Анализ методов оцифровки, защиты и длительного хранения материалов на Интернет-ресурсах

Б.С. Яковлев, Н.Е. Проскуряков, Н.Н. Архангельская
Тульский государственный университет
bor_yak@mail.ru, vippne@mail.ru, arhangel_nataly@mail.ru

Аннотация

Представлен анализ контента и разработан метод долговременного хранения и проверки целостности цифрового контента на Интернет-ресурсах, в том числе с периодической загрузкой файлов с сервера на компьютер разработчика с последующим сравнением контрольных сумм копий файлов. Создана программа для выполнения этих действий в ручном и автоматическом режимах. Были проведены эксперименты и исследование скорости проверки контента. Полученные результаты позволяют рекомендовать этот метод для широкого использования.

Ключевые слова: оцифровка; электронное издание; Интернет; защита; хранение; веб-сайт.

Введение

В настоящее время наблюдается рост активности в оцифровке и размещении в сети Интернет материалов архивов, библиотек, музеев, образовательных учреждений и др. организаций.

Процесс получения цифровых копий долгий, дорогой, кропотливый и сложный процесс, в котором можно выделить несколько постоянных и второстепенных этапов, зависящих от конкретной задачи.

Классический вариант технологического процесса оцифровки содержит следующие пункты:

- отбор материалов, исходя из их важности, ценности;
- процесс оцифровки;
- обработка результатов оцифровки;
- проверка качества проведенных этапов;
- отправка данных на длительное хранение.

Сегодняшнее состояние цифровых и телекоммуникационных технологий дает возможность дополнить классический процесс некоторыми специфическими этапами из-за потребности организаций выкладывать свои материалы в общий доступ, для рекламы и привлечения новых посетителей. Все

основные и дополнительные этапы обычно выполняются параллельно, поэтому они не сильно замедляют процесс подготовки Интернет-порталов к окончательному запуску, но в них содержатся главные проблемы защиты и будущее коллекций музеев, архивов др. организаций.

1. Выбор итогового формата для длительного хранения

Первый неочевидной, но, тем не менее, важной проблемой является выбор итогового формата для длительного хранения. Известно, что одной из первых стран, запустивших программу оцифровки своих фондов, являются США. Поэтому целесообразно посмотреть на опыт их работы и результаты. Когда в 1998 году прошла большая часть программы, стали появляться сообщения о том, что часть документов перестала воспроизводиться, либо устарели версии программ для их чтения.

Подобные проблемы могли быть выявлены только с течением времени, и это не является виной разработчиков проекта оцифровки документов.

В подобную ситуацию попали и российские музеи, они начали процесс активной оцифровки с 2005-х годов. В то время считалось, что оптимальным выбором станет JPEG, в силу этого крупные музеи выбирали именно его.

Но т.к. работа по оцифровке начата задолго до появления сжатия JPEG 2000, заявленного как алгоритм сохранения данных без потерь качества, то получается, что вся сегодняшняя технология оцифровки получает и хранит данные при базовом сжатии этого формата. Это значит, что в концепцию заложено следующее — уменьшение объемов графического файла за счет ухудшения качества, уменьшения воспроизводимых оттенков цветов и т.п.

Архивные учреждения тоже взяли за основу JPEG, хотя работа по их оцифровке осуществляется по программе с 2010-2020 г. и ориентировалась при создании на уже существующие методики в других странах, в том числе на Национальное управление архивов и документации (NARA — National Archives and Records Administration, USA) [1, 2].

Несмотря на то, что сохранение данных в цифровом виде регулируется международными стандартами оцифровки, мы считаем, что более выгодно для длительного хранения использовать изначально RAW-формат, несмотря на его итоговые объемы, т.к. в процессе дальнейшего развития оборудования, технологий и программных комплексов обработки графики и методик создания Интернет-порталов он более выгоден как основа при последующей обработке после оцифровки из-за своих изначальных свойств [3].

