УДК 004.056.53+004.021

СИСТЕМА ЗАЩИТЫ ВЕБ-ПРИЛОЖЕНИЙ

Ванюков Андрей Юрьевич, *ст. преп., КГТУ им. И.Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Ч.Айтматова 66. Тел: +996 (312) 542986, e-mail: vanjukov@rambler.ru, ORCID ID 0000-0003-4861-8945*

Чепашева Татьяна Сергеевна, *ст. преп., КГТУ им. И.Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Ч.Айтматова 66. Тел: +996 (312) 542986, e-mail: tatianas.chepasheva@list.ru, ORCID ID 0000-0002-2340-5006*

Аннотация. Рассматривается система защиты, которая предназначена для блокирования веб-приложений от основных видов атак. Суть заключается в тщательной проверке данных, которые поступают в приложение от пользователя. Устанавливается на том же сервере, где находится веб-приложение. Дополняет штатные механизмы защиты и работает на уровне приложения. Рассмотрены существенные виды атак, от которых

защищает данная система. Приведен алгоритм работы с подробным описанием каждого этапа. Разработанное программное обеспечение включает в себя предварительную обработку данных, аутентификацию, авторизацию, проверку адреса и переданных данных, анализ истории посещений. Для решения о допуске к приложению, рассчитывается сумма баллов, которая характеризует степень неправильного ввода параметров пользователем. Если эта сумма не превышает порог, то абонент допускается к работе и переданные им данные транслируются веб-приложению для последующей обработки. Блокировка пользователей происходит, если их поведение не отвечает логике приложения и сумма превышает пороговый уровень. В работе системы учитываются предыдущие действия в системе. При этом есть определённый запас, учитывающий неумышленные ошибки посетителя. Все данные регистрируются в журнале для дальнейшего анализа администратором. Предлагаемая система опробована на практике и показала свою эффективность.

Ключевые слова*: веб-приложение, система защиты, атака, уязвимость, сервер, пользователь, запрос, данные, база данных*

SECURITY SYSTEM OF WEB APPLICATIONS

Vaniukov Andrei, *senior teacher, KSTU named after I.Razzakov, Kyrgyzstan, 720044, c. Bishkek, Ch. Aitmatov avenue, 66. Phone: + 996 (312) 542986, e-mail: vanjukov@rambler.ru, ORCID ID 0000-0003-4861-8945*
Chepasheva Tatiana, *senior teacher, KSTU named after I.Razzakov, Kyrgyzstan, 720044, c. Bishkek, Ch. Aitmatov avenue, 66. Phone: + 996 (312) 542986, e-mail: tatianas.chepasheva@list.ru, ORCID ID 0000-0002-2340-5006*

Abstract. The security system which is intended for blocking of web applications from key types of attacks is considered. It is based on check of the data transferred to application from the user. It is installed on the same server where the web application is located. It supplements the staff protection mechanisms and works at the application level. Essential types of attacks from which the given system protects are considered. The algorithm of operation with the detailed description of each stage is resulted. The developed software includes preliminary processing of data, authentication, authorization, verification of address and transmitted data, analysis of the history of visits. To decide on the user's admission, the score that characterizes a level of wrong input of parameters the user is calculated. If this amount does not exceed the threshold, then the subscriber is allowed to work and the data transmitted to him is broadcast to the web application for further processing. Lock of users happens, if their behavior does not answer logic of the web application and the total exceeds trigger level. The system takes into account the previous actions in the system. At the same time, there is a certain reserve, which takes into account unintentional errors of the visitor.All data from system is registered in log for the subsequent analysis the administrator. The offered system is tested in practice and showed the efficiency.

