УДК: 004.9

КОНСТРУИРОВАНИЕ САЙТА, ЗАЩИЩЕННОГО ОТ БЛОКИРОВОК

Бартенев, В.А. , Негребецкая В.И.

ФГБОУ ВО «Курский государственный университет», колледж коммерции, технологий и сервиса, Россия, Курск, e-mail: vbibliotekar@bk.ru, violetta-negrebe1@mail.ru

Рассмотрена история обслуживания сайтов. Выявлены наиболее опасные вирусные программы. Выявлены причины понижения быстродействия и безопасности информации сайта. Выявлены наиболее удобные и надежные программы для повышения стабильности работы интернет ресурса. Указаны библиотеки и базы данных, позволяющие получить точную информацию о вирусной программе. Указаны оптимальные работы по повышению безопасности сайта, понижению используемого трафика, уменьшению копирования информации и уменьшению риска плагиата и полного копирования сайта.

Ключевые слова: обслуживание сайтов, безопасность сайтов, вирусные программы, Code Red, Download agent, Positive Technologies.

DESIGNING A SITE THAT IS PROTECTED FROM BLOCKING

Bartenev, V. A., Negrebetskaya V. I.

Kursk state University, College of Commerce, technology and service, Kursk, Russia, e-mail: vbibliotekar@bk.ru, violetta-negrebe1@mail.ru

The history of site maintenance is considered. The most dangerous virus programs have been identified. The reasons for reducing the performance and security of the site's information were identified. The most convenient and reliable programs for improving the stability of the Internet resource have been identified. Libraries and databases that allow you to get accurate information about the virus program are specified. The optimal work to improve site security, reduce the traffic used, reduce the copying of information and reduce the risk of plagiarism and complete site co-paging is indicated.

Keywords: site maintenance, site security, virus programs, Code Red, Download agent, Positive Technologies.

Информация должна быть в безопасности - пожалуй, эта аксиома известна всем. И несомненно, большинство пользователей знают, что такое Firewall и троянские вирусы и какими средствами можно обеспечить защиту сети. Однако не все знают, как они работают и как оптимально настроить систему защиты компании. Тем не менее именно от этого зависит не только сохранность данных, но и существование предприятия в целом. Важным звеном в

работе каждой кампании является ее рабочий Web-ресурс или на простом языке - сайт. Правильное ведение сайта определяет его дальнейшую работоспособность. Естественно, что у каждого профессионального программиста существуют свои методы работы с Web-ресурсами, с ведением и обслуживанием сайта и обеспечением ему должной защиты.

Интернет уже перестал быть простым набором html-страниц и представляет большую угрозу незащищенному ресурсу. Корпоративный firewall уже не решает всех проблем безопасности. Ведь равный и обобщенный подход к обеспечению безопасности данных каждого сотрудника компании неизбежно приводит к наличию "брешей" в защите. Из-за этого страдают нужды того или иного работника, когда оказываются закрыты критичные для выполнения работы ресурсы. Более того, многие системы безопасности отделяют от общего доступа только жизненно важные данные (например, бухгалтерский учет), в то время как остальные сведения, которые считаются менее важными, доступны всем. Конечно, это не означает, что сотрудники соседнего подразделения изучают данные своих коллег. Но такая открытость делает данные всех работников уязвимыми к атаке через одну-единственную лазейку в сети. Поэтому вместо порчи данных на 1-2 компьютерах - уязвима вся система. Данную проблему выявил червь Code Red. Известно как минимум о двух базовых версиях сетевого червя Code Red. Первая была запущена в пятницу, 12 июля 2001 года. Она не использовала для распространения ни электронную почту, ни заражение файлов приложений. Инфицируя новый компьютер червь создавал 10000 клонов самого себя, каждый из которых начинал искать новые цели для распространения через уязвимости веб-сервера IIS компании Microsoft. Как оказалось, в логике работы червя было несколько серьёзных ошибок, которые послужили причиной запуска второй версии вируса. Она появилась утром в 10:00 19 июля 2001 года и к 14:00 успела заразить примерно 359 тысяч компьютеров. Именно эта версия попала на первые страницы средств массовой информации.