2. Разработка и проектирование платформы сайта

Вторая по значимости проблема заключается в проектировании платформы сайта, которой будет удобно пользоваться посетителям и подходящую для воспроизведения любого типа контента. Стоит отметить, что под этим больше стоит понимать не дизайн, а именно программный подход к реализации этого сайта. Чаще всего разработчики не пишут сайт с нуля, а адаптируют уже известные разработки под потребности заказчиков. С точки зрения защиты информации подобная тенденция крайне опасна, т.к. популярные системы управления сайтами (CMS, движок для сайта) как раз являются самыми

заражаемыми ресурсами в сети Интернет, т.к. имеют открытый код, известную базовую структуру каталогов и размещение файлов, даже названия файлов одинаковы. Кроме того, в подобных системах применяются одинаковые настройки безопасности, а также методы передачи данных, например, между базами данных и PHP сценариями и т.п. Таким образом, можно сказать точно, что если сохранять базовые настройки для подобных CMS, то это будет всегда приводить к повторному заражению ресурса.

В большинстве случаев зараженные сайты ведут себя одинаково: при переходе на ссылку происходит перенаправление на другой ресурс. Осуществляется это за счет замены тегов <body> или </body> или аналогичных тех же тегов с добавлением кода перенаправления, например, на JavaScript:

<script language = javascript'> document.location.href= 'http://рекламныйсайт.ru'; </script>

Есть варианты заражения и баз данных, но данная проблема решается только частыми резервными копиями базы. Вычленить угрозу до заражения достаточно сложно.

Проблема безопасности сайтов и долговременных хранилищ библиотек, архивов, издательств, университетов в настоящее время усугубилась также из-за появления «вирусов-шифровальщиков», которые поражают локальные и сетевые диски, шифруя все популярные файлы, при этом каждый раз меняя пароль и алгоритм шифрования. Из-за этого на сегодняшний день данные после заражения восстановлению не подлежат.

Ситуация может резко ухудшиться из-за сообщения от 13 января 2016 г. международной антивирусной компании ESET о появлении нового шифратора Ransom32. Опасность этой новости заключается в том, что Ransom32 является первым шифратором, написанным на языке JavaScript и работающим на платформе NW.js. В настоящее время экспертами исследованы версии Ransom32 для Microsoft Windows, но программа может быть адаптирована для Linux и Apple OS X. Это в свою очередь означает, что не могут быть 100% защищены Интернет-площадки, т.к. большинство из них работает на системах Linux (Unix).

3. Программное обеспечение и защита данных на WEB-серверах

Чтобы определить, как бороться с описанными выше угрозами, необходимо понимать, как устроена работа WEB-серверов и что они из себя представляют.

Во-первых, нужно отличать серверное оборудование от программного обеспечения, которое осуществляет работу сайтов в Интернет.

Во-вторых, большинство серверов функционируют на платформе Linux, т.к. эта система на протяжении всех лет существования показывала высокую стойкость к заражению вирусами. Именно поэтому до появления вируса Ransom32 все заражения сайтов были относительно безвредны и быстро излечивались, кроме того они происходили исключительно из-за несовершенства разработанной платформы сайта, а не заражения хостинга.

Общая схема по работе оборудования представлена на рис. 1. Уязвимостям заражения подвержено то оборудование, которое участвует в обработке сигнала, т.е. «управляющий компьютер», получающий запросы от пользователей, обрабатывающий их, отправляющий уже свои запросы по локальной сети в основное хранилище к «стойкам».

