МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет математики

Кафедра безопасности информационных систем

**ПРИЛОЖЕНИЕ**

**К ОТЧЕТУ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ**

по специальности 10.05.01 Компьютерная безопасность

(уровень специалитета)

специализация №7 «Информационно-аналитическая и техническая экспертиза компьютерных систем»

Студент группы 6642-100501D\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.А.Бизин

Руководитель практики

доцент кафедры безопасности

информационных систем,

к.ф.-м.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_М.Е.Федина

Самара 2018

Тема выпускной квалификационной работы:

«Проектирование и разработка веб-фильтра для обеспечения контроля доступа к сетевым ресурсам»

Научный руководитель: доцент кафедры безопасности информационных систем, к.ф.-м.н. Федина М.Е.

**Индивидуальное задание на преддипломную практику**

На период прохождения практики была поставлена цель – написание первой главы выпускной квалификационной работы и проработка практической ее части. Объявлены следующие задачи:

* разработать и составить план первых двух глав выпускной квалификационной работы;
* проанализировать предметную область, изучить виды прокси-серверов, определить задачи, которые они выполняют;
* изучить принципы работы протоколов *HTTP*/*HTTPS*;
* проанализировать существующие методы и алгоритмы контентной фильтрации;
* разработать требования к программному обеспечению.

**Результаты, полученные за период прохождения практики**

За время прохождения практики были выполнены все поставленные задачи и достигнута цель практики: написана первая глава выпускной квалификационной работы и проработана ее практическая часть.

В начале прохождения практики был проведен анализ предметной области, который заключался в изучении теории и поиске информации о прокси-серверах, их видах и задачах, которые они решают. Также проводился поиск информации по контент-фильтрам (веб-фильтрам). В результате полученной информации сформировались требования к разрабатываемому веб-фильтру: многопоточный режим работы, фильтрация контента «на лету», поддержка механизма черных списков *IP*/*URL*, работа с протоколами *HTTP* и *HTTPS*, возможность управления веб-фильтром через пользовательский интерфейс (веб-консоль).

Также при анализе предметной области проводился поиск научных статей по методам контентной фильтрации. Был проведен обзор алгоритмов классификации текста, выявление сильных и слабых сторон этих алгоритмов. В результате данного исследования был выбран наивный байесовский классификатор в качестве алгоритма классификации контента для разрабатываемого веб-фильтра. Также были изучены два метода применения наивного байесовского классификатора – многомерная и мультиномиальная модели.

Для написания второй главы ВКР проводился поиск информации по серверным технологиям, которые будут задействованы при реализации веб-фильтра: протоколы *HTTP*/*HTTPS*, язык программирования *Java*, технология *Java Servlets*, *Java* *API* для работы с *TCP*-сокетами, библиотека для создания веб-сервера *Jetty*. Также проводился поиск информации по клиентским технологиям: язык программирования *JavaScript*, технология асинхронного взаимодействия между клиентом и сервером *AJAX*, библиотеки для упрощенного создания веб-интерфейса *jQuery* и *Bootstrap*. Также проводился поиск информации по способам хранения настроек программ, и в качестве хранилища была выбрана легковесная встроенная база данных *H2*. Так как язык *Java* был выбран для реализации веб-фильтра, то было приведено обоснование данного выбора.

Особое внимание было уделено изучению протокола *HTTP*, так как веб-фильтр анализирует контент, который передается по данному протоколу. Хотя веб-фильтр реализован для фильтрации контента в формате *HTML*, небольшими изменениями можно добавить поддержку фильтрации других форматов (например, текстовых, *XML*, *JSON*).

Перед классификацией контент следует подготовить: убрать незначащие для классификации слова (союзы, предлоги, частицы) и элементы (теги языка разметки *HTML*) и провести процедуру стемматизации (выделение основы слов). В связи с этим проводился поиск информации по библиотекам, которые могут помочь с данными процедурами. В итоге были выбраны библиотеки *Apache Lucene Morphology* (для стемминга) и *Jsoup* (для представления *HTML*-документа в виде дерева *DOM* и удаления тегов).

Для того, чтобы веб-фильтр мог анализировать контент, передаваемый по протоколу *HTTPS*, предварительно нужно этот контент расшифровывать. Но так как выработка общего симметричного ключа между браузером и веб-сервером происходит при помощи асимметричной криптографии, то в данной ситуации можно применить технологию *Trusted MITM* (*Trusted* в данном случае означает, что пользователь веб-фильтра знает о применении подхода *MITM* и согласен на это). Веб-фильтр в данной схеме находится посередине между браузером и веб-сервером и устанавливает доверенное соединение между собой и браузером при помощи собственного корневого самоподписанного сертификата формата *X.509*. Данный корневой сертификат необходимо импортировать в ОС компьютера, на котором находится браузер. При обращении к какому-либо сайту веб-фильтр генерирует сертификат формата *X.509* для домена сайта и подписывает его своим корневым. Получив сертификат, браузер видит, что он подписан корневым, которому он доверяет (предварительно импортировали его), и позволяет установить соединение. Установление соединения происходит по протоколу *TLS* при помощи механизма хэндшейков (обмен рукопожатиями). Поэтому также проводился поиск, анализ и изучение полученной информации по темам: *HTTPS* протокол, технология *Trusted MITM*, сертификаты формата *X.509*, протокол *TLS*, библиотека *OpenSSL* (для генерации корневого самоподписанного сертификата), библиотека *BouncyCastle*.