В 1985 году Стив Беллоуин (Steve Bellovin), член Совета по архитектуре интернета (Internet Architecture Board) опубликовал доклад об уязвимости TCP/IP-протокола. Натолкнул Стива на публикацию доклада - Кевин Митник. Кевин Митник- Знаковая фигура в сфере информационной безопасности. Консультант по компьютерной безопасности, писатель, бывший компьютерный хакер. В конце XX века был признан виновным в различных компьютерных и коммуникационных преступлениях. Известен по прозвищу Кондор, это прозвище Кевину досталось от главного персонажа одного американского фильма, где тот скрывался от ЦРУ, используя свои умения в работе с телефонной сетью.

Суть компьютерной безопасности не только в ее построении, но и в постоянном поиске уязвимости и устранении ее раньше, чем злоумышленники сумеют создать или найти "брешь" и запустить вирусную программу. Следовательно, безопасность - процесс

динамический, а не статический. Анализ и устранения рисков - его основная составляющая, которой нельзя пренебрегать.

Вирус Code Red вывел из строя именно те серверы, которые были защищены от элементарных атак из Сети интернет и не следили за своей растущей уязвимостью (ввиду отсутствия подобных прецедентов), так как прежде подобные атаки не проводились, никто о ней не думал. В итоге такая беспечность стоила миллионов долларов. Обобщим основные причины уязвимости веб-серверов:

1. Большинство растущих предприятий и компаний меняют конфигурацию своих сетей (добавление новых рабочих станций и серверов), но не производят тест ЛВС на безопасность. ЛВС (Локальная Вычислительная Сеть)- Компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий. Компьютеры могут соединяться между собой, используя различные среды доступа: медные проводники (витая пара), оптические проводники (оптические кабели) и через радиоканал (беспроводные технологии). Проводные, оптические связи устанавливаются через Ethernet и прочие средства. Отдельная локальная вычислительная сеть может иметь связь с другими локальными сетями через шлюзы, а также быть частью глобальной вычислительной сети (например, Интернет) или иметь подключение к ней. Вернемся к причинам. Разумеется, запретить подключать новых пользователей невозможно, но стоит задуматься о расширении сети заранее. Обозначить ее сегменты, которые способны к расширению, и проводить предварительное тестирование на безопасность.

2. Большинство веб-мастеров имеют корневой или администраторский доступ к серверу. Конечно будет надежнее прописать каждому сотруднику свою политику доступа, ограничивающую его сферу деятельности его же прямыми обязанностями. Например, сотрудник работает только с одним каталогом сервера, но имеет доступ на все остальные. Тем самым он ставит под угрозу не только свой сектор работ, но и все данные сервера. Конечно, статус веб-мастера имеет не каждый пользователь, хотя ограничить доступ из соображений безопасности следует и самым высоким профессионалам.

Следует перечислить основные механизмы и средства защиты ресурсов информационных систем: Идентификация и аутентификация, разграничение доступа, регистрация и аудит, контроль целостности, криптографические механизмы обеспечения конфиденциальности, целостности и аутентичности информации, контроль содержимого, обнаружение и противодействие атакам, анализ защищенности.

Безопасность Web-приложений уже не первый год является важным элементом защиты информационных систем. Учитывая тенденцию к переносу стандартных клиент-сервеных приложений в Web-среду, растущую популярность технологий AJAX и других

элементов Web 2.0 можно констатировать, что с течением времени актуальность защиты онлайн-приложений только растет. AJAX - Подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее. По-русски иногда произносится транслитом как «аякс». У аббревиатуры AJAX нет устоявшегося аналога на кириллице.

Ответить на вопрос о вероятности обнаружения той или иной проблемы можно с помощью информации из справочников (баз данных) уязвимостей. На сегодняшний день признанным отраслевым стандартом в этой области является список Common Vulnerabilities and Exposures (CVE). Однако непосредственно сам список слабо упорядочен и требует серьезной аналитической работы для получения полезных статистически результатов. CVE (англ. Common Vulnerabilities and Exposures) — база данных общеизвестных уязвимостей информационной безопасности. Каждой уязвимости присваивается идентификационный номер вида CVE-год-номер, описание и ряд общедоступных ссылок с описанием. Поддержкой CVE занимается организация MITRE. Финансированием проекта CVE занимается US-CERT. Проект CVE был официально запущен для общественности в сентябре 1999 года. В то время большинство инструментов информационной безопасности использовали свои собственные базы данных с их собственными именами для уязвимостей. Были значительные различия между продуктами и не было простого способа определить, когда разные базы данных ссылались на одну и ту же проблему. Последствиями были потенциальные пробелы в охвате безопасности и отсутствие совместимости между разрозненными базами данных и инструментами. Кроме того, поставщики инструментов по-разному считали количество уязвимостей, которые они обнаружили.

