9. *Максимова Е.А., Корнева В.А.* Оптимизация технологии безопасного информационного взаимодействия в корпоративных системах // Материалы XII Международной научно-практической конференции «ИБ-2012». Ч. II. – Таганрог: Изд-во ТТИ ЮФУ, 2012. – С. 124-129.

Статью рекомендовал к опубликованию д.т.н., профессор О.Б. Макаревич.

Никишова Арина Валерьевна – Волгоградский государственный университет; e-mail: arinanv@mail.ru; 400062, г. Волгоград, пр. Университетский, 100; тел.: 88442460368; кафедра информационной безопасности; старший преподаватель.

Чурилина Александра Евгеньевна – e-mail: sashune4ka-ch@mail.ru; кафедра информационной безопасности; ассистент.

Nikishova Arina Valerievna – Volgograd State University; e-mail: arinanv@mail.ru; 100, Universitetsky pr., Volgograd, Russia, 400062; phone: +78442460368; the department of informational security; senior lecturer.

Churilina Aleksandra Eugenievna – e-mail: sashune4ka-ch@mail.ru; the department of informational security; assistant professor.

УДК 004.052.2

А.М. Максимов

АНАЛИЗ ОСОБЕННОСТЕЙ ОСУЩЕСТВЛЕНИЯ АТАК НА ВЕБ-СЕРВЕР ПОСРЕДСТВОМ ГЕНЕРАЦИИ ОШИБОЧНЫХ ЗАПРОСОВ

Статья посвящена рассмотрению особенностей функционирования веб-сервера Microsoft IIS. Рассматриваются условия, когда результатом на запросы к веб-ресурсу (умышленные или неумышленные) вместо действительной информации происходит выдача кодов ошибок (например, ошибки класса 4xx, которые являются кодами состояния HTTP). В результате таких запросов может происходить повышение потребления ресурсов веб-сервером, что при достижении некоторых установленных предельных лимитах может стать причиной отказа в обслуживании веб-сервером, или его сильного замедления обработки новых поступающих запросов.

Помимо описания проблемы в статье приводятся некоторые данные проведённых тестов, условия, в которых были получены результаты. Также указываются параметры, которые необходимо учитывать при вынесении прогнозов и оценок конечного результата. По результатам рассмотрения выявленных данных предлагаются некоторые методы, которые могут помочь сгладить негативные результаты запросов, вызывающих выдачу ошибок вместо действительного результата. Не оставлены без рассмотрения и негативные аспекты, которые могут быть следствием предлагаемых методов, и которым следует уделить внимание для того чтобы они не оказали существенного отрицательного влияния на конечный результат.

Веб-сервер; код состояния HTTP; нагрузка; отказ в обслуживании; IIS; HTTP.

A.M. Maximov

ANALYSIS OF ATTACK FEATURES TO WEB SERVER THROUGH REQUESTS WITH ERRORS

The article contains review of functioning features of Microsoft IIS web server under conditions when errors occur (error class 4xx, which are HTTP status codes in fact, for example) as result of requests (intentional or unintentional) instead of actual information. Increased resource usage may occur as result of these requests with errors, so if preset limits are reached, it can be a cause of web server denial of service or significant delays in processing of requests of all type (even actual, without evil intent).

Besides of description, article contains some test results and conditions for obtaining of these results. Also specifies parameters that must be consider while predicting and evaluating final results and offers methods which can help to smooth negative results for requests, which occurs errors instead of actual results. After reviewing suggested methods, the possible negative aspects of their using were identified. Also was described ways for neutralization their negative effects and reduction of impact on the final result.

Web server; HTTP status code; workload; DoS attack; IIS; HTTP.

В связи с развитием способов применения уже известных инструментов для работы в сети Интернет, увеличивается актуальность вопроса устойчивости данных инструментов, поскольку ресурсы теперь представлены не только достаточно простым контентом, но и комплексными сложными порталами для общей работы, устойчивость и стабильность для которых имеет крайне весомое значение.

