УДК 004.056.53

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ТЕСТИРОВАНИЯ УЯЗВИМОСТЕЙ *WEB*-РЕСУРСОВ

Голяткина Любовь Игоревна[1], Мельникова Ольга Игоревна[2]

[1]*Студент;*
ГБОУ ВО МО «Университет «Дубна»;
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: lubovgolyatkina@mail.ru.

[2]*Кандидат технических наук, доцент;*
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: oimelnik@mail.ru.

Работа посвящена вопросу разработки информационной системы тестирования уязвимостей web-ресурсов. Вследствие развития Интернета становится важным вопрос безопасности web-ресурсов, на которых концентрируется множество интеллектуальной, управленческой и конфиденциальной информации, требующей тщательной защиты.

Ключевые слова: система тестирования уязвимостей, критерии безопасности, уязвимости *web*-ресурсов.

Для цитирования:
Голяткина, Л. И., Мельникова, О. И. Разработка информационной системы тестирования уязвимостей web-ресурсов // Системный анализ в науке и образовании: сетевое на ное издание. – 2020. – № 3. – С. 23–30. – URL: http://sanse.ru/download/402.

DEVELOPMENT OF INFORMATION SYSTEM FOR TESTING VULNERABILITIES WEB-RESOURCES

Golyatkina Lubov[1], Melnikova Olga[2]

[1]*Student;*
Dubna State University;
Institute of the system analysis and management;
141980, Moscow region, Dubna, Universitetskaya str.,19;
e-mail: lubovgolyatkina@mail.ru.

[2]*Candidate of Science in Engineering, associate professor;*
Dubna State University,
Institute of the system analysis and management;
141980, Moscow region, Dubna, Universitetskaya str.,19;
e-mail: oimelnik@mail.ru.

The work is devoted to the development of an information system for testing web-resource vulnerabilities. Due to the development of the Internet, it becomes an important issue of security of web resources, on which a lot of intellectual, managerial and confidential information is concentrated, requiring careful protection.

Keywords: vulnerability testing system, security criteria, web-resource vulnerabilities.

For citation:
Golyatkina, L., Melnikova, O. Development of information system for testing vulnerabilities web-resources = Разработка информационной системы тестирования уязвимостей web-ресурсов // System Analysis in Science and Education. – 2020. – № 3. – Pp. 23–30. – URL: http://sanse.ru/download/402.

Введение

В современном мире Интернет выступает в роли мощного инструмента по поиску и предоставлению информации. На данный момент посредством Интернета можно совершать покупки, оплачивать коммунальные услуги, приобретать билеты, находиться в курсе событий и многое другое. Вопрос безопасности информационных ресурсов становится крайне актуальным, и в настоящее время он выделился в отдельное направление исследований и практических интересов. Были созданы международные стандарты, законы и целые направления в законодательствах, регулирующие аспекты информационной безопасности (ИБ) и обработку персональных данных. Информационной безопасностью называют меры по защите информации от несанкционированного доступа, разрушения, модификации, раскрытия системы данных. Она включает в себя меры по защите процессов создания данных, их ввода, обработки и вывода. Целью информационной безопасности является сбалансированная защита ценностей системы, защита и гарантированная точность и целостность информации, и минимизация разрушений, которые могут иметь место, если информация будет изменена или разрушена. Информационная безопасность требует учета всех событий, в ходе которых информация создается и модифицируется.

В наше время важно осуществлять контроль уязвимостей на *web*-ресурсах в связи с этим перекрываются каналы утечки конфиденциальной информации, выстраиваются перспективы перекрытия возможных хакерских атак и своевременно выявляются слабые места в безопасности сайта. Одним из элементов систем защиты является информационная система тестирования уязвимостей *web*–ресурсов. В большинстве случаев традиционные СТУ базируются на знаниях экспертов, в частности, на их знаниях угроз в отношении защищаемой системы. Администратор безопасности системы может обнаруживать вредоносное поведение в системе, за счёт использования специальных инструментов, которые позволяют постоянно контролировать состояние системы и сообщать о «нестандартных активностях».