Keywords: web application, security system, attack, vulnerability, server, user, inquiry, data, database

Введение

Развитие Интернет во всем мире привело к появлению веб-приложений, которые широко используются для вычислений, обработки данных и поиска информации. Работают веб-приложения по стандартной для сети Интернет технологии «клиент-сервер» [5]. Клиент формирует и посылает запрос, а сервер даёт ответ на запрос. Клиентом в веб-ресурсах является браузер, который создаёт запросы к серверу, обрабатывает ответы от него и реализует интерфейс. Серверная часть обрабатывает запросы, осуществляет вычисления,

создаёт страницу и отсылает её клиенту по сети Интернет.

Проблема обеспечения информационной безопасности является актуальной, так как популярные веб-приложения часто становятся жертвой злоумышленников, несмотря на наличие элементов защиты. Так как такие проекты являются достаточно сложными по своей структуре и включают большой объём разнородного программного обеспечения, то создать и настроить приложение с высоким уровнем защиты достаточно трудно.

Целью исследования является создание системы защиты от атак, ориентированных на веб-приложение, которая основана на аудите данных, передаваемых от пользователя.

Постановка задачи

В области безопасности веб-ресурсов авторитетной является организация OWASP (Open Web Application Security Project) [2]. Согласно данным этой ассоциации, а также данным других источников [3,4,6,8,9] составлена таблица с перечислением уязвимостей веб-приложений, которые могут привести к успешным атакам, направленным на веб-ресурсы.

Таблица 1.

Основные информационные риски веб-приложения

№	Краткое название уязвимости	Защита на уровне системы	Защита вне системы
1	Инъекции	+	
2	Некорректная аутентификация	+	
3	Недостаточно защищённые данные		+
4	Некорректная авторизация	+	
5	Неправильная начальная конфигурация		+
6	Межсайтовый скриптинг	+	
7	Использование компонентов с уязвимостями		+
8	Неэффективный мониторинг	+	

Рассмотрим кратко уязвимости, перечисленные в таблице 1.

1. Если данные от пользователя содержат специальные команды и символы, к которым чувствительны базы данных или интерпретатор, то появляется возможность просматривать, редактировать и удалять базы данных или даже управлять сервером. Встречается ситуация, когда данные без какой-либо обработки поступают на интерпретатор или в базу данных.

2. Данная уязвимость может привести к обходу механизмов аутентификации веб-серверов. Злоумышленник, не имея учетной записи в веб-приложении, может попытаться войти в систему, минуя штатную процедуру аутентификации. Например, нарушитель может подбирать автоматически пароли, если приложение это позволяет. В приложении могут использоваться слабые пароли.

3. Данные доступа к банковским картам, медицинские карты и персональные данные должны хранится в зашифрованном виде в базе данных и передаваться по протоколу HTTPS. Также причиной уязвимости может являться слабое шифрование и сохранение резервных копий в открытом виде.

4. Серьёзной ошибкой приложения является некорректная авторизация. Во многих веб-приложениях существуют категории пользователей, которым доступна разная информация и неодинаковые действия. Для этого веб-приложение выполняет процедуру авторизации. Используя уязвимость приложения, злоумышленник может повысить свои привилегии и получить доступ к закрытым ресурсам. Иногда пользователям удаётся получить доступ к интерфейсу администратора, что даёт широкие возможности для организации атак.

5. Многие составляющие веб-приложения, например база данных, интерпретатор, веб-сервер устанавливаются с настройками по умолчанию, которые не являются правильными с точки зрения безопасности, поэтому требуется настройка каждой системы в соответствии с обеспечением защиты информации.

6. Серверу передаётся в данных пользователя исполняемый код на языке JavaScript/HTML. Вредоносный скрипт в этом случае выполняется на компьютере клиента. Чаще всего данная уязвимость используется для перехвата сессий.

7. Многие веб-приложения работают на основе программных модулей сторонних производителей, в которых случайно или намеренно введены уязвимости. Также может использоваться программное обеспечение, которое устарело и не поддерживается. Встречается ситуация, когда на сервере размещено программное обеспечение, не обязательное для его работы.

