Давид А. Хотелов[1], Виктор Ю. Радыгин[2], Анастасия С. Меркушева,[1]
Иван К. Егоров[1], Алена Ю. Парушкина[1]
[1]Московский политехнический университет,
ул. Б.Семёновская, 38, г. Москва, 107023, Россия
e-mail: david.home@mail.ru, http://orcid.org/0000-0003-1050-0167
e-mail: merkushevanastasiya@gmail.com, http://orcid.org/0000-0002-3475-6434
e-mail: egorov.k@yandex.ru, http://orcid.org/0000-0002-1542-6990
e-mail: p1996@yandex.ru, http://orcid.org/0000-0003-1202-7422
[2]Национальный исследовательский ядерный университет «МИФИ»,
Каширское ш., 31, г. Москва, 115409, Россия
e-mail: vyradygin@mephi.ru, http://orcid.org/0000-0001-5999-524X

РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА БЕЗОПАСНОСТИ ДЛЯ КЛАСТЕРА ИНФОРМАЦИОННЫХ СИСТЕМ, БАЗИРУЮЩИХСЯ НА ПЛАТФОРМЕ RUBY ON RAILS

DOI: http://dx.doi.org/10.26583/bit.2018.3.09

Аннотация. В настоящее время информационная среда многих крупных организаций представляет собой кластер веб-ориентированных систем, базирующихся на платформе Ruby on Rails. Одной из важнейших задач обеспечения комплексной информационной безопасности подобных организаций является мониторинг состояния всех компонент кластера в реальном времени. Проведённый в данной работе обзор современных средств мониторинга показал, что на сегодняшний день, не смотря на наличие эффективных технологий контроля отдельных компонент веб-ориентированной информационной среды, не существует комплексных средств, обеспечивающих интегральную работу с СУБД, сервером приложений, веб-сервером, межсетевым экраном уровня приложения и системной и аппаратной составляющими. Таким образом, данная работа посвящена созданию на основе свободных средств системы комплексного мониторинга кластера веб-приложений, позволяющих обеспечить администратора механизмами оперативного обнаружение сбоев и потенциально опасных ситуаций. Проведённый анализ существующих средств разработки выявил, что наиболее оптимальной является архитектура на основе комбинации технологий Nginx, ModSecurity, puma, Ruby on Rails, PostgreSQL, Redis, Sidekiq. Веб-серверы Nginx и WAF ModSecurity обеспечивают первичную обработку запросов. Puma, PostgreSQL и Ruby on Rails используются для реализации ядра системы. Sidekiq и Redis применяются для создания системы отложенных задач. Веб-интерфейс созданной централизованной системы мониторинга предлагает разносторонние средства интегральной инфографики, позволяющие как следить за состоянием системы в текущий момент, так и изучать поведение всех компонент в прошлом. Разработанный программный комплекс успешно апробирован на ERP-системе НИЯУ МИФИ и показал себя как эффективное средство интегрального мониторинга кластера веб-систем, не требующее существенных финансовых затрат для внедрения.

Ключевые слова: система мониторинга, Ruby on Rails, кластер, gem-пакет, безопасность, загрузка сервера, Nginx, сбои кластера.
Для цитирования: ХОТЕЛОВ, Давид А. et al. РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА БЕЗОПАСНОСТИ ДЛЯ КЛАСТЕРА ИНФОРМАЦИОННЫХ СИСТЕМ, БАЗИРУЮЩИХСЯ НА ПЛАТФОРМЕ RUBY ON RAILS. Безопасность информационных технологий, [S.l.], n. 3, p. 88-100, 2018. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/1143>. Дата доступа: 28 aug. 2018. doi:http://dx.doi.org/10.26583/bit.2018.3.09.

David A. Khotelov[1], Victor Y. Radygin[2], Anastasia S. Merkusheva[1],
Ivan K. Egorov[1], Alyona Y. Parushkina[1]
[1]Moscow Polytechnic University,
Bolshaya Semenovskaya st., 38, Moscow, 107023, Russia
e-mail: david.home@mail.ru, http://orcid.org/0000-0003-1050-0167[3]
e-mail: merkushevanastasiya@gmail.com, http://orcid.org/0000-0002-3475-6434
e-mail: egorov.k@yandex.ru, http://orcid.org/0000-0002-1542-6990
e-mail: p1996@yandex.ru, http://orcid.org/0000-0003-1202-7422
[2]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),

Kashirskoye shosse, 31, Moscow, 115409, Russia
e-mail: vyradygin@mephi.ru, http://orcid.org/0000-0001-5999-524X

Development of the security monitoring system for cluster of information systems based on the Ruby on Rails framework

DOI: http://dx.doi.org/10.26583/bit.2018.3.09

Abstract. Currently, the information environment of many large organizations is represented by a cluster of web-oriented information systems. Often these information systems are based on the Ruby on Rails framework. One of the most important tasks of complex information security in such organizations is monitoring of all cluster components in real-time. The review of modern monitoring tools carried out in this work has shown two important points. There are many effective programs for monitoring the separate components of the web environment. But there is no one complex tool that supports interaction with DBMS, an application server, a web server, a web application firewall and system or hardware resources. Thus, this paper is devoted to the development of a complex monitoring system for a cluster of web applications. The created system is based on free software and can be used by administrators to operational detection of failures or potentially dangerous situations. Analysis of existed development technologies is carried out. Application architecture is based on a combination of the following tools: Nginx, ModSecurity, puma, Ruby on Rails, PostgreSQL, Redis, Sidekiq. The Nginx web server and WAF ModSecurity provide primary processing of requests. Puma, PostgreSQL and Ruby on Rails are used to create the application core. Sidekiq and Redis implement the mechanism of delayed jobs. The web interface of the developed centralized system provides various integrated infographic tools that allow the administrators to control current status of the system and investigate the states of all components in past. The created software was successfully tested on the NRNU MEPhI ERP-system. It has proved to be an effective tool for the complex monitoring of a cluster of web applications. Its implementation does not involve significant financial costs.

Keywords: monitoring system, Ruby on Rails, cluster, gem package, security, server load, Nginx, cluster failures.

For citation: KHOTELOV, David A. et al. Development of the security monitoring system for cluster of information systems based on the Ruby on Rails framework. IT Security (Russia), [S.l.], n. 3, p. 88-100, 2018. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/1143>. Date accessed: 28 aug. 2018. doi:http://dx.doi.org/10.26583/bit.2018.3.09.