Для проектирования веб-интерфейса контент-фильтра проводился поиск информации и изучение архитектуры построение веб-приложений *REST*. Также для реализации клиентской части проводился анализ библиотек, которые могут быть применены. В результате были выбраны библиотеки *jQuery* и *Bootstrap*. Также была изучена технология асинхронного взаимодействия между браузером и веб-сервером *AJAX*.

Во время написания третьей главы ВКР проводилась реализация веб-фильтра. Реализацию можно разбить на несколько этапов:

1. На первом этапе проводилась разработка общего каркаса веб-фильтра в виде прокси-сервера на *TCP*-сокетах в многопоточном режиме. На данном этапе было реализовано простое пропускание *HTTP*-трафика между веб-сервером и браузером через веб-фильтр с использованием многопоточности, т.е. каждый запрос браузера выполнялся в отдельном потоке.
2. На втором этапе проводилась разработка серверной части веб-фильтра с применением технологии *Java Servlets* и библиотеки для создания веб-сервера *Jetty*. Данный этап решал задачу управления веб-фильтром через веб-консоль со стороны сервера. Были реализованы такие функциональности, как запуск и остановка веб-фильтра, сохранение и применение настроек, отслеживание статуса веб-фильтра, сохранение *IP*-адреса или *URL* хоста в черный список.
3. На третьем этапе проводилась разработка компонентов, отвечающих за работу с базой данных *H2*. База данных работает во встроенном режиме и представляет собой один файл. На данном этапе была определена структура базы данных: таблица настроек и таблица черного списка, а также реализованы *Java* классы для выполнения *CRUD* операций с данными таблицами. В коде было оформлено в виде паттерна проектирования *DAO*.
4. На четвертом этапе проводилась разработка клиентской части веб-фильтра – веб-интерфейса для управления контент-фильтром – с использованием *JavaScript*, библиотек *jQuery*, *Bootstrap* и применением технологии *AJAX*. На данном этапе были сделаны страницы *HTML* с управляющими *JavaScript* скриптами. Технология *AJAX* применялась для получения состояния веб-фильтра по таймеру.
5. На пятом этапе проводилась разработка компонентов, отвечающих за обучение классификатора, разбор, обработку (стемминг) и классификацию текста. На данном этапе был реализован байесовский классификатор, на вход которому подается список обработанных слов (текст страницы) и который возвращает карту категория -> вероятность попадания данного текста в эту категорию. Также были реализованы компоненты, которые предварительно обрабатывают слова при помощи библиотек *Apache Lucene Morphology* и *Jsoup*.
6. На шестом этапе проводилась разработка компонентов, предназначенных для генерации поддельных сертификатов формата *X.509* и обработки процедуры хэндшейков в протоколе *TLS*. Для разработки использовалась криптографическая библиотека *BouncyCastle*.

**Работы, необходимые для завершения ВКР**

Для завершения ВКР необходимо провести тестирование работы веб-фильтра по следующим параметрам:

1. Замерить общую скорость загрузки веб-страницы, пропущенной через веб-фильтр, для *HTTP* и *HTTPS* протоколов и сравнить ее со скоростью загрузки без фильтрации.
2. Замерить время генерации сертификатов с учетом кэширования и без. При генерации сертификатов реализовано их кэширование для ускорения последующих запросов к этому же хосту. Кэш в данном случае – это карта хост -> сгенерированный сертификат для данного хоста. При повторном обращении к данному хосту сертификат уже не будет генерироваться, а будет браться из кэша.
3. Замерить время классификации веб-страниц и время обучения классификатора. Классификатор обучается при старте веб-фильтра, поэтому данное время будет влиять только на время запуска веб-фильтра.
4. Также следует провести оценку качества классификации веб-страниц.

**Отзыв**

За время прохождения практики были выполнены все поставленные задачи, проанализирована предметная область и разработано программное обеспечение. Кроме того, была написана статья и поданы материалы для участия в конференции «ИТНТ-2019» по предмету разработанного программного обеспечения. Хочется отметить отличную проработку студентом предметной области, грамотное владение инструментарием создания программного обеспечения и качественное оформление результатов своей работы.

Консультант практики

старший преподаватель

кафедры безопасности

информационных систем,

к.ф.-м.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.А.Бурлов