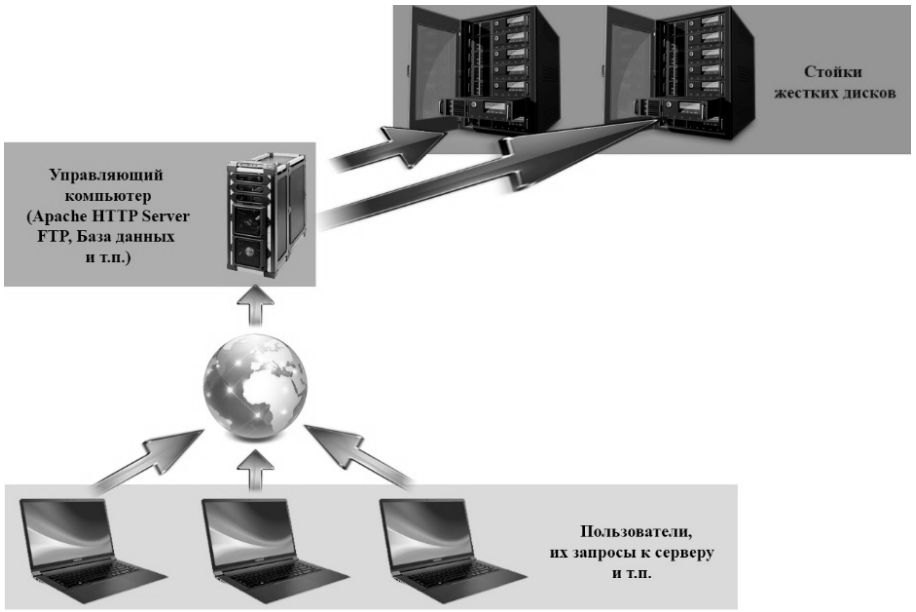

Рис. 1.. Общая схема оборудования для web-сервера

В классической схеме на «управляющем компьютере» установлено программное обеспечение по резервному копированию, антивирусная система и программы, регламентирующие права пользователей, доступа к файлам и т.п. Он получает, проверяет, обрабатывает запросы от пользователей и после этого принимает решение давать доступ к файлам, отдавать ли ответ на запросы или нет.

«Стойки» представляют из себя корпус с отсеками под жесткие диски. В каждом отсеке есть micro-материнская плата с предустановленной операционной системой Linux, LAN разъемом. Кроме того, данное оборудование поддерживает работу RAID-массивов. Обмен данными между «стойками» и «управляющим компьютером» осуществляется через запросы, текстовыми командами.

В текущей ситуации в этой компоновке заразиться может «управляющий компьютер», т.к. гипотетически он может быть создан и не на системах UNIX. На жестких дисках в «стойках» хранится относительно статичная информация и автоматически создается резервное копирование при каждой записи, перезаписи файлов, т.к. используется RAID-массив. Поэтому мы имеем отличную систему хранения файлов на сервере, позволяющую почти на 100% защитить файлы от

потери, но они могут быть заражены программным образом. Связано это с тем, что в «стойках» не может работать и обновляться антивирус, т.к. система относительно замкнута и большинство фирм, работающие на рынке Интернет услуг имеют огромные объемы данных и проверять файлы пользователей обычным сканером в реальном времени они просто не в состоянии.

Поэтому в современных реалиях появились «антивирусы для сайтов». Обычно выделяют два способа проверки сайтов на вирусы:

- проверка сайта на уязвимости при помощи on-line сервисов;
- проверка файлов сайта на вредоносный код.

Первая категория известными запросами проверяет сайт на устойчивость и уязвимости, и в итоге может дать хорошую информацию о безопасности выбранного вами поставщика услуг по размещению сайта в Интернет. Однако, она бесполезна с точки зрения обнаружения самих вирусов на сайте, т.к. внешние сайты не могут получать содержимое PHP, ASP файлов. Они могут анализировать только итог их работы, тот результат, который выдает сервер после их выполнения.

Что касается второго варианта проверки сайтов непосредственно на вирусы, то он тоже далек от идеала. Его реализация осуществляется за счет запуска «секретной» страницы на вашем сайте вручную или по таймеру. Ее задача сводится к опросу всех файлов в корневой директории сайта и во всех подкаталогах в ней, затем быстрый обход этих файлов циклом на поиск известных на сегодняшний день вирусных записей.

Но тут опять возникают 3 проблемы:

- Таймеры в серверных языках программирования не работают, пока не запущена страница с этим кодом, а значит мы, не прибегая к платным сервисам, не сможем автоматизировать запуск подобных решений. Поэтому вы должны либо держать открытой данную страницу всегда, либо использовать платные сервисы, с проверкой файлов сайтов по расписанию, которых в сети Интернет много.
- Сами страницы «антивируса» могут стать разносчиками вредоносного кода по сайту, т.к. могут заразиться и содержать в себе функцию, которая каждый раз при опросе любого файла на сайте применяет к нему замену стандартных тегов на вредоносный код. Это будет катастрофой для вашего сайта, т.к. в этом случае будут опрошены все важнейшие сценарные файлы на сервере и заразятся.
- Вирусы могут содержаться не только в кодах страниц сайта. Базы данных, файлы архивов, графические данные могут быть заражены. Эти файлы WEB-антивирусы проверить не смогут, т.к. они не работают с бинарной структурой файла.