Введение

На сегодняшний день одним из важнейших аспектов информатизации крупных организаций является задача обеспечения комплексной информационной безопасности. Данная задача является особо актуальной при использовании в бизнес-процессах предприятия открытых для внешнего доступа веб-ориентированных информационных систем. Она включает в себя целый ряд компонент и требует всестороннего подхода как в техническом, так и в организационном обеспечении. Причем наряду с такими важными составляющими процесса защиты, как разработка единой политики информационной безопасности, обучение сотрудников и повышение их квалификации, аттестация помещений и рабочих мест, обеспечение физической защиты и т.д., немаловажную роль играет вопрос мониторинга состояния всех компонент информационной среды в реальном времени.

Вопросам мониторинга задач обеспечения безопасности, не связанных непосредственно с самим веб-сервером и его программным и аппаратным обеспечением, посвящён целый ряд работ. Например, в работе Н.Г. Милославской [1] рассматривается вопрос мониторинга сети организации. Ю. Лиу в своей работе [2] показывает особенности вопроса мониторинга DNS-серверов. В работе В.С. Оладько [3] описан вопрос мониторинга остаточной информации на компьютерах конечных пользователей систем. В работе А.С. Зайцева [4] проводится комплексное исследования вопросов мониторинга безопасности на уровне целой отрасли – банковской сферы Российской Федерации.

Тем не менее при обеспечении безопасности кластера постоянно развивающихся веб-ориентированных информационных систем многие проблемы информационной защиты и даже самого функционирования комплекса связаны с большим числом

обновлений, постоянно разворачиваемых как на отдельных компонентах, так и на самой системе в целом. В последнем случае имеются в виду обновления операционной системы и критически важных библиотек. Проблемы могут возникать из-за программных ошибок в обновлениях, возникновения дефицита аппаратных и системных ресурсов из-за нового функционала, другого поведения системы в случае простейших и ранее безопасных атак злоумышленников на веб-сервер и работающие под его управлением приложения. Проблемы могут возникнуть также из-за поведения внешних невредоносных агентов. Например, из-за поисковых роботов Google или Yandex и изменения их поведения с течением времени.

Глубокая взаимосвязь обновлений программного обеспечения и вопросов информационной безопасности широко подтверждена в работах многих современных авторов. К примеру, на высокую сложность решения данного вопроса (применительно к обновлению расширений и вспомогательных библиотек) указывает Р.Г. Кула [5]. Более широко вопрос безопасности совместной разработки и обновления программного обеспечения раскрыт в работе Л. Мугарза [6]. Таким образом, вопрос обеспечения безопасности на данном уровне крайне важен и требует в том числе эффективных средств оперативного мониторинга.

Тем не менее на сегодняшний день средства мониторинга, направленные на оперативное обнаружение проблем на уровне веб-сервера и сервера приложений, сильно дифференцированы и проблемно-ориентированы. Как следствие, отсутствуют малозатратные интегральные средства, решающие данную комплексную задачу.

Анализ состояния средств мониторинга веб-ориентированных сред

При работе веб-ориентированной информационной системы можно выделить как минимум четыре составляющих, мониторинг за которыми может быть полезен с точки зрения анализа устойчивости и защищённости системы. Прежде всего, это аппаратные и системные ресурсы, которые включают: загруженность сетевых адаптеров и сетевой трафик, загруженность процессора, оперативной памяти, дисковых накопителей, количество используемых файловых дескрипторов, объемы используемых разделяемых ресурсов. Вторая составляющая – это веб-сервер и связанный с ним межсетевой экран уровня приложений (WAF), их загруженность с точки зрения обращений, количества ошибок, количества отклонённых запросов и т.д. Третья и четвертая составляющие – это сервер приложения и сервер базы данных.

Для каждой из рассмотренных четырёх составляющих на сегодняшний день можно найти эффективные средства мониторинга. Например, К. Чу в своей работе [7] рассматривает вопрос построения системы серверного мониторинга на основе технологии Nagios. Аналогичную идею использует в своей работе Р. Хан [8]. Средства мониторинга веб-серверов также широко востребованы в современных системах безопасности и технического контроля. В частности, для сервера Nginx широко используется технология Amplify [9]. Для сервера Apache доступна технология Anturis [10]. Средства мониторинга СУБД также широко распространены. Например, для СУБД PostgreSQL существует целая плеяда средств мониторинга, в том числе Mamonsu [11], Zabbix [12]. Для средств мониторинга данных аудита межсетевых экранов уровня приложения также существует целый ряд средств. Например, для AWS WAF [13] от компании Amazon предусмотрен целый набор средств мониторинга, таких как Amazon CloudWatch Alarms, AWS CloudTrail Log Monitoring и т.д. С другой стороны, для решений с открытым кодом в данной области, таких как ModSecurity [14], Naxsi [15] и других, качественных средств мониторинга на сегодняшний день нет.

Наибольшую сложность с точки зрения выбора средств мониторинга вносит сервер приложения, так как в общем случае он представляет собой произвольное программное обеспечение, функционирующее по собственным алгоритмам и принципам. С другой стороны, большинство современных веб-ориентированных информационных систем базируются на определённых платформах веб-разработки, гарантирующих единую

концепцию функционирования основных компонент приложения. Таким образом, при выборе средства мониторинга сервера приложений можно ориентироваться на системы мониторинга данных платформ, а не разрабатывать уникальные вспомогательные средства.

На сегодняшний день одним из наиболее популярных средств для разработки веб-ориентированных информационных систем является платформа веб-разработки Ruby on Rails. По данным австралийской компании BuiltWith Pty Ltd [16], доля рынка веб-приложений среди сайтов с количеством посетителей свыше десяти тысяч, соответствующая данной технологии, составляет порядка 11 %. Применение платформы Ruby on Rails можно найти во многих областях деятельности человека. Например, в работе авторов [17] раскрывается её применение в сфере образования. В работе Л. Хуанг [18] показано применение данной платформы веб-разработки для задач медицины. При этом, есть большое число работ, посвящённых вопросам безопасности технологии Ruby on Rails и базирующихся на её основе программных решений. Например, в работе Е.А. Роганова [19] раскрывается вопрос разработки единой системы аутентификации для Ruby on Rails-приложений.