8. Для поддержания надлежащего уровня защиты требуется регулярное ведение и просмотр журналов сервера с ошибками доступа. Следует отметить, что журналы веб-сервера не удобны для аналитики, поэтому желательно дополнительное ведение журнала приложением.

Рассмотрев уязвимости, приходим к выводу, что основной причиной проблем в безопасности является слабая фильтрация данных, поступающих в веб-приложение. В таблице 1 указано, от каких уязвимостей защищает разработанное программное обеспечение, а какие требуют дополнительных методов, например, сетевого экрана [1]. Для разрабатываемой системы защиты зададим следующие требования:

- высокий уровень защиты от большинства атак;
- низкая степень ложных срабатываний;
- простая в настройке и удобная в работе;
- полная интеграция с приложением.

Система защиты создаётся под будущее веб-приложение на этапе его разработки и является «прослойкой» между веб-приложением и веб-сервером (рис.1). Когда пользователь приложения работает с ним, по протоколу HTTP/HTTPS передаются запросы и ответы. Запросы поступают на веб-сервер, который передаёт их системе защиты, где все переданные от пользователей данные проверяются на допустимость, фильтруются и регистрируются. В случае если пользователь допускается, то данные посетителя передаются веб-приложению.

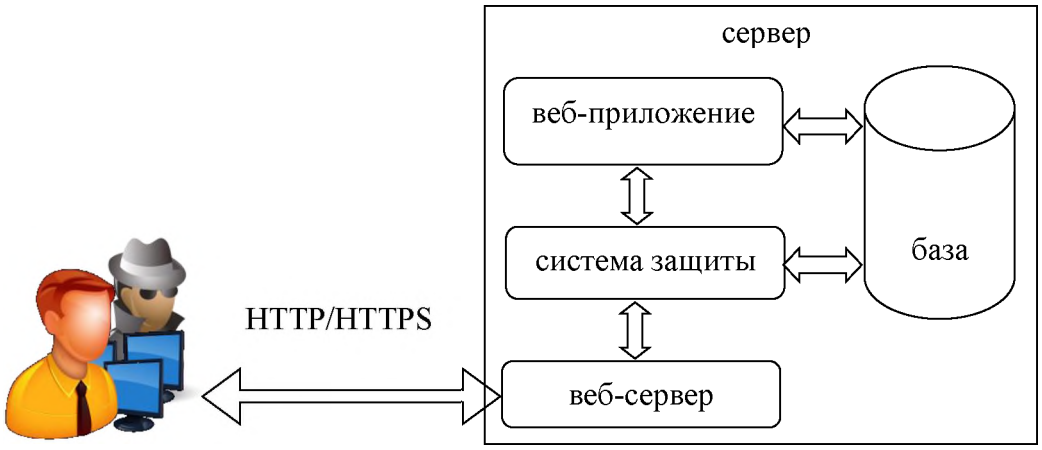


Рис. 1.

Решение задачи

Протокол HTTP обладает двумя основными методами: GET и POST. Оба метода предназначены для передачи параметров сценариям. С помощью небольших файлов

COOKIE происходит идентификация браузера, данный параметр часто используется для организации систем аутентификации.

Для создания веб-приложений на серверной стороне применятся различные языки программирования и технологии. Одним из самых известных языков для разработки является язык PHP [7,9], причиной этого является относительная простота, высокая скорость выполнения сценариев, хорошая документация, а также поддержка большим числом серверов.

Предлагаемая система защиты представляет собой модули, написанные на языке программирования PHP. Выбор языка обусловлен тем, что само веб-приложение написано на этом языке. Система защиты регистрирует данные (рис. 2), которые затем анализируются на допустимость.