Для исследования по двум крупным причинам был выбран веб-сервер от компании Microsoft. Во-первых, он является одним из трёх крупнейших по распространённости веб-серверов и занимает (по данным портала netcraft.com на май 2013 г.) долю в 16,7 % рынка (против 53,4 % и 15,5 % у Apache и nginx соответственно). Во-вторых, немаловажным фактором является наличие достаточно удобного функционала. Он представлен в виде присутствующих в стандартной конфигурации различных дополнительных средств, например таких, как счётчики производительности, что позволяет не отвлекаться на дополнительные факторы обеспечения исследования веб-сервера, а целиком заниматься поставленной целью и при этом не волноваться, что этот дополнительный функционал как-то повлияет на результаты. Данная ситуация возможна из-за того, что веб-сервер изначально рассчитан на возможность совместного функционирования с данными измерительно-контрольными средствами.

Логика, которая производит поиск нужной страницы для выдачи, может быть обеспечена двумя путями.

Первый вариант. Эта логика может быть отдана системе управления контентом (CMS). При таком варианте, в случае генерации ошибочного запроса (например, запрос на показ несуществующей страницы, стандартный код состояния HTTP 404), обработка происходит непосредственно системой управления. При этом сама веб-платформа (в данном случае IIS) производит полную обработку данных, поскольку при таком варианте развития событий она не делает различий, существует страница, или нет. Веб-платформа, в данном случае, производит полную обработку сценария, с её позиции сценарий выполняется полностью и конечному клиенту транслируется уже результат (то, что было сгенерировано данным сценарием CMS). Факт наличия страницы определяется сценариями самой системы управления контентом, и взаимодействием этих сценариев с базой данных, подключённой к CMS. В общем виде схема варианта с наличием системы управления контентом представлена на рис. 1.

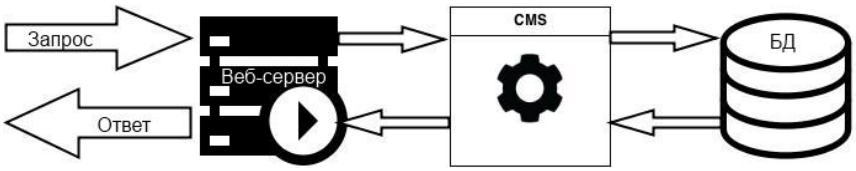

Рис. 1. Схема выдачи результата при использовании CMS

В случае значительного количества запросов подавляющая часть нагрузки будет уходить к БД, и, как следствие, дисковым подсистемам. Сам веб-сервер просто выполнит сценарий.

Рассмотрение данного подхода для модернизации является проблемным с той точки зрения, что рынок наводнён множеством различных систем управления контентом (только достаточно известных на сентябрь 2013 г. набралось больше сотни). Выработать общий подход вряд ли возможно, поскольку проекты имеют разную политику и принципы дальнейшего развития и включают в себя и open-source разработки, и проприетарные продукты, и продукты, работающие по схеме «SaaS» (программное обеспечение как услуга).

Второй вариант. Обработка логики сайта происходит через непосредственно сам веб-сервер, при этом на него также возложена и функция генерации диагностических сообщений. При таком варианте событий веб-платформа не просто ведёт обработку запроса на выдачу страниц, но и вынуждена обрабатывать дополнительные запросы из числа внутренних, если запрос извне нельзя корректно выполнить. Таким образом, происходит добавление задач сверх планируемых для веб-сервера. Такие внутренние запросы также необходимо обрабатывать для нормального информирования пользователя о результатах его первоначального запроса.

В общем случае второй вариант, т.е. система, где обработка сценариев для выполнения логики запрашиваемого веб-ресурса не предусматривает использование CMS, представлена на рис. 2.