Средства мониторинга сервера приложений для платформы веб-разработки Ruby on Rails также широко распространены. Наиболее популярной из них является система Errbit [20], построенная на основе технологии Airbrake [21]. Данная технология позволяет отслеживать в том числе и многие аспекты, связанные с работой СУБД. Тем не менее комплексного решения, обеспечивающего единый мониторинг аппаратных и системных ресурсов, веб-сервера, СУБД и сервера приложений для систем, базирующихся на технологии Ruby on Rails (как и на любых других популярных платформах веб-разработки), на сегодняшний день нет. Более того, набор средств, обеспечивающий все потребности комплексного мониторинга, в большинстве случаев требует наличия одной или двух проприетарных компонент, что делает его достаточно дорогостоящим средством. При этом именно комплексный мониторинг зачастую позволяет выявить критические компоненты приложения или замаскированные атаки злоумышленников. Таким образом, на сегодняшний день задача создания недорогого интегрального средства мониторинга среды веб-приложения, базирующихся на основе платформы веб-разработки Ruby on Rails, является актуальной. В связи с этим, коллективом авторов был разработан программный комплекс для универсального мониторинга Ruby on Rails-систем, позволяющий в единой среде контролировать состояние всех основные компонент веб-ориентированного приложения.

Архитектура системы

При разработке архитектуры системы интегрального мониторинга были выдвинуты три основных требования: система должна базироваться только на бесплатном программном обеспечении, система должна быть централизованной, система не должна замедлять работу основных информационных систем кластера.

Учитывая тот факт, что разрабатываемая система предназначена для мониторинга кластера веб-приложений, базирующихся на платформе Ruby on Rails, в качестве технологической основы для её создания также была выбрана данная платформа веб-разработки. Подобный выбор позволил рассматривать систему мониторинга как ещё одну единицу кластера, и дал возможность избежать дополнительных требований к прикладному программному обеспечению и квалификации системных администраторов.

В качестве возможной СУБД для построения архитектуры системы были рассмотрены три наиболее популярных продукта со свободной или не требующей оплаты лицензией, а именно PostgreSQL, MySQL и Oracle. При анализе преимуществ и недостатков указанных СУБД дополнительно учитывался фактор качества драйвера, обеспечивающего функционирование объектно-реляционного преобразователя ActiveRecord (является компонентой платформу веб-разработки Ruby on Rails). Драйвер СУБД Oracle, в отличие от драйверов СУБД PostgreSQL и MySQL, не позволяет в полной

мере использовать все особенности технологии ActiveRecord, что делает использование данного продукта для разработки системы нецелесообразным.

На сегодняшний день стабильный релиз MySQL, в отличие от PostgreSQL, не включает поддержку многопоточности [22]. Данная проблема решается дополнительными продуктами, такими как MySQL Cluster, ProxySQL и т.д. Тем не менее данный факт является недостатком MySQL по сравнению с PostgreSQL в рамках поставленной задачи. Кроме того, учитывая тот факт, что часть информации, необходимой для выполнения отложенного мониторинга, представляет собой данные в формате JSON, к преимуществу PostgreSQL можно отнести встроенную поддержку бинарного типа данных JSONB. Рассмотренные причины обуславливают выбор СУБД PostgreSQL в качестве основной реляционной базы данных разрабатываемой системы мониторинга.

Для обеспечения эффективного обмена данными между зависимыми системами и системой мониторинга был осуществлён выбор механизмов организации отложенных задач и связанной с ним документоориентированной СУБД. Для платформы Ruby on Rails существует две основные технологии организации отложенных задача: Sidekiq и DelayedJob. Наиболее полные функциональные возможности предоставляет Sidekiq, что обуславливает его выбор для разработки системы мониторинга. Для организации хранилища отложенных задач в оперативной памяти были рассмотрены три наиболее популярных NoSQL СУБД: MongoDB, Redis, Berkeley DB. По совокупности параметров производительность/простота в использовании была выбрана документноориентированная СУБД Redis.

Отдельной задачей при выборе технологий для построения системы мониторинга является выбор механизмов автоматического или полуавтоматического разбора журналов веб-сервера и межсетевого экрана уровня приложения. Прежде всего было проведено исследование наиболее востребованных технологий, используемых в нашей стране при разработке информационных систем на платформе Ruby on Rails. В связи с отсутствием качественной статистики по данному вопросу в открытом доступе, исследование проводилось в формате опроса специалистов различных компаний, занимающихся созданием и поддержкой веб-приложений. По результатам исследования, превалирующим набором технологий является комбинация веб-сервера Nginx и экрана ModSecurity. Таким образом, система мониторинга преимущественно ориентирована на анализ журналов, создаваемых данными программными средствами. Наиболее качественным свободно распространяемым средством разбора журналов Nginx на сегодняшний день является транслятор Nginx GoAccess [23]. Средств разбора журнала экрана ModSecurity на сегодняшний день крайне мало. Фактически необходимой функциональностью в той или иной мере обладает только пакет ModSecurity log parser. Именно данный продукт и был выбран в качестве средства полуавтоматического разбора журнала межсетевого экрана уровня приложения.

Общий вид архитектуры системы мониторинга показан на рис. 1. Непосредственно сама система мониторинга обозначена как «Главное приложение», а пример отдельной информационной системы кластера обозначен как «Приложение из кластера». Сбор данных происходит параллельно на трёх уровнях.

Прежде всего на первом уровне за счёт переопределения встроенной системы логирования осуществляется мониторинг сервера обработки Ruby on Rails приложений puma. В дополнение к стандартному режиму записи журнала в набор текстовых файлов происходит формирование JSON представления каждого события и его размещение в очереди готовых событий системы отложенных задач Sidekiq.

На втором уровне выполняется мониторинг веб-сервера nginx. Он осуществляется специальным daemon-приложением, развёрнутым на стороне каждого из приложений кластера. Daemon-приложение осуществляет с заданной частотой (по умолчанию один раз в минуту) преобразование новых событий в текстовом журнале веб-сервера в JSON-пакет. Затем JSON-пакет, в свою очередь, размещается в общей (для всех трёх уровней) очереди готовых событий системы отложенных задач Sidekiq.

Мониторинг аппаратных и системных ресурсов осуществляется при помощи gem-пакета server-metrics. Сбор данных осуществляется тем же daemon-приложением, что и на втором уровне. Формируемый с определённой временной частой JSON-пакет включает в себя данные: о входящем и исходящем трафиках каждого из сетевых интерфейсов, о загруженности процессоров, о свободном месте, на разделах жёстких дисков, об объёмах свободной и задействованной оперативной памяти, доступном и используемом количестве файловых дескрипторов. В общем случае набор отслеживаемых параметров может быть легко расширен. Созданный JSON-пакет размещается в общей очереди готовых событий системы отложенных задач Sidekiq.