Ежегодно аналитиками Mitre проводится анализ информации в базе данных и публикуется отчет о распределении уязвимостей по различным критериям. Согласно отчету более четверти проблем, обнаруженных в 2006 году, приходится на недостатки безопасности Web-приложений. Аналогичную информацию публикуют и другие базы данных уязвимостей. По информации портала SecurityLab.ru, около 40% всех обнаруженных в 2007 уязвимостей приходится на Web-приложения.

Несмотря на то, что информация из баз данных уязвимостей достаточно интересна с теоретической точки зрения, она мало подходит для практического использования в рассматриваемом контексте. Это связанно с тем, что в базы данных попадает информация о широко распространенных приложениях, чье внедрение носит массовых характер. Что касается Web-приложений, то зачастую они создаются под конкретную задачу и могут быть

развернуты только в одной сети или в единственном экземпляре. В последнее время наметилась тенденция публикации в базах данных информации не только об популярных Web-приложениях, но и в часто используемых on-line сервисах. Например - в общедоступных системах электронной почты, поисковых системах, социальных сетях и т.д. Однако пока это больше исключение, чем правило. Таким образом, ответить на вопрос «Насколько вероятно обнаружение уязвимости в Web-приложении» с помощью баз данных уязвимостей невозможно.

Существуют и другие подходы, например, использование в качестве исходной информации результатов работ по анализу и оценке защищенности Web-приложений. Как правило, это метод используется консалтинговыми компаниями, имеющими большой опыт в области безопасности приложений.

В качестве примера подобных отчетов можно привести ежегодный отчет «Статистика уязвимости Web-приложений» компании Positive Technologies (PT) и ежеквартальное обозрение «Website Security Statistics Report» компании WhiteHat Security (WH). Оба отчета имеют сходную структуру и содержат статистику уязвимостей Web-приложении, полученную в ходе работ по тестированию на проникновение, аудита безопасности и др. Для получения данных использовались различные подходы, от сканирования Web-приложений с помощью сканеров с последующей проверкой результатов до тестирования методом "белого ящика", включающего также частичный анализ исходного кода.

Positive Technologies — международная компания, специализирующаяся на разработке программного обеспечения в области информационной безопасности. Предоставляет услуги в области анализа защищенности и управления соответствием.

Оба отчета используют классификацию уязвимостей Web Application Security Consortium Web Security Threat Classification. В этом документе собраны воедино и организованы различные угрозы безопасности Web-приложений. Проект является попыткой разработки и популяризации стандартной терминологии описания проблем безопасности в Web-приложениях. Распространенные уязвимости Web-приложений организованны в структурированный список, состоящий из шести классов, каждый из которых содержит несколько типов уязвимостей и атак. В настоящее время готовится к публикации вторая редакция классификации, содержащая девять классов угроз. Оба отчета используют элементы Web Security Threat Classification version 2 для описания таких проблем как «Расщепление HTTP-ответа» (HTTP Response Splitting) и «Подделка HTTP-запроса» (Cross-Site Request Forgery).

Наиболее распространенной уязвимостью высокой степенью риска обе компании признают «Внедрение операторов SQL» (SQL Injection), вероятность обнаружения которой

составляет 31% - PT и 16% - WH. Меньшая вероятность обнаружения проблем различных типов в трактовке WhiteHat Security вполне объяснима большей зрелостью западного рынка и использованием результатов повторных проверок приложений одного и того же клиента. К сожалению, большинство российских владельцев Web-приложений находится на этапе становления процесса управления информационной безопасностью.

Внедрение SQL-кода (англ. SQL injection) — один из распространённых способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода. Внедрение SQL, в зависимости от типа используемой СУБД и условий внедрения, может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере. Атака типа внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах. Разработчик прикладных программ, работающих с базами данных, должен знать о таких уязвимостях и принимать меры противодействия внедрению SQL.

В некоторых пунктах, например по количеству уязвимостей типа «Утечка информации» (Information Leakage) отчеты достаточно серьезно расходятся (90% - PT и 40% - WH). Это связанно с тем, что WhiteHat Security включает в отчет только уязвимости, имеющие «критичный», «неотложный» и «высокий» уровень риска, в то время как Positive Technologies задействует все найденные недочеты.