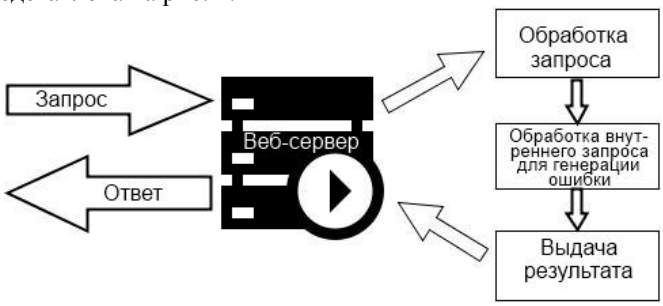

Рис. 2. Схема обработки запроса с использованием средств только веб-сервера.

В случае осуществления атаки на веб-сервер самым вероятным вариантом является атака отказа в обслуживании.

И если канал связи атакуемого ресурса оказался способен выдерживать поток запросов (например, путём отрезания ICMP echo-request пакетов), а сам сервер справляться с SYN-флудом (путём правильной настройки правил подключения), то в случае целенаправленного странного поведения могут возникнуть проблемы, например, раздувание потребления службами веб-сервера ресурсов сервера. Проблема может быть проигнорирована в случае наличия достаточно производительных платформ, что, по сути, является решением проблемы методом грубой силы, либо же переведена в такое русло, когда возникновение проблемы не столь заметно сказывается на потреблении ресурсов физической платформы. То есть увеличение мощности процессоров, предоставление больших объёмов памяти под веб-сервер, либо предоставление новых более быстрых дисковых подсистем, например на основе SSD. Явным минусом данного решения является цена его организации. Вопрос отказоустойчивости здесь стоит достаточно остро, поскольку в случае отказа восстановление данных является затруднительным из-за особенностей функционирования встроенного в твердотельные накопители фирменного программного обеспечения [1], поэтому необходимо добавлять к немалой стоимости решения ещё и стоимость резервирования. Но, тем не менее, простой опор на производительность технических средств не может решить причину проблемы. Обработка внутреннего сценария веб-сервера из-за недействительного запроса.

В случае действительного запроса нагрузка от обработки запроса, по большому счёту, роли не играет, поскольку конечный клиент должен, так или иначе, получить некий определённый ответ. Но возникает ещё один вопрос. А что же, произойдёт, если запрос не может быть обработан? Здесь начинаются проблемы вида, когда нагрузка есть, а результат заранее известен – получение ошибки с кодами состояния. Фактическое потребление ресурсов в данном случае сравнимо с потреблением ресурсов при обработке сложного действительного запроса (например, на выдачу веб-страницы объёмом более 500 Кб). При этом выдача коротких ненагруженных функционалом страниц в 5–10 Кб является не нагружающей сервер операцией. Парадоксом ситуации является то, что сценарий для выдачи кода состояния, несмотря на простоту выдаваемой информации (диагностическое сообщение +коды ошибок), вызывает нагрузку как запрос на показ веб-страницы с достаточно объёмным содержанием.

Проблему, опять же, можно решить кардинальным способом. Оставить веб-серверу возможность обрабатывать пользовательские запросы как обычно, но при этом, постаравшись минимизировать количество раз, которое обрабатываются внутренние сценарии в случае ошибок.

Вариантом реализации такого подхода может быть ситуация, когда все приходящие к веб-серверу запросы первоначально фильтруются межсетевыми экранами и пропускаются запросы только к существующим страницам. Но данный способ имеет достаточно крупные недостатки: добавление дополнительного компонента в систему функционирования веб-ресурса, что повышает сложность системы в целом, и поднятие нагрузки на межсетевой экран.

Помимо всего прочего пришлось бы обеспечивать на сетевом экране актуальность множества правил, поскольку последние пришлось бы модифицировать каждый раз, когда происходит правка структуры веб-ресурса.

Помимо предположения о том, что можно было бы сделать через механизмы, доступные в межсетевых экранах, было проведено небольшое исследование на предмет взаимосвязи нагрузки на сервер и обработки запросов с ошибками к веб-серверу.