Непосредственная отправка сформированных в Sidekiq JSON-пакетов событий осуществляется самим сервером приложений путём отправки запроса с подтверждением к веб-серверу главного приложения. Для обеспечения бесперебойной работы в условиях непостоянного соединения Sidekiq кэшируется в оперативной памяти посредством СУБД Redis. Если отправка не выполняется по каким-либо причинам, то задача сохраняется в Redis и ждёт восстановления соединения по сети для повторной передачи.

Система мониторинга реализована в виде клиентского и серверного программных приложений. Клиентское приложение создано в виде gem-пакета «hot_catch», который необходимо включить в состав Ruby on Rails приложений информационных систем кластера.

Серверное приложение включает в себя систему сбора и обработки данных от информационных систем кластера и интерфейс администратора, позволяющий осуществлять мониторинг. Обработка данных также является многоуровневой, где каждый уровень главного приложения принимает данные от уровня приложения из кластера.

На первом уровне данные веб-сервера nginx преобразуются средствами GoAccess в статистические показатели, записанные в формате JSON.

Второй и третий уровни похожи. Здесь аппаратные данные и журнал Ruby on Rails обрабатываются и аккумулируются в СУБД PostgreSQL.

Учитывая тот факт, что серверное приложение реализовано в виде веб-сервиса, базирующегося на платформе Ruby on Rails, то сама система мониторинга может быть включена в список информационных систем, отслеживаемых в кластере, что в свою очередь позволяет избежать вызванных ею технических сбоев.

Реализация интерфейсной части

Серверное приложение состоит из двух основных компонент: API для взаимодействия с клиентскими приложениями и аккумуляции полученных от них данных о событиях в информационных системах кластера и интерфейсной части, обеспечивающей администраторов кластеров и/или администраторов отдельных информационных систем кластера средствами интегрального мониторинга. Обе компоненты реализованы на основе MVC-проектирования и базируются на платформе веб-разработки Ruby on Rails. Единая база данных под управлением СУБД PostgreSQL обеспечивает как хранение массивов данных мониторинга, так и хранение локальных компонент системы, таких как, например, распределение пользовательских прав. С целью повышения производительности системы и обеспечения её безопасности от внешних атак прямой доступ к серверу приложений не осуществляется.

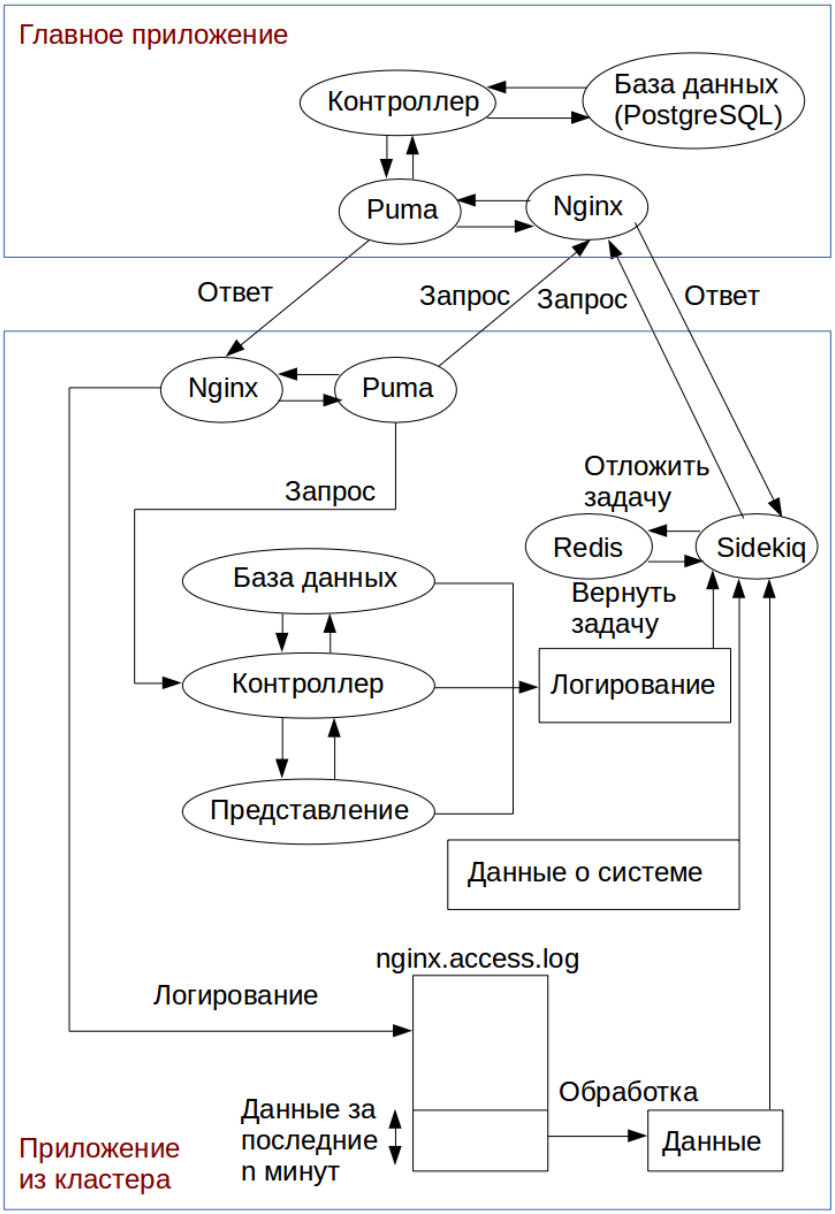

Рис. 1. Архитектура системы мониторинга
(Fig. 1. The architecture of the monitoring system)

Внешние обращения обрабатываются веб-сервером Nginx, который проксирует посредством Unix-сокетов сервер Ruby on Rails приложений puma. Повышение производительности осуществляется за счёт выполнения на уровне Nginx запросов на статичные материалы (шрифты, таблицы стилей, статичные HTML-страницы, изображения и т.д.). Три уровня мультизадачности (несколько процессов у Nginx, несколько процессов у Puma, несколько процессов у СУБД PostgreSQL) обеспечивают эффективность системы при большом числе одновременных запросов. Повышение безопасности достигается за счёт применения на уровне Nginx межсетевого экрана уровня приложения ModSecurity, а также использования SSL-шифрования. Решение данных задач непосредственно на уровне сервера приложений puma крайне затруднительно.