Вопрос классификации степени риска, связанного с уязвимостями приложений является важной темой. В настоящий момент существует множество методик оценки опасности уязвимости, но наиболее распространены следующие подходы:

- классическая «светофорная» оценка, выделяющая уязвимости «высокой», «средней» и «низкой» степени риска;

- пятиуровневая модель, принятая в стандарте PSI DSS и определяющая уровни «критичный», «неотложный», «высокий», «средний» и «низкий» (Urgent, Critical, High, Medium, Low)

- метод Common Vulnerability Scoring System (CVSS) ,оценивающий степень риска как число от 0 до 10.

Общая система оценки уязвимостей (CVSS) - это свободный и открытый отраслевой стандарт для оценки степени серьезности уязвимостей компьютерной системы. CVSS пытается присвоить уязвимостям баллы серьезности, позволяя ответчикам определять приоритеты ответов и ресурсов в соответствии с угрозой. Баллы рассчитываются по формуле, которая зависит от нескольких показателей, приближенных к простоте эксплойта и

его влиянию. Баллы варьируются от 0 до 10, причем 10 баллов - самые тяжелые. Хотя многие используют только базовый балл CVSS для определения степени серьезности, существуют также временные и экологические баллы, учитывающие доступность мер по смягчению последствий и степень распространенности уязвимых систем в организации, соответственно.

Не касаясь достоинств или недостатков каждого из методов можно выделить следующие особенности, которые могут влиять на достоверность оценки:

- зависимость от контекста;
- зависимость от конфигурации системы;
- зависимость от метода определения.

В различных приложениях уязвимости одного типа могут иметь различную степень риска. Так, уязвимость «Подделка HTTP-запроса» может не представлять угрозы для типичного репрезентативного сайта или поисковой машины, и наоборот - классифицироваться как проблема высокой степени риска в Web-интерфейсе электронной почты или платежной системы. В результате утечки информации злоумышленник может получить доступ к журналам работы приложения (низкая или средняя степень риска), а может загрузить резервную копию исходных текстов сайта (высокая степень риска).

Конфигурация конкретной системы также может оказывать серьезное влияние на степень риска. Так, уязвимость «Внедрение операторов SQL» обычно классифицируют как имеющую высокую опасность. Однако в случае если Web-приложение работает с сервером СУБД с ограниченными привилегиями, она может быть отнесена к проблемам средней или низкой степени риска. В другой инсталляции или реализации приложения эта же уязвимость может быть использована для получения доступа к операционной системе с правами суперпользователя, что естественно делает её наиболее критичной.

В зависимости от метода у глубины анализа степень одна и та же уязвимость может быть оценена по-разному. Если взять приведенный выше пример «Внедрения операторов SQL», использование сетевого сканера позволит только констатировать наличие проблемы. Для определения привилегий, доступных потенциальному злоумышленнику требуется либо попытаться использовать ошибку, либо уточнить порядок взаимодействия между Web-приложением и СУБД методом «белого ящика».

Тестирование белого ящика (англ. white-box testing), также тестирование стеклянного ящика (англ. glass-box testing), структурное тестирование (англ. structural testing) — тестирование, которое учитывает внутренние механизмы системы или компонента (ISO/IEC/IEEE 24765).

Обычно включает тестирование ветвей, маршрутов, операторов (см. покрытие кода). При тестировании выбирают входы для выполнения разных частей кода и определяют ожидаемые результаты. Это напоминает внутрисхемное тестирование (англ.).

Традиционно тестирование белого ящика выполняется на уровне модулей, однако оно используется для тестирования интеграции систем и системного тестирования, тестирования внутри устройства и путей между устройствами. Этот метод тестирования не может выявить невыполненные части спецификации, отсутствие требований или создание не того приложения.

Каждый из приведенных методов оценки использует свои подходы для учета обозначенных обстоятельств. Это может быть абстрактное «экспертное мнение» в «светофорной» оценке или весовые коэффициенты в CVSS, но в любом случае - метод и глубина анализа оказывают серьезное влияние на оценку рисков. Это обстоятельство необходимо принимать внимание при работе с отчетами и статистическими данными.

Для определения основных рисков можно следовать следующей цепочке: источник угрозы > фактор (уязвимость) > угроза (действие) > последствия (атака).

- Источник угрозы - это потенциальные антропогенные, техногенные или стихийные носители угрозы безопасности.

- Угроза (действие) - это возможная опасность (потенциальная или реально существующая) совершения какого-либо деяния (действия или бездействия), направленного против объекта защиты (информационных ресурсов), наносящего ущерб собственнику, владельцу или пользователю, проявляющегося в опасности искажения и потери информации.