Вероятно, понимая, что подобные «антивирусные» решения имеют всегда открытый код и никакой самозащиты выстроить нельзя, ни одна серьезная фирма по разработке антивирусного ПО не адаптирует свои вирусные базы и методы обнаружения под нужды держателей сайтов. Поэтому существующие в Интернете решения «WEB-антивирусов» создают небольшие фирмы, которые всегда имеют ограниченные ресурсы по обнаружению новых угроз и поддержки баз данных в оптимальном состоянии. Из-за подобных обстоятельств и

несовершенства технологии поиска данных, такие программы в результате дают только информационный отчет, не производя лечения.

Стоит, однако, обратить внимание на то, что такие WEB-антивирусы все же находят зараженные файлы, но это уже результат действия заражения. По факту сам источник заражения — тоже код, который очень сложно найти даже человеку, т.к. он повторяет структуру и манеру программирования человека. Т.е. чаще всего это с виду вполне рабочая функция с нормальным синтаксисом и описанием действий на любом из серверных языков программирования. Особенно тяжело тело вируса найти в CMS, т.к. вы не имеете понятия, где искать и за что отвечает тот или иной участок кода CMS.

Что касается угроз, от которых защищают WEB-антивирусы, то это классические вирусы перенаправления, рекламные баннеры и прочие визуальные эффекты, мешающие пользователям использовать ваш сайт.

К сожалению, есть и другая проблема в области защиты данных на WEB-серверах: сам сервер может быть заражен, если он создан не на основе системы Unix. Тогда как бы вы не лечили сайт, какие бы уязвимости не нашли и исправили, все равно будете постоянно заражены и придется постоянно лечить свой Интернет-ресурс. Исправить это нельзя из-за того, что WEB-антивирусы не могут проверять файлы, находящиеся на уровень выше, чем находится сам сайт. Для того чтобы понять это необходимо вспомнить как устроено взаимодействие программ на WEB-сервере и какое ПО на нем установлено.

Все программное обеспечение установлено на «управляющем компьютере» (рис. 1). В комплект программ входят ядро (Apache), серверный язык (PHP, ASP), FTP-сервер (обычно FileZilla FTP-Server), база данных (MySQL или Oracle) и модуль отправки писем (SMTP-server). При этом ядро связывает все программы воедино, путем отправки запросов в соответствующие службы. Система конфигурации Apache основана на текстовых конфигурационных файлах. Имеет три условных уровня конфигурации: конфигурация сервера (httpd.conf); конфигурация виртуального хоста (httpd.conf, начиная с версии 2.2 — extra/httpd-vhosts.conf); конфигурация уровня директории (.htaccess).

Доступ к файлу конфигурации уровня директории (.htaccess) может получить владелец сайта, но конфигурация сервера (httpd.conf) и конфигурация виртуального хоста (httpd.conf) в иерархии выше, и до них владелец сайта доступа не имеет.

Чтобы осуществить такое разделение прав разработчики Apache создали 2 области — системную и пользовательскую. Формально в пользовательской области находятся все файлы вашего сайта и всех клиентов данной фирмы, а в служебной — только файлы Apache и его настройки. При попытке зайти на сервер при помощи FTP-протокола пользователи просто не увидят каталоги служебной области Apache.

Подводя промежуточный итог можно сформулировать следующие проблемы Интернет-сайтов:

- антивирусы для сайтов малоэффективны и недейственны;
- хостинги не проводят постоянный контроль за вирусами на своих серверах из-за больших объемов данных пользователей;
- «вирусы-шифровальщики» — очень опасны, их стоит выделять отдельно, т.к. после их действий информацию можно восстановить

только при наличии резервных копий файлов, незатронутых данным вирусом. В таких ситуациях RAID-массивы не помогут, т.к. они всего лишь дублируют данные, и в случае заражения запомнят измененный вирусом файл.