Реализация API выполнена в соответствии с современными стандартами обеспечения безопасности передачи веб-данных [24].

Интерфейсная часть построена таким образом, чтобы пользователь получал набор стандартных средств мониторинга, дополненный интегральными показателями и инфографикой. В частности, администратору доступна возможность просмотра событий

Ruby on Rails-приложения и связанной с ним СУБД (включая ошибки, предупреждения и информационные сообщения) в формате, характерном для приложения errbit (рис. 2).

Администратору также доступен интерфейс анализа журналов веб-сервера Nginx. При этом функциональные возможности данного интерфейса выполнены в соответствии с наиболее популярными средствами, применяемыми в данной области. Аналогичным образом интерфейс анализа аппаратных и системных ресурсов в случае необходимости доступен в виде отдельной компоненты, стиль подачи информации в который выполнен в соответствии с популярными стандартами.

Основным преимуществом разработанной системы мониторинга является возможность просмотра интегральных показателей сразу по всем компонентам. При этом администратору доступна возможность самостоятельно включать интересующие его составляющие. Например, просмотр событий журнала веб-сервера Nginx и событий Ruby on Rails приложения по информационной системе 1 и просмотр журнала событий веб-сервера Nginx информационной системы 2.

Особым преимуществом является наличие формируемой онлайн инфографики. Реализация инфографики выполнена с использованием библиотеки C3 [25], основанной на JavaScript и SVG, и не требует установки каких-либо дополнительных плагинов на стороне браузера клиента. Инфографика включает широкий спектр диаграмм и графиков с большим количеством настраиваемых параметров. Например, на рис. 3 показан пример отслеживания нагрузки на веб-сервер одной или нескольких информационных систем кластера (выбираются при помощи IP - адреса, привязанного к системе) по всем или указанным (выбираются также по IP-адресу, с которого идёт запрос) пользователям. Оператору также доступна настройка границ анализируемого временного отрезка и шага построения гистограммы.

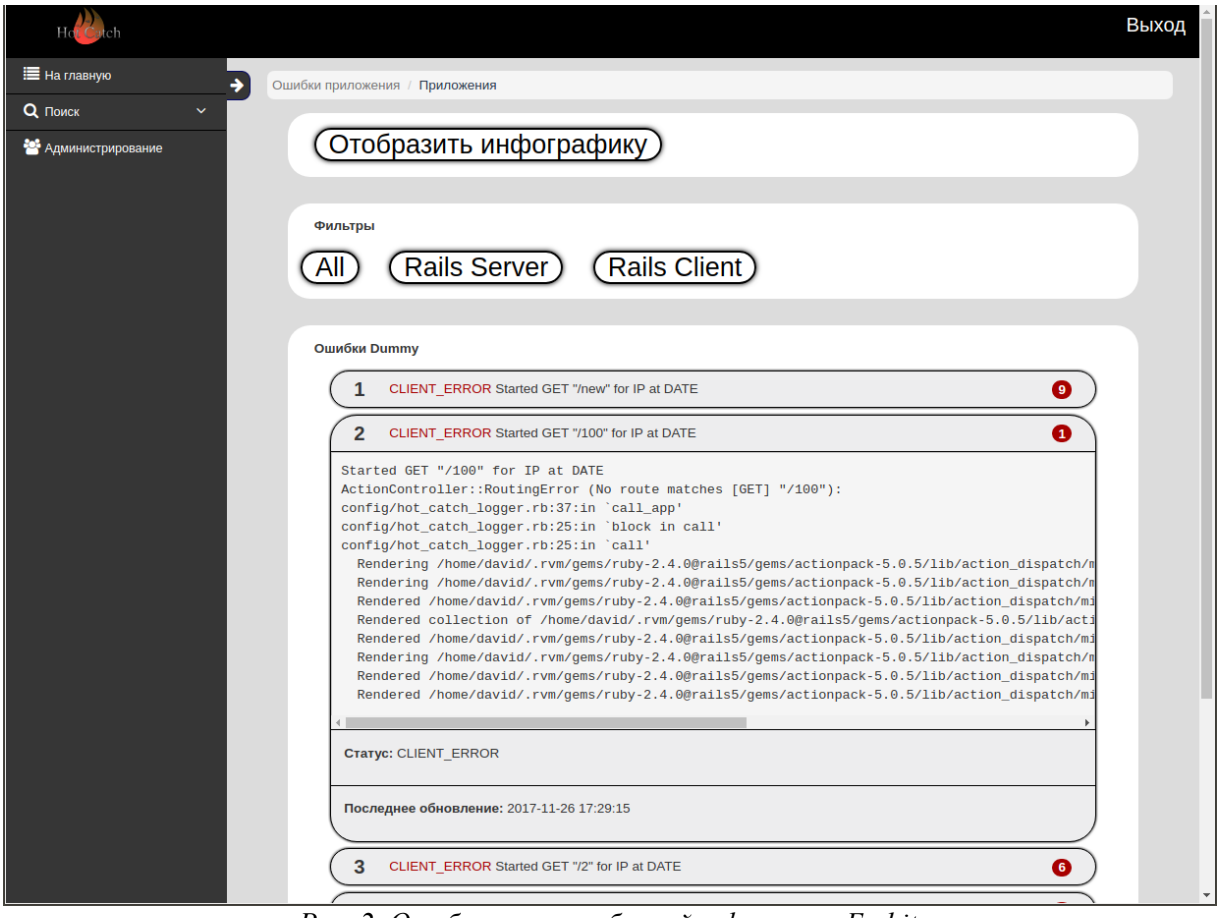

Рис. 2. Отображение событий в формате Errbit
(Fig. 2. Representation of events in Errbit format)

Встроенная в серверное приложение гибкая ролевая система позволяет использовать систему мониторинга как главным администратором кластера, так и администраторами или разработчиками его отдельных составляющих (информационных систем). Интерфейсная часть поддерживает различные варианты организации аутентификации, включая аутентификацию посредством CAS-протокола [19].

Апробация системы мониторинга

Апробация системы мониторинга осуществлялась на базе кластера приложений НИЯУ МИФИ и ряда его филиалов. В частности, были подключены 12 информационных систем Московской площадки НИЯУ МИФИ, в том числе: 7 систем, доступных из сети «Интернет», и 5 систем, доступных только из локальной сети университета, а также одна общая для всех площадок информационная система и одна информационная система в Северском филиале.