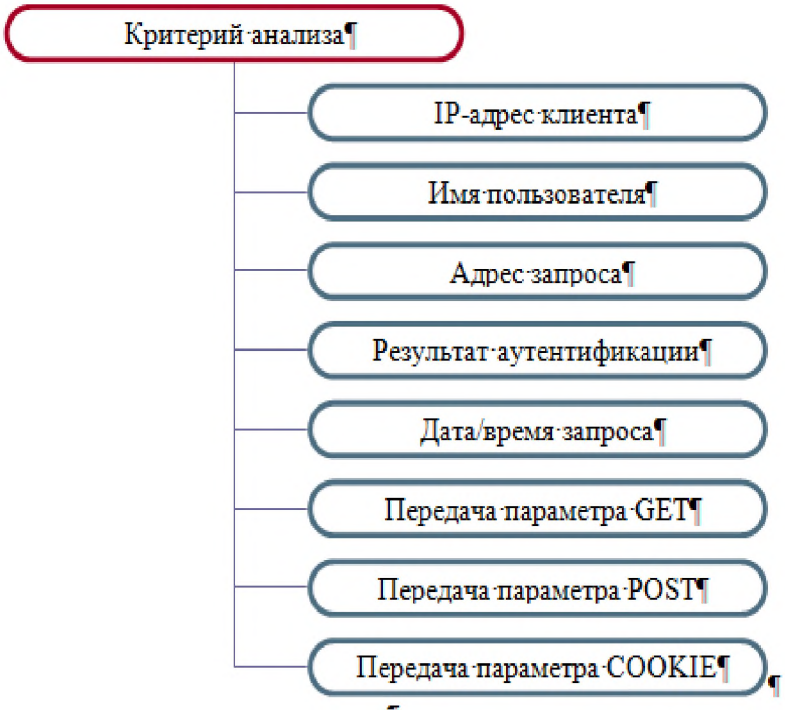
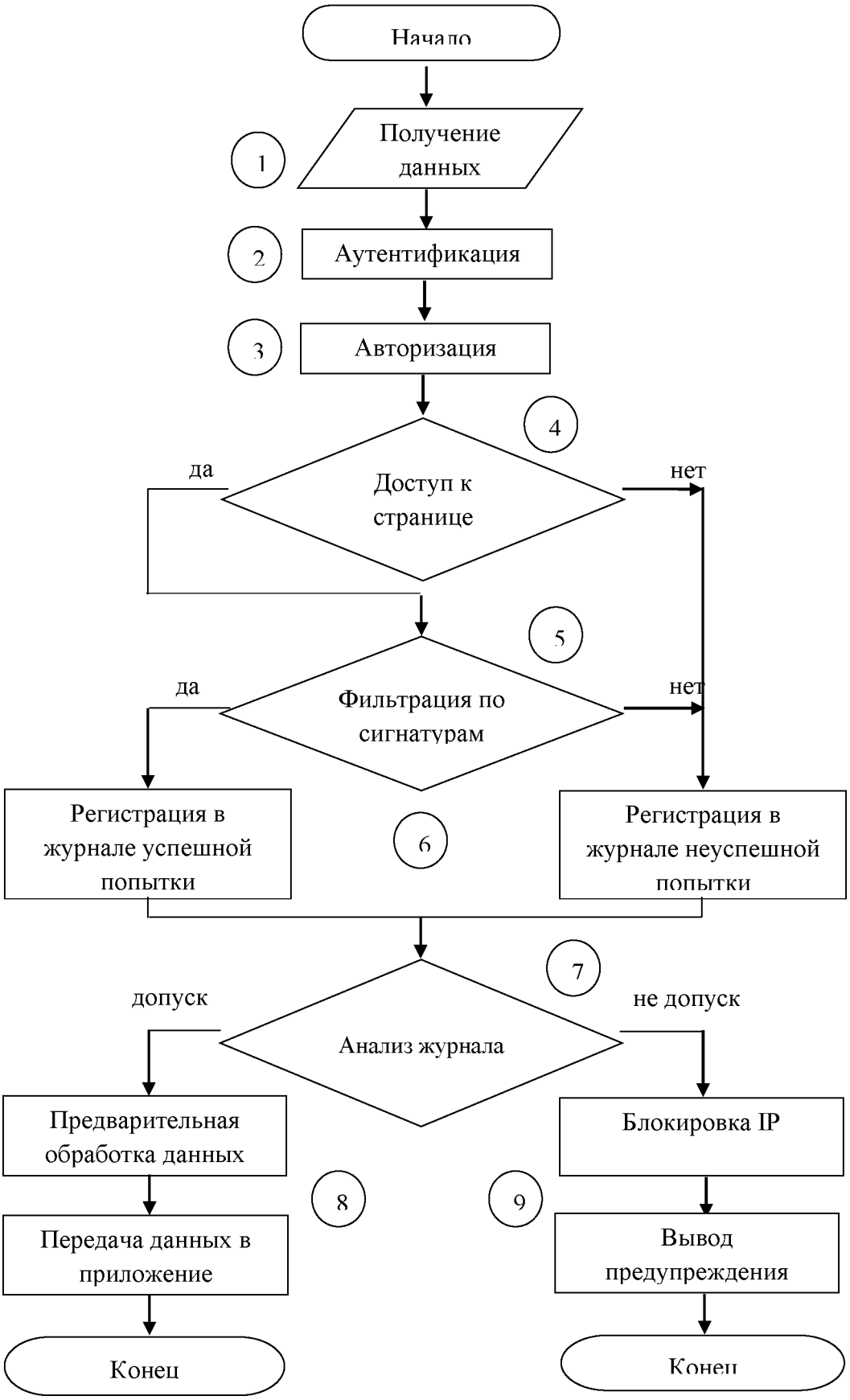