4. Разработка метода защиты и длительного хранения материалов на Интернет-ресурсах

Решая перечисленные выше проблемы, выявленные в ходе анализа, можно предложить, что третий пункт может быть решен при помощи технологий управления версий (Version Control System, VCS или Revision Control System) и сетевых хранилищ от Яндекс, Mail.ru, Google Диск и т.п. При этом более правильным решением является использовать Google Диск совместно Google Drive, который автоматически синхронизирует данные из вашего каталога файлов со своим облачным хранилищем и не только сохраняет изменения в нем, но и ведет контроль версий файлов.

Первые же две проблемы можно решить несколькими способами:

- Постоянно визуально проверять состояние и поведение вашего сайта в Интернет частыми заходами на него;
- Как угодно часто загружать файлы с вашего сайта к себе на PC во временную директорию и проводить проверку антивирусной программой. При обнаружении угроз перезаписывать данные, содержащиеся на сервере вашими незараженными копиями;
- Сверять файлы на сервере с копиями на вашем жестком диске.

Стоит пояснить, что результат заражения вирусами визуально выявляются быстрее, чем обход всех файлов программным образом, но он не может быть автоматизирован, а значит и не может относительно часто применяться, что делает его практически бесполезным.

Второй способ более действенен, но требует больших временных затрат и, что более важно, очень сильно связан с реакцией на зараженный файл со стороны вашего антивируса. Известно, что корпорации ESET и Kaspersky Lab при обнаружении зараженного файла вирусами перенаправления удаляют или перемещают эти файлы, не производя их лечения. Dr.WEB более лоялен и производит автоматическое лечение этих файлов. Но данный способ все равно содержит один явный проблемный пункт — антивирусные системы не распознают источник заражения, т.е. само тело вируса, т.к. для них это обычный код страниц Интернет. Данную проблему вручную решает сам человек.

Третий способ подразумевает использование проверки контрольных сумм файлов (CRC). Он более универсален, т.к. может быть применен в большинстве случаев, не зависит от описаний вирусных сигнатур в базах антивирусных программ и, главное, может обезвредить тело вируса, потому что в случае заражения файлов на сервере код так или иначе изменит размер файла.

Исходя из вышеописанных рассуждений можно сделать вывод, что наиболее действенным способом проверки файлов WEB-серверов на вирусы станет проверка контрольных сумм.

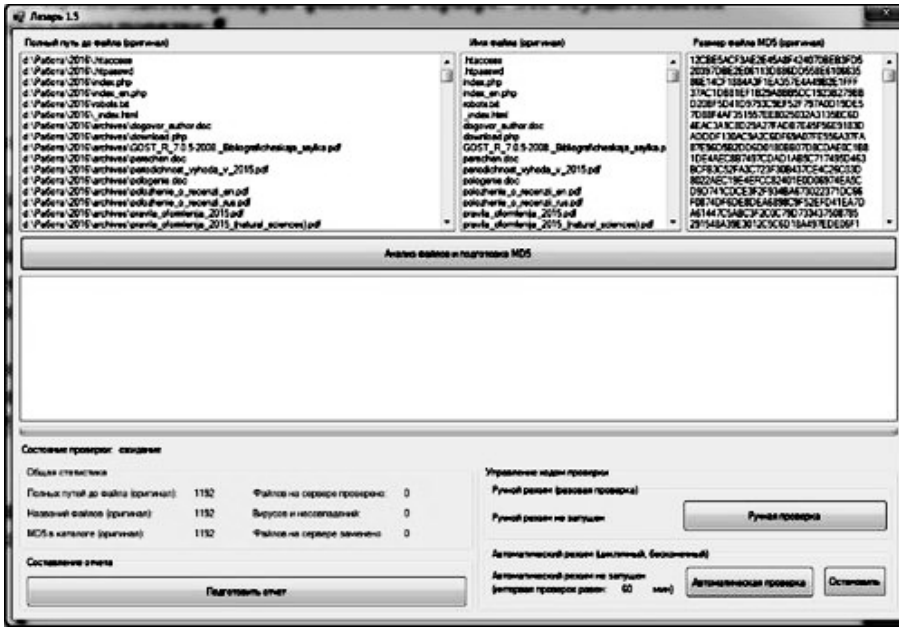

Рис. 2. Внешний вид программы проверки контрольных сумм

Программным образом получить CRC по запросам к файлам по FTP невозможно. Поэтому можно предложить 2 альтернативных способа проверки:

- опрос файлов сервера средствами серверных языков (PHP, ASP);
- последовательная загрузка файлов с сервера на PC и проверка контрольных сумм файлов.

Первый способ будет работать более быстро, но в нем заложены некоторые недостатки: при осуществлении такого опроса сам файл может стать переносчиком заражения, и при этом есть вероятность ошибки определения контрольной суммы из-за разницы файловой системы Unix с пользовательской (чаще всего FAT32, NTFS).

Поэтому предлагается использовать последний вариант. Чтобы избежать ошибки определения контрольной суммы предполагается загрузка файлов с сервера на PC пользователя, последующее сравнение суммы с копией файлов сайта.

С этой целью авторами статьи в январе 2016 г. была создана программа на языке программирования PHP, специально разработанного для написания WEB-приложений, которая выполняет данные действия в ручном (одноразовая проверка по требованию пользователя) и автоматическом режимах (постоянный режим проверки через установленное пользователем время), представленная на рис. 2. На разработанное ПО авторы статьи отправили в мае 2016 г. заявку в Роспатент на государственную регистрацию программы для ЭВМ.

Суть работы ПО заключается в следующем:

1. Указывается каталог где хранится оригинальная (эталонная) копия проверяемого сайта на PC.

2. В указанном каталоге, проводится опрос существующих файлов.

3. Производится подсчет контрольных сумм файлов и перевод этих данных в MD5.

4. Производится проверка файлов на сервере. Это осуществляется в следующем порядке:

4.1. Сохранение файла средствами FTP во временной директории;

4.2. Получение контрольной суммы временного файла в MD5;

4.3. Сравнение контрольной суммы временного файла с MD5 оригинального (эталонного) файла.

5. В случае обнаружения несовпадения файл из эталонного каталога загружается на сервер с полной заменой. Также данный файл не удаляется из временного каталога программы проверки сумм, для дальнейшего ручного анализа человеком. В случае отсутствия несовпадений временный файл удаляется с компьютера пользователя и происходит повторение операции 4.1 для следующего файла.

6. В случае активации автоматического режима работы ПО будет повторять пункты 4–5 через заданное пользователем время. Если же был активирован ручной режим, то по окончании проверка остановится.

7. По окончании проверки можно просмотреть сохраненный отчет работы.

5. Экспериментальная проверка разработанного метода

Данное программное обеспечение (ПО) было разработано и использовано для контроля целостности данных сайта журналов серии «Известия Тульского государственного университета» («Известия ТулГУ»), входящих в перечень ВАК РФ.

В ходе работы было выявлено, что вирусы или ошибки WEB-сервера могут приводить к удалению файлов с сайта. Встречались случаи обнаружения сбойных файлов, когда происходил разрыв соединения с сетью. Предложенный метод и разработанное ПО справляются с данными проблемами.

Используя результаты предыдущих исследований [4], были проведены эксперименты для изучения зависимости времени проверки от объемов данных и степени их заражения, результаты которых приведены в табл. 1. В качестве испытуемого файла использовался архив из разных типов контента (текст, аудио- и видеофайлы, EXE-файлы и пр.), разбитый на части по 100 Мб. Определялось время проверки данных объемом 100 Мб, 500 Мб и 1 Гб.

Также устанавливалось время, затрачиваемое на лечение сайта в случае его полного заражения. Для этого файл был разбит на части по 100 Мб и выбрана одна из его частей. Проверяемый объем создавался за счет добавления этой части на сервер в каталоги с порядковыми номерами от 1 до 10. Это позволило строго контролировать объем данных. В случае проверки сайта на заражение, в каталоги помещался архив объемом 100 Мб с названием файла оригинала. Этим создавалось искусственное несоответствие CRC оригинального файла с проверяемым на сервере.