Тестирование производительности системы осуществлялось за счёт искусственной генерации большого числа наиболее популярных запросов к основным информационным системам. В частности, использовались пять информационных систем: «Корпоративный портал НИЯУ МИФИ», «ВКР НИЯУ МИФИ», «Контингент обучающихся», «Контингент сотрудников», «Расписание занятий». К данным системам одновременно осуществлялось порядка 2000 обращений в минуту, в том числе 1000 обращений, связанных с ошибочными запросами и запросами, эмулирующими различные атаки. Среднее время выполнения одного обращения к информационным системам кластера, до подключения системы мониторинга на пиковой нагрузке составляло 370 мс. С включённой системой мониторинга изменение среднего времени запросов составило всего 0,5 % (порядка 1-2 мс). Было получено, что при условии эквивалентности аппаратного обеспечения и числа выделенных параллельных потоков информационных систем кластера и сервера системы мониторинга, то с ростом числа запросов потеря времени не изменяется. Исключение составляют ситуации, когда запросы к информационным системам по скорости выполнения превышают время взаимодействия с системой мониторинга, что характерно только для большого числа обращений, отклонённых на уровне сервера приложений (например, запросов, завершившихся ошибкой 404 «Страница не найдена»). Тем не менее даже в такой ситуации потеря производительности составляет только двукратный прирост по времени выполнения запроса.

Система мониторинга показала высокую эффективность с точки зрения нагрузки и позволила эффективно исследовать вопрос стабильности работы таких информационных систем, как «Корпоративный портал НИЯУ МИФИ» и «Контингент обучающихся». В частности, с помощью системы мониторинга были выявлены причины периодических сбоев сервера приложений puma, связанные с моментами пиковой нагрузки по ряду параметров в ночное время суток. Кроме того, с помощью системы мониторинга и формируемых с её помощью наборов данных была выполнена классификация всех видов атак, применяемых к информационным системам НИЯУ МИФИ в течение периода с октября 2017 по март 2018 года, что позволило предусмотреть оптимизации методов защиты от них и минимизировать воздействие таких атак на производительность информационных систем.

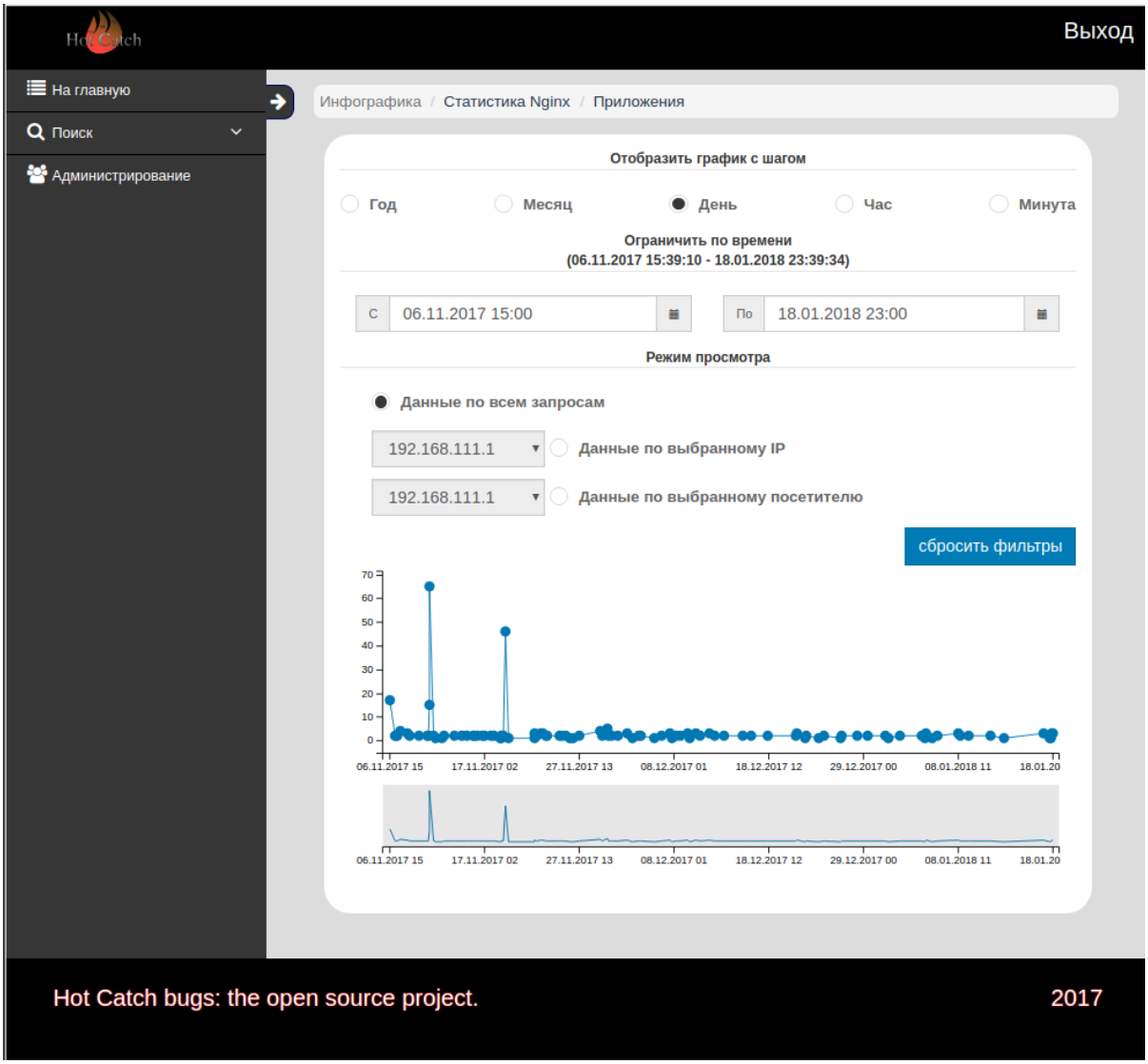

Рис. 3. Пример отображения простейшей инфографики
(Fig. 3 An example of a simple infographic interface)

Заключение

Разработанная система мониторинга безопасности для кластера информационных систем избавляет администратора от необходимости устанавливать, осваивать и использовать целый ряд средств мониторинга различных компонент жизненного цикла веб-приложений, базирующихся на платформе Ruby on Rails. Возможности решения интегральных задач мониторинга в одной среде, предоставляемые системой, снимают с администратора дополнительную нагрузку, связанную со сведением в единой целое данных из различных программных продуктов, например из Errbit и Zabbix. Широкие средства инфографики позволяют быстро и оперативно обнаружить технические сбои, узкие места и возможные атаки. При этом единый универсальный формат представления информации о состоянии всех систем и удобное API позволяют, в случае необходимости, администратору дополнять систему собственным средствами анализа или подключать интеллектуальные системы безопасности.