- Фактор (уязвимость) - это присущие объекту информатизации причины, приводящие к нарушению безопасности информации на конкретном объекте и обусловленные недостатками процесса функционирования объекта информатизации, свойствами архитектуры автоматизированной системы, протоколами обмена и интерфейсами, применяемыми программным обеспечением и аппаратной платформой, условиями эксплуатации.

- Последствия (атака) - это возможные последствия реализации угрозы (возможные действия) при взаимодействии источника угрозы через имеющиеся факторы (уязвимости).

Антропогенными источниками угроз безопасности информации выступают субъекты, действия которых могут быть квалифицированы как умышленные или случайные преступления. Методы противодействия напрямую зависят от организаторов защиты информации.

В качестве антропогенного источника угроз можно рассматривать субъекта, имеющего доступ (санкционированный или несанкционированный) к работе со штатными

средствами защищаемого объекта. Или, простыми словами -- это либо хакер-злоумышленник или их группа, либо персонал компании, мотивированный теми или иными факторами на противоправные или противозаконные деяния. Субъекты (источники), действия которых могут привести к нарушению безопасности информации могут быть как внешние, так и внутренние.

Внешние источники могут быть случайными или преднамеренными и иметь разный уровень квалификации. К ним относятся:

- криминальные структуры;
- потенциальные преступники и хакеры;
- недобросовестные партнеры;
- технический персонал поставщиков телематических услуг;
- представители надзорных организаций и аварийных служб;
- представители силовых структур.

Внутренние субъекты (источники), как правило, представляют собой высококвалифицированных специалистов в области разработки и эксплуатации программного обеспечения и технических средств, знакомы со спецификой решаемых задач, структурой и основными функциями и принципами работы программно-аппаратных средств защиты информации, имеют возможность использования штатного оборудования и технических средств сети. К ним относятся: основной персонал (пользователи, программисты, разработчики); представители службы защиты информации.

Способы защиты сайта: защита от копирования; защита от вирусов; защита от скачивания.

Защита от копирования подразумевает запрет на какие-либо манипуляции с текстом первоисточника (сайта):

- запрет копирования. Можно использовать скрипты, запрещающие выделение и перетаскивание текста. Конечно, опытные пользователи без труда обойдут это ограничение, но вероятность копирования будет снижена. Скрипт (сценарий) — это последовательность действий, описанных с помощью скриптового языка программирования (JavaScript, PHP, Perl, Python и др.) для автоматического выполнения определенных задач;

- помещать название сайта в текст. Если человек бездумно скопирует и вставит информацию, то адрес или имя ресурса помогут определить первоисточник;

- лайки и ретвиты. Если пользователи активно сигнализируют о том, как им нравится страница сайта, поисковики не смогут пропустить это. Чем быстрее произойдёт индексация, тем выше вероятность определения сайта как первоисточника;

- указывать авторство. Можно связать свои статьи с профилем в Google+, тогда имя и фамилия укажут на то, кто был первым;

- предупреждать поисковик заранее. Яндекс.Вебмастер предоставляет очень удобный инструмент «Оригинальные тексты», с помощью которого можно заблаговременно уведомить поисковую систему о скором выходе нового материала. Получив и обработав текст статьи, роботы будут знать, кто автор и где искать первоисточник;

- кросспостинг. Если другие сайты трубят о том, что у вышла новая статья, поисковые роботы смогут гораздо быстрее об этом узнать.

Вредоносные программы способны нанести непоправимый ущерб сайту, который может перейти в денежный эквивалент. Поэтому стоит обращать должное внимание на пользователей посещающих сайт, а также следует постоянно искать «бреши» в защите и перманентно их нейтрализовывать:

- проверять антивирусом компьютеры людей, у которых имеется доступ к административной части сайта;

- стараться избегать использования Internet Explorer. Являясь наиболее популярным браузером, IE привлекает к себе пристальное внимание создателей вирусов;

- используя CMS, необходимо своевременно переходить на новейшие версии программного обеспечения. В системе управления содержимым могут находиться самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и так далее. Такая система часто используется для хранения, управления, пересмотра и публикации документации. Контроль версий является одной из важных возможностей, когда содержимое изменяется группой лиц;

- обновлять антивирус как можно чаще.

В некоторых случаях появляется нужда в проверке безопасности не только своего сайта, но и сторонних ресурсов. В этом вопросе сможет помочь сайт Antivirus-alarm.ru.

