрядных ламп:

1) Светоотдача 70... 110 Лм/Вт;
2) Больший срок службы (8... 12 тыс. часов);
3) Возможно получение любого спектра.

Недостатки:

– более дорогостоящие лампы;

– условия эксплуатации зависят от питающего напряжения и температуры;

– лампы могут создавать радиопомехи;

– значительный коэффициент пульсации светового потока, что может привести к стробоскопическому эффекту.

Виды газоразрядных ламп:

1) Люминесцентные лампы;
2) Дуговые ртутные люминесцентные лампы;
3) Металлогенные лампы;
4) Дуговые ксеноновые трубчатые лампы;
5) Дуговые натриевые лампы.

Светильники – основная часть осветительной установки. Состоит из источника света или лампы, узкорегулирующей аппаратуры и осветительной аппаратуры. Светильники различаются по конструктивному исполнению и светотехническим характеристикам.

Заключение

В рамках данного проекта были разработаны рекомендации по обеспечению защиты сайта от сканирования и хаотичных интенсивных запросов с помощью специальных методов противодействия.

Защита от сканирования:

- использование временной задержки между запросами при превышении порога запросов в определенный промежуток времени;

- использование на сервере mod_rewrite;

- отказ от использования указания закрытых частей сайта в файле robots.txt;

- использование нестандартных способов перенаправления;

- задание правильной настройки доступа для поисковых роботов;

- регулярное обновление (замена и удаления) скриптов;

- для закрытых частей сайта использование протокола SSL;

- проведение регулярного аудита сайта на защищенность;

Защита от хаотичных интенсивных запросов:

- использование временной задержки между запросами при превышении порога запросов в определенный промежуток времени;

- принуждение пользователей изменять время от времени свои пароли и создавать их с определенной сложностью;

- создание запрета на отправку абсолютно одинаковых сообщений;

- предоставление возможности отправки сообщения лишь определенным пользователям;

- блокировка ip постоянно отсылающих недопустимо большое количество запросов в определенный промежуток времени.

- установка в файле robots.txt ограничение запроса по времени для поисковых роботов;

- установка в файле sitemap.xml времени на запрос определенных страниц сайта для поисковых роботов.

Также был разработан php-сценарий обеспечивающий один из самых эффективных способов по защите сайта от сканирования и хаотичных интенсивных запросов - установка ограничения на количество запросов и задание списков всегда разрешенных и запрещенных IP адресов. Этот же сценарий помогает защитить сайты от взлома паролей методом перебора (брутфорса) делая его неэффективным или даже бесполезным.

В экономической части диплома была подтверждена экономическая обоснованность разработанного php-сценария и используемых методов по защите сайта. Так как они в совокупности с представленными методами снижает затраты, как в экономическом плане так и в плане ресурсоемкости по сравнению с IDS, IPS и другими дорогостоящими способами и методами защиты. В частности не требуется дополнительное оборудование и дорогостоящее платное программное обеспечение.

В части посвященной охране труда и окружающей среды были рассмотрены проблемы негативного влияния шума, инфразвука, ультразвука, вибрации и воздействие электромагнитных полей с мерами по уменьшению их негативного воздействия на людей. А также меры безопасности при работе с электрическим током.