Особыми преимуществами системы являются её свободное распространение (что снижает затраты на её внедрение) и низкие требования к дополнительно устанавливаемому программному обеспечению. Адаптивный веб-интерфейс позволяет решать задачи мониторинга с использованием любых мобильных устройств.

Прототип системы доступен для свободного скачивания на сайте github. Выпуск стабильного релиза предполагается в июле - августе 2018 года.

Учитывая тот факт, что большинство технологий, используемых для взаимодействия системы мониторинга и информационных систем кластера универсально, данный подход может быть применён и для построения систем мониторинга для кластеров, базирующихся на других платформах веб-разработки.

Дальнейшее развитие разработанной системы мониторинга веб-приложений, базирующихся на платформе Ruby on Rails, подразумевает расширение возможностей мониторинга СУБД, включение возможностей мониторинга ресурсов System V, а также разработка модуля автоматического обнаружения потенциальных угроз и ошибок.

СПИСОК ЛИТЕРАТУРЫ:

1. Н. Г. Милославская, А. И. Толстой. Мониторинг сетевой безопасности с применением коммутаторов. // Безопасность информационных технологий – Том 20, № 3, 2013. – С. 65-67. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/320>. Дата доступа: 11 мая 2018.
2. Ю. Лиу, Т. Жи, З. Лиу. DNSAM: Система мониторинга и анализа в режиме реального времени данных DNS-серверов. // Международный журнал инновационных компьютерных технологий и управления – Том 13, Выпуск 4, 2017. – С. 1425-1432.
3. В.С. Оладько. Мониторинг и аудит остаточной информации на компьютере пользователя // Безопасность информационных технологий – Том 22, № 2, 2015. – С. 85-91. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/120>. Дата доступа: 11 мая 2018.
4. А.С. Зайцев. Исследование процесса мониторинга информационной безопасности в организации банковской системы Российской Федерации // Безопасность информационных технологий – Том 20, № 3, 2013. – С. 78-82. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/323>. Дата доступа: 11 мая 2018.
5. Р.Г. Кулэ, Д.М. Герман, А. Оуни, Т. Ишио, К. Иное. Обновляют ли разработчики связанные библиотеки? Эмпирическое исследование важности обновления связанных библиотек с точки зрения информационной безопасности // Эмпирическая разработка программного обеспечения – Том 23, Выпуск 1, 2018. – С. 384-417. doi:http://dx.doi.org/10.1007/s10664-017-9521-5.
6. Л. Мугарза, Д. Парра, Е. Якоб. Безопасное обновление программного обеспечения и защищённая совместная разработка // Лекции по информатики (включая лекции по искусственному интеллекту и биоинформатике) – Том 10489, 2017. – С. 199-210. doi:http://dx.doi.org/10.1007/978-3-319-66284-8_17.
7. К. Жу, Л. Мяо Хадуп система мониторинга реального времени, основанная на Ganglia, Nagios и mongoDB // Энергетика и прикладные технологии - Материалы 2-й Международной конференции по энергетике и прикладной технике, ESAT 2015. 2015. С. 483-488.
8. Р. Хан, С.Ю. Хан. Разработка и внедрение автоматического мониторинга сети и отчетности системы // Журнал интеграции промышленной информации. – 2018. – С. 24-34.
9. Система мониторинга Веб-сервера Nginx Amplify [Электронный ресурс] Режим доступа: https://www.nginx.com/products/nginx-amplify/, свободный (дата обращения 23.04.2018).
10. Система мониторинга Веб-сервера Apache Anturis [Электронный ресурс] Режим доступа: https://anturis.com/apache-monitoring/, свободный (дата обращения 23.04.2018).
11. Система мониторинга СУБД PostgreSQL Mamonsu [Электронный ресурс] Режим доступа: https://github.com/postgrespro/mamonsu, свободный (дата обращения 23.04.2018).
12. Система мониторинга Zabbix [Электронный ресурс] Режим доступа: https://www.zabbix.com/rn/rn3.4.7, свободный (дата обращения 23.04.2018).
13. Межсетевой экран уровня приложений AWS WAF [Электронный ресурс] Режим доступа: https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html, свободный (дата обращения 23.04.2018).
14. Межсетевой экран уровня приложений ModSecurity [Электронный ресурс] Режим доступа: https://modsecurity.org/, свободный (дата обращения 23.04.2018).
15. Межсетевой экран уровня приложений Naxsi [Электронный ресурс] Режим доступа https://github.com/nbs-system/naxsi, свободный (дата обращения 23.04.2018).
16. Статистика использования различных платформ веб-разработки в сети «Интернет» по информации компании BuiltWith Pty Ltd [Электронный ресурс] Режим доступа: http://trends.builtwith.com/framework, свободный (дата обращения 23.04.2018).
17. В.Ю. Радыгин, Н.В. Лукьянова, Д.Ю. Куприянов. LMS-система в университете для студентов очного обучения: синергия бесплатного программного обеспечения, конкурентного подхода и технологии социальных сетей // Материалы Международной научно-практической конференции по информационным технологиям в образовании XXI века, ITE-XXI 2015. - Труды конференции AIP, 2017, Vol. 1797 – № 020015. doi:http://dx.doi.org/10.1063/1.4972435
18. Л. Хуан, Х. Фернандес, Х. Зиа, П. Тавасоли, Х. Реннет, Д. Писапи, М. Имельянски, А. Сбонер, М. Рубин, М. Клюк, О. Элементно. Точность знаний о раке, основанных на структурированных медицинских классах мутаций и нтерпретаций // Журнал Американской ассоциации медицинской информатики. – 2017. Vol. 24, Is. 3. – С. 513-519. doi:http://dx.doi.org/10.1093/jamia/ocw148