В результате проведённого тестирования было установлено, что 250 одновременных запросов на показ достаточно объёмной страницы вызывают нагрузку на процессор тестового сервера в среднем в размере 17 % (усреднённые данные за 10 тыс. испытаний, максимальное отклонение не превышало 2 % в высшую сторону и 4% в нижнюю). При этом запросы, вызывающие генерацию страницы с отчётом об ошибке, в такой же ситуации показали примерно схожие результаты – 15 % потребления процессорного времени при 250 одновременных запросах (так же усреднённые данные примерно за 10 тыс. испытаний). Важно заметить, что результат тестирования более простой страницы с такими же показаниями (250 одновременных запросов при 10 тыс. испытаний) оказался достаточно интересен. Обработка простой страницы в 689 байт с несколькими строками текста в качестве содержания занимала в среднем всего лишь 6 % процессорного времени.

Вывод достаточно очевиден. Нагрузка при обработке простых страниц меньше, чем нагрузка в случае обработок ошибочных запросов. Эту идею можно использовать для попыток создания автоматизированных интеллектуальных систем, которые смогли бы не просто отслеживать приходящие запросы, события и вести журналы событий, но и производить выполнение некоторого набора действий, призванного снизить нагрузку на сервер за счёт изменения принципа показа страниц с ошибками и кодами состояний.

Здесь, в качестве варианта, можно создать гибрид из веб-сервера и схемы, применяемой в системах управления контентом. Заимствованной схемой, в данном случае, будет то, что страницы с ошибками и кодами состояний будут показывать-

ся не те, что генерируются сервером путём обработки своих внутренних сценариев, а те, которые предоставили бы CMS (или подобные этим страницам). То есть, со стороны сервера данная схема будет выглядеть как обработка обычной действительной простой страницы, не нагружающей сервер. Вопросом остаётся то, как эта система может быть реализована. Здесь на помощь как раз и должна приходить уже упомянутая интеллектуальная система, анализирующая приходящие запросы и историю запросов за некий уже прошедший период времени. Поскольку вероятность того, что разные реальные люди станут запрашивать одни и те же несуществующие страницы не так уж и велика, то наличие факта такого потока запросов может свидетельствовать о целенаправленном воздействии на веб-сервер, то есть, фактически, об атаке в виде множества сгенерированных ошибочных запросов. В данном случае то, что послужило источником – не сильно важно, поскольку повышенная нагрузка на сервер возникает так или иначе. Все такие приходящие запросы анализируются по факту запроса или постфактум, на основе анализа журналов событий и обращений [2]. При этом анализу можно не подвергать всю имеющуюся базу логов, а достаточен анализ за некоторое последнее время (последние несколько часов или дней максимум, поскольку более длительными атаки обычно не бывают). Этого более чем достаточно, чтобы выявить несуществующие адреса, по которым осуществляются запросы, и, в случае достаточно постоянного списка, можно разместить действительные простые веб-страницы, являющиеся заглушками, по данным адресам.

Это действие приведёт к тому, что удастся перевести часть запросов из разряда запросов, генерирующих страницы с кодами ошибок и вызывающих повышенную нагрузку на сервер, в разряд запросов, которые фактически вызывают теперь уже действительные страницы с минимальной нагрузкой на сервер. Процесс размещения страниц-заглушек также может быть отдан на исполнение системе обработки недействительных запросов.

Однако использование данного подхода, как и использование других, сочетает как положительные стороны, так и некоторые моменты, по которым могут возникнуть вопросы.