Перечень ссылок

1. Стюарт Мак-Клар, Саумил Шах, Шрирай Шах. Хакинг в Web: атаки и защита: Пер. с англ.. – М.: Издательский дом «Вильямс», 2003. – 384 с.
2. Джоел Скембрей, Майк Шема. Секреты хакеров. Безопасность Web-приложений – готовые решения: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 382 с.
3. Стюарт Макклуре, Джоел Скембрэй, Джордж Курц. Секреты хакеров. Проблемы и решения сетевой защиты: Пер. с англ. – М.: Издательство «Лори», 2001. – 435 с.
4. Фленов М.Е. Web-сервер глазами хакера. – СПб.: БХВ-Петербург, 2007. – 288с.: ил.
5. Стюарт Мак-Клар, Джоел Скембрей, Джордж Курц . Секреты хакеров. Безопасность сетей – готовые решения, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 736 с.: ил.
6. Alex WebKnacKer. Быстро и легко. Хакинг и антихакинг: защита и нападение. Учебное пособие. – М.: Лучшие книги, 2004. – 400 с.: ил.
7. Уэнстром М. Организация защиты сетей Cisco.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 768 с.
8. Фленов М.Е. PHP глазами хакера. – СПб.: БХВ-Петербург, 2005. – 304 с.: ил.
9. Ашманов И., Иванов А. Оптимизация и продвижение сайтов в поисковых системах. – СПб.: Питер, 2008. – 400 с.: ил.
10. Брэгг Роберта. Безопасность сети на основе Microsoft Windows Server 2003. Учебный курс Microsoft / Пер. с англ. – М.: Издательско-торговый дом «Русская Редакция», СПб.: «Питер», 2006. – 672 стр.: ил.
11. Норткат Стивен, Новак Джуди. Обнаружение нарушений безопасности в сетях, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 448с.: ил. – Парал. тит. англ.
12. Щеглов А.Ю. Защита компьютерной информации от

несанкционированного доступа. – СПб.: «Наука и техника», 2004. – 384 с.

13. Лаура Томпсон, Люк Веллинг. Разработка Web-приложений на PHP и MySQL: Пер. с англ. / Лаура Томсон, Люк Веллинг. – 2-е изд., испр. – СПб: ООО «ДиаСофтЮП», 2003. – 672 с.

14. Калачанов В.Д., Кобко Л.И. Экономическая эффективность внедрения информационных технологий. Уч. пособ. Гриф УМО Минобрнауки России. – М.: Издательство МАИ, 2006.

15. Положение по бухгалтерскому учету «Учет расходов на научно-исследовательские, опытно-конструкторские и технологические работы» (ПБУ 17/02).

16. ССБТ ГОСТ 12 О.Т. м. 1990г.

17. Бобков Н.И. Охрана труда. МАИ 1995г.

18. Яров В.Н. Выбор и расчет элементов электрозащиты. МАИ 1999г.

Доклад

[слайд №1]

Тема моей выпускной квалификационной работы – “Разработка метода защиты сайта от сканирования и хаотичных интенсивных запросов”

Хаотичные интенсивные запросы сильно нагружают сервера и транспортные каналы, существенно замедляя работу сайта.

С помощью сканирования злоумышленники копируют содержимое сайтов и выявляют слабые стороны в их защите, нанося при этом значительный ущерб. Кроме того, запросы к сайту, производящиеся в процессе сканирования, также отрицательно влияют на производительность.

Поэтому целью моей работы является разработка рекомендаций по защите веб-ресурса от сканирования и хаотичных интенсивных запросов.

[слайд №2]

Чаще всего проблема медленной работы сайтов касается крупных порталов с высокой посещаемостью. Но она может коснуться и небольших сайтов, так как даже при малой посещаемости сайт может подвергаться высокой нагрузке. Высокая нагрузка создается различными роботами, постоянно сканирующими сайты. При этом работа сайта может сильно замедлиться, или он вообще может оказаться недоступным.

Сканирование сайта производится программами, сторонними сайтами или вручную. При этом создается большое количество запросов в короткий промежуток времени. Сканирование сайта чаще всего используют для поиска в нем уязвимостей или копирования содержимого сайта.

Далее в работе были разработаны рекомендации по защите вэб-сайта от сканирования, приведенные на слайде 3.

[слайд №3]

Проанализировав виды сканирования, я разработал рекомендации по защите сайта от них. Достаточно эффективной мерой защиты сайта от сканирования будет разграничение прав доступа к ресурсам сайта.