Не редки такие случаи, когда разработчик сайта не желает, чтобы с его ресурса часто скачивали информацию. Например, у Вас есть свой сайт, на котором вы публикуете обои для рабочего стола. Общий объем сайта — 500mb, посещаемость 7 000 хостов в сутки, примерный трафик — 300Гб в месяц или 10 Гб в день. Добавим к этим посетителям еще 20 человек, скачавших сайт целиком. Получаем увеличение трафика на 10Гб или в два раза. Или другими словами 0.28% посетителей создали 50% трафика. Не совсем честно, особенно если пользователь оплачивает трафик.

По этой причине существуют некоторые способы ограничения или запрета скачивания:

- user agent — так называются данные, которые каждый браузер передает серверу. Эти данные могут содержать в себе такую информацию, как тип браузера, операционная система, список плагинов и многое другое. Это наиболее простой, но наименее эффективный способ. Его преимущество в том, что ни кого лишнего вы не запретите, а недостаток в том, что практический каждый агент загрузки (Download Agent) может маскироваться под стандартные браузеры;

- ограничение по количеству просмотренных страниц за определенный промежуток времени. Достаточно спорный метод. Но надо понимать, что нормальный человек не может просмотреть 60 страниц за 1 минуту. Но с другой стороны и агент загрузки может делать паузы между скачиванием страниц. Даже если владелец сайта не заблокирует Агента загрузки совсем, то по крайней мере, сильно затрудните скачивание;

- запрет с помощью скрытой ссылки является одним из наиболее правильных и распространенных методов. Владелец сайта должен сделать скрытую ссылку на странице, по которой «живой» человек не перейдет, а агенты загрузки и прочие роботы сделают это. IP адрес с которого производится просмотр скрытой страницы блокируется, скажем, на 3 минуты.

Главный недостаток — это то, что при этом блокируются поисковые роботы. Бороться с этим можно двумя способами:

- Проверять $http_user_agent. Для этого необходимо будет знать то, каким образом подписываются все поисковые роботы. Кроме того, при таком способе агент загрузки сможет замаскироваться под поискового робота.

- Запрещать ip адрес можно не по факту загрузки скрытой страницы, а по факту загрузки картинки, установленной на скрытой странице. Поисковые роботы обычно не запрашивают изображения размещенные на страницах, а агенты загрузки обычно делают это.

В завершении следует сделать следующий вывод. Компьютерные технологии бурно развиваются в наше время. Интернет теперь не является простой функцией ПК. Теперь это отдельное пространство со своими законами, понятиями и отношениями. Иметь свой сайт и получать от него доход в наше время является деятельностью показывающей высокий статус владельца. Не каждый человек способен вести и обслуживать свой web-ресурс продолжительный период времени, но благодаря этой статье мы смогли разобраться в текущем положении дел, узнать какие опасности могут ждать наш ресурс, откуда их можно ждать и как их нейтрализовать и предотвратить следующие атаки.

Список литературы:

1. AJAX [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/AJAX (дата обращения: 26.01.20)

2. Code Red [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/Code_Red (дата обращения: 26.01.20)

3. Common Vulnerability Scoring System [Электронный ресурс] / URL: https://en.wikipedia.org/wiki/Common_Vulnerability_Scoring_System (дата обращения: 26.01.20)

4. База данных Common Vulnerabilities and Exposures [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures (дата обращения: 26.01.20)

5. Внедрение SQL-кода [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki (дата обращения: 26.01.20)

6. Защита Web-сайтов [Электронный ресурс] / URL: https://revolution.allbest.ru/programming/00655446_0.html (дата обращения: 26.01.20)

7. Защита сайта от скачивания [Электронный ресурс] / URL: https://yandex.ru/turbo?text=https%3A%2F%2Fwww.internet-technologies.ru%2Farticles%2Fzaschita-sayta-ot-skachivaniya.html (дата обращения: 26.01.20)

8. Кевин Митник [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/% (дата обращения: 26.01.20)

9. Компания Positive Technologies [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/Positive_Technologies (дата обращения: 26.01.20)

10. Локальная вычислительная сеть (ЛВС) [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/ (дата обращения: 26.01.20)

11. Система управления содержимым (СУС) [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki/ (дата обращения: 26.01.20)

12. Тестирование белого ящика [Электронный ресурс] / URL: https://ru.wikipedia.org/wiki (дата обращения: 26.01.20)