Первым вопросом является загромождение структуры сайта, которое может быть вызвано подобными заглушками. Фактически, автоматизированная подстановка страниц-заглушек в различных разделах сайта может решить проблему нагрузки, но при этом породить огромное количество одинаковых страниц во всех директориях сайта. Эта ситуация может затруднять как управление сайтом (или его модификацию), так и своевременную подчистку заглушек, потерявших актуальность. Кроме того, наличие множества мелких записей (коими будут являться подобные образования) может, опять же, снижать производительность системы, поскольку файловая система будет вынуждена обрабатывать множество записей. Решением данного вопроса может быть использование такого инструмента веб-сервера IIS, как виртуальный каталог. Таким образом, удастся собрать все заглушки в едином месте хранения, отделив их от действительно функциональной части веб-ресурса. Вопрос подчистки потерявших актуальность файлов также становится легче решить, например, отслеживая даты последних обращений к тем или иным файлам и удаляя потерявшие актуальность.

Вторым вопросом является борьба со случайными срабатываниями, когда ошибка – действительно непреднамеренная ошибка, а не атака. Для борьбы с такой проблемой можно просто использовать механизмы определения частоты события. Если частота обращений не достигает за определённый временной интервал некоего установленного лимита, то и создания страницы не происходит.

Третьим вопросом является ситуация, когда нагрузка создаётся запросами, которые не повторяют обращений к одним и тем же несуществующим страницам, либо количества обращений и запросов к страницам недостаточно, чтобы преодолеть минимально необходимый лимит. В таком случае на автоматизированную систему можно возложить функцию анализа взаимосвязи ошибочных запросов и отправителя [3]. Простейшим случаем будет блокировка запросов с определённых адресов, от которых пришло достаточно весомое количество ошибочных запросов за некий промежуток времени.

Реализация динамической блокировки в начале атаки и снятие блокировки после в данном случае не является сложной задачей, поскольку автоматизированная система может просто обновлять конфигурации межсетевых экранов в части блокировок. Ситуация отличается от предыдущей с использованием межсетевых экранов тем, последним нет необходимости держать в конфигурациях всю базу актуальных адресов, а лишь достаточно адресов, подвергающихся блокировке.

Таким образом, применяя комплекс перечисленных мер, есть возможность сократить нагрузку от специально сгенерированных ошибочных запросов на значительную величину (касательно тестового сервера, использовавшегося для исследования, сокращение нагрузки при реализации подобной схемы защиты в идеальных условиях составило бы до 9 %, т.е. более чем на половину).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК.

1. *Максимов А.М., Тищенко Е.Н.* Особенности использования носителей информации в защищённых информационных системах // Известия ЮФУ. Технические науки. – 2011. – № 12 (125). – С. 238-244.
2. *Ciesielski V., Anand L.* Data mining of web access logs from an academic web site. Design and application of hybrid intelligent systems // IOS Press Amsterdam. – 2003. – С. 1034-1043.
3. *Тищенко Е.Н., Шарыпова Т.Н.* Формализация выбора различных вариантов системы защиты информации от несанкционированного доступа в среде электронного документооборота // Вестник Ростовского государственного экономического университета (РИНХ). – 2010. – № 32. – С. 226-233.

Статью рекомендовал к опубликованию д.т.н., профессор П.Н. Башлы.

Максимов Алексей Михайлович – Ростовский Государственный Экономический Университет (РИНХ); e-mail: ironmanpc@rambler.ru; 344002, г. Ростов-на-Дону, ул. Б. Садовая, 69, к. 211, 306а; тел.: 88632613858; аспирант.

Maximov Alexey Mikhailovich – Rostov State University of Economics; e-mail: ironmanpc@rambler.ru; 69, B. Sadovaya streeet, room 211, 306a, Rostov-on-Don; 344002, Russia; phone: +78632613858; postgraduate student.

УДК 004.054

А.Г. Богораз, О.Ю. Пескова

МЕТОДИКА ТЕСТИРОВАНИЯ И ОЦЕНКИ МЕЖСЕТЕВЫХ ЭКРАНОВ*

Рассмотрены основные российские и зарубежные методики оценки защищенности информационных систем в приложении к анализу межсетевых экранов. Показана необходимость разработки методологии тестирования межсетевых экранов как на стендах, так и в реальной системе. Был выполнен анализ следующих методик тестирования инфор-

* Работа поддержана грантом РФФИ 13-07-00244-а.