Информацию о структуре сайта поможет скрыть модуль apache mod_rewrite изменяющий ссылки. А сделать неэффективным сканирование ссылок и, одновременно, снизить нагрузку поможет установка временной задержки между частыми запросами исходящими от одного пользователя. Для поддержания эффективной защиты от сканирования и хаотичных интенсивных запросов необходим регулярный аудит вэб-ресурсов.

[слайд №4]

Хаотичные интенсивные запросы – это случайные или злонамеренные многочисленные запросы в короткий промежуток времени на страницы сайта со стороны пользователей или роботов. К примеру случайных интенсивных запросов относится частое обновление страницы. К злонамеренным многочисленным запросам относится спам на страницы сайта со стороны пользователей или DoS атаки.

[слайд №5]

К хаотичным интенсивным запросам так же относится способ подбора паролей методом перебора. Подобрать пароль можно как вручную, так и при помощи специальных программ. Вручную пароль подбирается лишь в тех случаях, когда известны его возможные варианты. В других случаях используются специальные программы, осуществляющие автоматический подбор пары логина и пароля, т.е. программы для брутфорса.

[слайд №6]

На основании проанализированной угрозы хаотичных интенсивных запросов мною были разработаны рекомендации для обеспечения стабильной работы сайта путем уменьшения их интенсивности. К эффективным методам защиты сайта от хаотичных интенсивных запросов относятся: установка временной задержки между запросами в определенный промежуток времени, создание черного и белого списков, установка для поисковых систем временной задержки между запросами страниц сайта в файле robots.txt и установка периода обновления страниц в файле sitemap.xml.

[слайд №7]

В практической части мной был реализован один из методов по защите сайта от сканирования и хаотичных интенсивных запросов, который заключается в подсчете количества запросов в определенный промежуток времени и установке временной задержки при превышении установленного порога. В частности этот метод делает неэффективным или даже бесполезным способ взлома пароля путём перебора, потому что затраченное на перебор время будет слишком велико.

[слайд №8]

На слайде представлен пример работы разработанного мною php-сценария. Данный сценарий включается в страницы сайта с помощью php метода include, в котором указывается относительный или абсолютный путь до php-сценария.

Рассмотрим работу разработанного мною php-сценария с целью защиты сайта от сканирования и хаотичных интенсивных запросов:

1) Данный блок служит для определения IP адреса пользователя или робота.

2) Вот эти два блока служат для реализации списка IP адресов с неограниченным доступом и всегда заблокированных. Т.е. черных и белых списков.

3) Следующий блок реализует проверку интенсивности запросов, исходящих от пользователей или роботов.

4) Остальные блоки отслеживают время блокировки.

Результаты работы разработанного мною php-сценария представлены на следующих слайдах.

[слайд №9]

Эта страница выдается, если пользователь превысит установленный порог запросов в указанный промежуток времени. Выдаваемое сообщение можно настроить путем простого редактирования php файла со сценарием в любом текстовом редакторе.

[слайд №10]

Эта страница будет показана пользователю, если он находится в черном списке. Сообщение на ней так же можно настроить путем изменения текста в php файле со сценарием.

[слайд №11]

Сайт с внедренным php-сценарием был протестирован в web-сервисе www.pr-cy.ru на доступность при многочисленных запросах. В скрипте был установлен порог в 3 запроса за 15 секунд. После превышения этого порога ip был заблокирован, что мы и видим на представленном скриншоте.

В заключении можно добавить: использование разработанного мною php-сценария с рекомендованными мерами обеспечивает стабильное функционирование сайта при хаотичных интенсивных запросах и защищает от сканирования.

В экономической части диплома была подтверждена экономическая обоснованность разработанного php-сценария и используемых методов по защите сайта.

В части посвященной охране труда и окружающей среды были проанализированы меры по уменьшению негативного воздействия на людей при работе с разработанными методами по защите от сканирования и хаотичных интенсивных запросов.