Стоит обратить внимание на то, что, если бы использовались малые по объему файлы, время проверки было бы меньшим, т.к. процесс опроса файлов проходил бы более динамично. Результаты исследования представлены на рис. 3 и 4.

Таблица 1. Время проверки CRC в зависимости от объемов файлов, с

Объем файла	Время проверки контрольных сумм					Среднее значение
	опыт 1	опыт 2	опыт 3	опыт 4	опыт 5	
заражение отсутствует						
100 Мб	16	15	15	17	15	16
500 Мб	84	83	84	85	84	84
1 Гб	161	162	161	161	163	161
файлы отсутствуют						
100 Мб	51	43	45	43	46	44
500 Мб	220	221	221	220	222	221
1 Гб	429	428	429	430	427	428
файлы заражены						
100 Мб	61	62	61	60	61	61
500 Мб	310	312	310	309	308	310
1 Гб	657	655	657	656	657	657

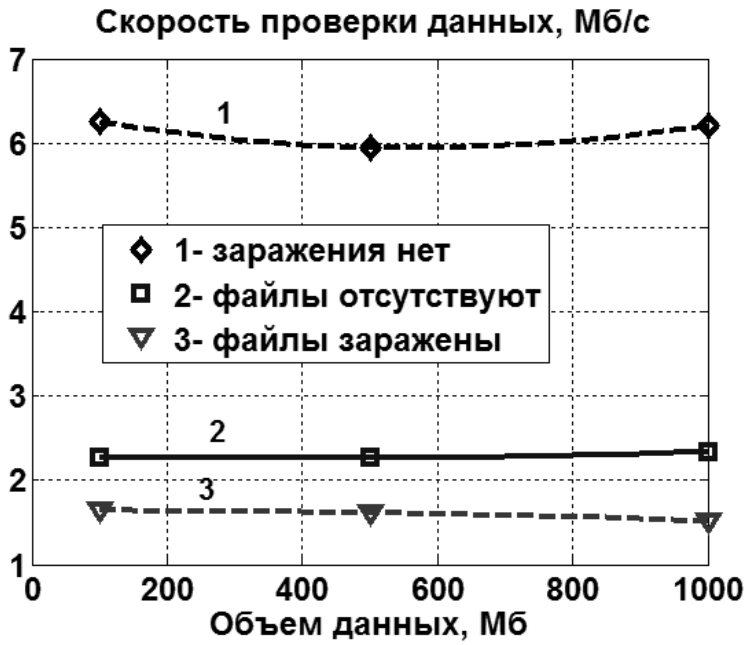

Рис. 3. Графики скорости проверки данных:
1 – заражения нет; 2 - файлы отсутствуют;3 - файлы заражены

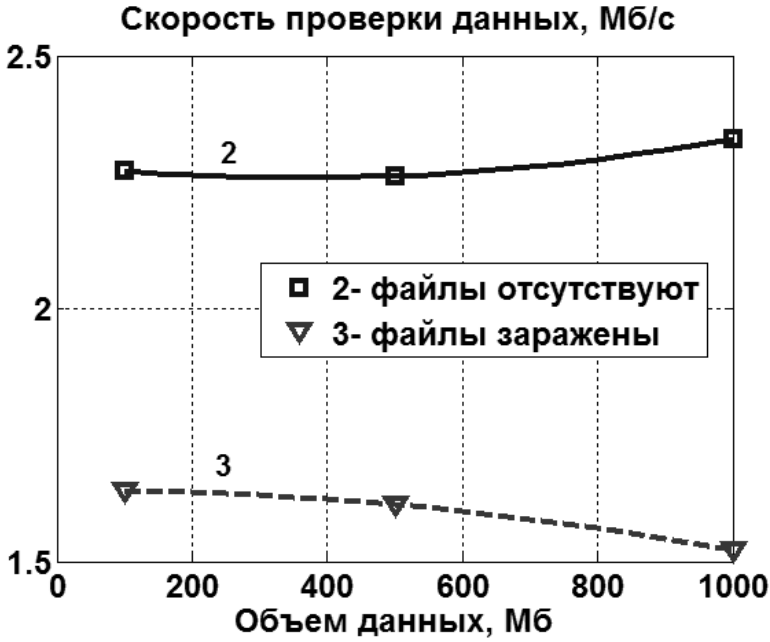

Рис. 4. График скорости проверки данных при:
2 - отсутствии файлов; 3 - заражении файлов