Рис. 2.

На рис.3 приведена блок-схема, отражающая работу системы защиты от атак. Алгоритм работы программного обеспечения включает несколько этапов.


Начало
1
Получение данных
2
Аутентификация
3
Авторизация
4
Доступ к странице
да
нет
5
Фильтрация по сигнатурам
да
нет
Регистрация в журнале успешной попытки
6
Регистрация в журнале неуспешной попытки
7
Анализ журнала
допуск
не допуск
Предварительная обработка данных
8
Передача данных в приложение
Конец
Блокировка IP
9
Вывод предупреждения
Конец

Этап 1. Когда пользователь работает с приложением в браузере, то на веб-сервер передаются данные, которые в программе мы собираем из массивов PHP: $_GET, $_POST, $_COOKIE в общий массив и далее обрабатываем.

Этап 2. Категория пользователя определяется после аутентификации, состоящей в проверке соответствия пароля и адреса электронной почты, которые вводятся на странице входа в приложение. По таблице пользователей в базе данных определяем категорию пользователя. Если процедура аутентификации не пройдена, то пользователь считается незарегистрированным. Для веб-приложения мы используем несколько категорий пользователей:

1. Администратор
2. Временно зарегистрированный пользователь
3. Зарегистрированный пользователь
4. Незарегистрированный пользователь

Администратор, обладает правами, которые могут нанести вред веб-приложению при неправильных действиях. Остальные пользователи не обладают достаточными привилегиями, чтобы в результате злого умысла нанести вред системе, при условии соответствующего уровня защиты веб-приложения.

Этапы 3-4. В зависимости от категории пользователя выбирается таблица из базы данных с разрешёнными запросами GET, POST и допустимыми адресами. Здесь учитывается что не все пользователи имеют доступ ко всем страницам и не все страницы требуют передачи параметров. Доступ к страницам задан согласно логике приложения.

Этап 5. Все переменные переданные пользователем проверяются по сигнатурам, известным как правила, которые содержат допустимые данные. Система защиты включает в себя базу сигнатур и «срабатывает», когда входящие данные не совпадают с сигнатурами. Так как сигнатуры разработаны, чтобы максимально соответствовать логике приложения, то это приводит к небольшому уровню ошибочных срабатываний. Сигнатура выполнена на основе регулярного выражения [10], т.е. строке, состоящей из символов и мета символов и задающей правило поиска. Фильтрация на этом этапе защищает от большого количества атак, направленных на приложение.