19. А.И. Александров, Е.А. Роганов. Модификация CAS-протокола для повышения уровня защиты веб-приложений от несанкционированного доступа // Безопасность информационных технологий – Том 24, № 3, 2017. – С. 43-49. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/263>. Дата доступа: 11 мая 2018. doi:http://dx.doi.org/10.26583/bit.2017.3.05.
20. Система отслеживания ошибок Errbit [Электронный ресурс] Режим доступа: https://errbit.com/, свободный (дата обращения 23.04.2018).
21. Система мониторинга Airbrake [Электронный ресурс] Режим доступа: https://airbrake.io/, свободный (дата обращения 23.04.2018).
22. Д. Морело. Сравнение PostgreSQL и MySQL по состоянию на 2018 год [Электронный ресурс] Режим доступа: https://linuxhint.com/postgresql-vs-mysql-2018/, свободный (дата обращения 23.04.2018).
23. Система анализатора журналов Nginx GoAccess [Электронный ресурс] Режим доступа: https://goaccess.io/, свободный (дата обращения 23.04.2018).
24. А. А. Хейн, Б. А. Щукин. Обеспечение информационной безопасности при взаимодействии с веб-сервисом // Безопасность информационных технологий – Том 19, № 1, 2012. – С. 124-127. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/567>. Дата доступа: 11 мая 2018.
25. JavaScript библиотека построения графиков C3 [Электронный ресурс] Режим доступа: http://c3js.org/, свободный (дата обращения 23.04.2018).

REFERENCES:

[1] N.G. Miloslavskaya, A.I. Tolstoy Network Security Monitoring on the basis of Switches. IT Security (Russia). – Vol. 20, N. 3, 2013 – pp. 65-71. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/320>. Date accessed: 11 may 2018. (in Russian).
[2] Y. Liu, T. Zhi, Z. Liu DNSAM: A DNS data real-time analysis and monitoring system. International Journal of Innovative Computing, Information and Control – Vol. 13, Iss. 4, 2017. – pp. 1425-1432.
[3] V.S. Oladk Monitoring and Auditing Residual Information on the User's Computer. IT Security (Russia), – Vol. 22, N. 2, 2015. – pp. 85-91. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/120>. Date accessed: 11 may 2018. (in Russian).
[4] A.S. Zaytsev Information Security Monitoring Process Research in Russian Federation Banking System Organization. IT Security (Russia), – Vol. 20, N. 3, 2013 – pp. 78-82. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/323>. Date accessed: 11 may 2018. (in Russian).
[5] R.G. Kula, D.M. German, A. Ouni, T. Ishio, K. Inoue Do developers update their library dependencies?: An empirical study on the impact of security advisories on library migration. Empirical Software Engineering – Vol. 23, Is. 1, 2018. – pp. 384-417. doi:http://dx.doi.org/10.1007/s10664-017-9521-5.
[6] L. Mugarza, J. Parra, E. Jacob Software updates in safety and security co-engineering. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) – Vol. 10489, 2017. – pp. 199-210. doi:http://dx.doi.org/10.1007/978-3-319-66284-8_17.
[7] Q. Zhu, L. Miao Hadoop real-time monitoring system based on Ganglia, Nagios and mongoDB. Energy Science and Applied Technology. Proceedings of the 2nd International Conference on Energy Science and Applied Technology, ESAT 2015. 2015. – pp. 483-488.
[8] R. Khan, S.U. Khan Design and implementation of an automated network monitoring and reporting back system. Journal of Industrial Information Integration. – 2018. – pp. 24-34. doi:http://dx.doi.org/10.1016/j.jii.2017.11.001.
[9] Monitoring system of the web server Nginx Amplify https://www.nginx.com/products/nginx-amplify/ (access date: 23.04.2018).
[10] Monitoring system of the web server Apache Anturis https://anturis.com/apache-monitoring/ (access date: 23.04.2018).
[11] Monitoring system of the DBMS PostgreSQL Mamonsu https://github.com/postgrespro/mamonsu (access date: 23.04.2018).
[12] Monitoring system Zabbix https://www.zabbix.com/rn/rn3.4.7 (access date: 23.04.2018). (in Russian).
[13] Web application firewall AWS WAF https://docs.aws.amazon.com/waf/latest/developerguide/waf-chapter.html (access date: 23.04.2018).
[14] Web application firewall ModSecurity https://modsecurity.org/ (access date: 23.04.2018). (in Russian).
[15] Web application firewall Naxsi https://github.com/nbs-system/naxsi (access date: 23.04.2018).
[16] Statistics on use of various frameworks technologies in the internet according information company BuiltWith Pty Ltd http://trends.builtwith.com/framework (access date: 23.04.2018).
[17] V.Y. Radygin, N.V. Lukyanova, D.Yu. Kupriyanov LMS in university for in-class education: Synergy of free software, competitive approach and social networks technology. Proceedings of International Scientific-Practical Conference on Information Technologies in Education of the XXI Century, ITE-XXI 2015. – AIP Conference Proceedings, 2017, Vol. 1797 – article number 020015. doi:http://dx.doi.org/10.1063/1.4972435.
[18] L. Huang, H. Fernandes, H. Zia, P. Tavassoli, H. Rennert, D. Pisapia, M. Imielinski, A. Sboner, M.A. Rubin, M. Kluk, O. Elemento The cancer precision medicine knowledge base for structured clinical-grade mutations and interpretations. Journal of the American Medical Informatics Association. –Vol. 24, Is. 3, 2017. – pp. 513-519. doi:http://dx.doi.org/10.1093/jamia/ocw148.

[19] A.I. Alexandrov; E.A. Roganov Modification of CAS-protocol for improvement of security web-applications from unauthorized access. IT Security (Russia), – Vol. 24, N. 3, 2017. –pp. 43-49. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/263>. Date accessed: 11 may 2018. doi:http://dx.doi.org/10.26583/bit.2017.3.05. (in Russian).
[20] Errbit system https://errbit.com/ (access date: 23.04.2018).
[21] Monitoring system Airbrake https://airbrake.io/ (access date: 23.04.2018).
[22] D.Morelo PostgreSQL vs MySQL 2018 https://linuxhint.com/postgresql-vs-mysql-2018/ (access date: 23.04.2018).
[23] Analyzer of system journals Nginx GoAccess https://goaccess.io/ (access date: 23.04.2018).
[24] A.A. Hein, B.A. Shchukin Providing Information Security when Interaction with Web Service. IT Security (Russia), – Vol. 19, N. 1, 2013. – pp. 124-127. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/567>. Date accessed: 11 may 2018. (in Russian).
[25] JavaScript library for creating charts C3 http://c3js.org/ (access date: 23.04.2018).

Поступила в редакцию – 07 мая 2018 г. Окончательный вариант – 23 августа 2018 г.
Received – May 07, 2018. The final version – August 23, 2018.