Анализ скорости проверки данных (рис. 3) показал, что при отсутствии заражения файлов на сервере скорость сравнения контрольных сумм в 3–4 раза больше, чем в ситуации, когда файлы отсутствуют или были заражены. Это обусловлено тем, что по предлагаемому алгоритму файлы должны быть сохранены с сайта, а после при не совпадении контрольных сумм, загружены обратно на сервер.

В случае отсутствия файлов на сервере (кривая 2 на рис. 4), скорость проверки данных равна скорости их прямой загрузки на сервер, т.е. реальной скорости работы Интернет-подключения. Данная связь проиллюстрирована на рис. 4.

Минусом предложенной системы проверки сайтов является снижение скорости проверки при заражении файлов (кривая 3 на рис. 4), т.к. фактически это приводит к двукратному увеличению времени проверки.

Однако, такой подход необходим, во-первых, из-за отсутствия явного метода опроса свойств файлов через FTP-соединение средствами языков программирования, во-вторых, из-за разницы файловых систем сервера и PC.

Заключение

Проведенные исследования позволяют сделать следующие основные выводы по работе:

- В случае организации длительного хранения оцифрованных данных на WEB-сервере или локальной сети необходимо использовать только бинарный способ записи файлов.
- В большинстве случаев WEB-антивирусы не в состоянии справиться с реальной угрозой для сайтов.
- Метод борьбы с вирусными угрозами для WEB-ресурсов, основанный на проверке контрольных сумм показывает хорошие результаты и может применяться для этих задач.
- В современных условиях необходимо более активно использовать решения по резервному копированию данных, использовать облачные технологии и другие способы.
- При разработке сайтов архивных, музейных, образовательных учреждений необходимо стараться группировать файлы сценариев по важности, т.к. это повлияет на быстроту проверки ресурса.

Литература

[1] Малых В.В. Российская vs американская концепции развития госархивной отрасли. [Электронный ресурс]. Режим доступа: URL: http://www.pcweek.ru/ecm/blog/ecm/6723.php#29642 (дата обращения: 10.03.2016).

[2] Юмашева Ю.Ю. Методические рекомендации по электронному копированию архивных документов и управлению полученным информационным массивом. [Электронный ресурс]. Режим доступа: URL: http://archives.ru/documents/rekomend_el-copy-archival-documents.shtml (дата обращения: 28.02.2016).

[3] RAW — плюсы и минусы. Или JPEG? [Электронный ресурс]. Режим доступа: URL: http://www.compartstudio.com/school/book/1-01/1-01.htm (дата обращения: 10.03.2016).

[4] Яковлев Б.С. Исследование стойкости несетевых электронных изданий и основных видов контента // Б.С. Яковлев, Н.Н. Архангельская, Н.Е. Проскуряков / Информационное общество: образование, наука, культура и технологии будущего. Труды XVIII объединенной конференции «Интернет и современное общество» (IMS-2015). Университет ИТМО; Библиотека Российской академии наук. Санкт-Петербург, 2015. С. 153–166.- URL: http://elibrary.ru/item.asp?id=24269693.

Analysis of digitization methods, of protection and storage of materials in the Internet resources

B.S. Yakovlev, N.E. Proskuriakov, N.N. Arhangelskaia
Tula State University

The content analysis submitted and the method of long-term storage and check of the integrity of digital content on the internet resources, including the periodic downloading of files from the server to the computer of designer and comparing them the checksum with the copy of files offered. The program created to perform these actions in manual and automatic modes. Research and analysis of the speed of data check conducted. Their results allow us to recommend this method for widespread use.

Keywords: digitization; electronic edition; Internet; protection; storage; site.