Этап 6. Происходит регистрация событий, причём как успешных, та и неуспешных. Журнал с данными об успешных посещениях используется для оптимизации содержания страницы. Регистрация о неуспешных событиях необходима для последующего анализа администратором, который на основе анализа может корректировать работу системы защиты и веб-приложения и реагировать на возможные атаки. Информация обо всех событиях (рис.2) сохраняется в базе данных.

Этап 7. Анализ журнала позволяет выявить активность пользователя. Данный параметр вычисляется по числу обращений за один час с этого IP – адреса для незарегистрированного пользователя или по номеру пользователя в случае входа в систему. Расчёт ведётся в соответствии с таблицей 2: если сумма баллов превысила пороговый уровень (100 баллов), то переходим к блокировке пользователя на один час. В таблице 2 указано, как рассчитываются баллы для различных вариантов. Если адрес пользователя верный, мы тоже добавляем штрафные баллы, чтобы исключить влияние роботов (кроме поисковых), которые собирают информацию или ищут уязвимости. При «нормальном поведении» пользователя, в этом случае пороговой суммы здесь не достигается. Для пользователей, не имеющих злого умысла, имеется вероятность набрать некорректные данные, поэтому данная система баллов при нечастых ошибках позволяет пользователю работать с приложением прозрачно.

Таблица 2.

Начисление баллов за действия пользователя

Параметр	Адрес		GET / POST		COOKIE		неверный пароль
	верный	неверный	не подлежит передаче	неверный	не подлежит передаче	неверный	
Баллы	0,2	2	5	3	10	5	10

Этап 8. Данные, приводятся к корректному виду и выдаются для дальнейшей обработки веб-приложению. Время обработки запросов от пользователя по алгоритму (рис.3) по результатам опытной проверки не превышает 0,1 секунды.

Этап 9. IP – адрес или номер пользователя заносится в журнал в запрещённый список. Для не прошедших проверку пользователей доступ к приложению закрыт с выводом информации о возможности повторить попытку позднее. В течение часа пользователь будет блокироваться.

Выводы

Поставленные задачи выполнены, система защиты интегрирована в веб-приложение и опробована на практике. Обеспечение безопасности, реализованное рассмотренной системой, защищает от большинства возможных атак на уровне приложения. Защита и мониторинг операционной системы, веб-сервера, интерпретатора PHP и базы данных осуществляется другими методами.

Список литературы

1. The netfilter.org "iptables" project. Available at: http://www.netfilter.org/projects/iptables/index.html (accessed 08 January 2018)

2. Welcome to OWASP: the free and open software security community. Available at: https://www.owasp.org (accessed 08 January 2018)

3. Жуков Ю. В. Основы веб-хакинга: нападение и защита (+DVD) 2-е изд. – СПб.: Питер, 2012. – 208 с.

4. Козиол Дж., Личфилд Д., Эйтэл Д., Энли К. и др. Искусство взлома и защиты систем. – СПб.: Питер, 2006. – 416 с.

5. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. – СПб.: Питер, 2016. – 992 с.

6. Скембрей Дж., Шема М. Секреты хакеров. Безопасность Web – приложений – готовые решения.: Пер. с англ. – Изд. дом «Вильямс», 2003. – 384 с.

7. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов.: Пер. с англ. – СПб.: ООО «Альфа-книга», 2017. – 464 с.

8. Уязвимости. URL: https://www.securitylab.ru/vulnerability/ (дата обращения: 17.04.2008), свободный. – загл. с экрана. – яз. рус.

9. Фленов М.Е. PHP глазами хакера: 2-е изд., доп. и перераб. – СПб.: БХВ-Петербург, 2010. – 336 с.

10. Фридл Дж. Регулярные выражения, 3-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2008. – 608 с.