В последнее время на *web*-ресурсах Интернета участились случаи хакинга *web*-ресурсов. *Web*-ресурсы становятся жертвами злонамеренных нападений, так как они содержат персональную информацию клиентов, включая имена, пароли и сведения о кредитных карточках. В процессе скрытой манипуляции нападающий может изменить поля для формы, связанные с персональными данными, что позволит ему получить доступ к конфиденциальной информации без аутентификации. Распространён такой вид атаки, как нападение за счёт переполнения буфера, а также атакующий может применять заражение *cookie*-файла. Хакер может найти всю критичную информацию в одном месте, которые передаются путём ввода данных в форму регистрации на *web*-ресурсе.

От большинства злоумышленных действий, большую часть из которых составляют удаленные вторжения, можно защититься путем правильного использования совокупности организационных и технических мер. На сегодняшний день системы тестирования уязвимостей являются важным элементом комплексной системы защиты *web*-ресурсов, как мелких, так и крупных организаций. Задачи систем тестирования уязвимостей направлены на выявление возможных проникновений в систему и занесении информации в «логи» о найденных уязвимостях.

Концепция систем тестирования уязвимостей

Тестирование уязвимостей – это имитация действий потенциального злоумышленника с целью оценки возможности несанкционированного доступа к информационной системе и демонстрации уязвимостей существующей системы информационной безопасности (ИБ). Тестирование уязвимостей позволяет выявить уязвимости и слабые места в системе ИБ до того, как это сделают злоумышленники, оценить «практическую» защищенность от атак из «реального мира».

Тестирование уязвимостей на базе технических методов проводится со стороны внешнего «злоумышленника» и предполагает использование метода «чёрного ящика»: имитация нарушителя, не

обладающего никакими сведениями об организации и доступом к ее сети. Основные этапы работ: начало работ, добавление базы решающих правил, тестирование уязвимостей, информирование о критичных выявлениях, предоставление результатов пользователю. Стандартная структура представлена на рисунке ниже (рис. 1).

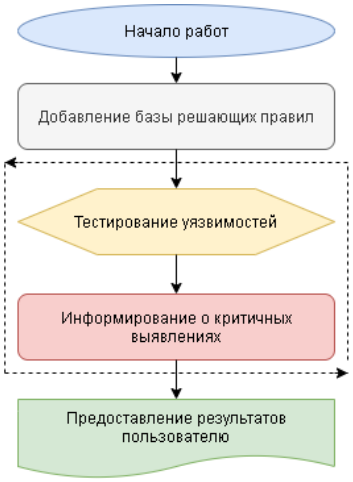

Рис. 1. Стандартная структура СТУ

С позиции потенциального злоумышленника осуществляются санкционированные попытки обойти существующие средства защиты, выявляются возможные сценарии проникновения в сеть и достижения целей тестирования (получение прав доступа, кража конфиденциальной информации, внесение изменений в информационные системы, нарушение работы отдельных компонентов сети и системы безопасности или бизнес-процессов).

На сегодняшний день широко применяются системы обнаружения вторжений. В данной работе предпринята попытка создать систему тестирования уязвимостей, используя подходы, которые используются в СОВ.

Система тестирования уязвимостей должна обеспечивать выявление фактов несанкционированных атак на *web*-ресурсы и фиксировать полученные данные, а также выдавать клиенту информацию о найденных уязвимостях, которая инициирует проведение мероприятий по обеспечению должной безопасности *web*-ресурса.

В результате разрабатываемая информационная система тестирования уязвимостей *web*-ресурсов приобрела следующие наиболее актуальные характеристики, присущие системам обнаружения вторжений:

- по методу обнаружения: поведенческая, так как СТУ использует информацию о нормальном поведении контролируемого *web*-ресурса;
- поведение после обнаружения: активное, система тестирования уязвимостей выявляет лазейки, которыми могут воспользоваться потенциальные нарушители;
- расположение источников результата аудита: «логи» (база данных);
- частота использования: непрерывный мониторинг.

Система тестирования уязвимостей классифицируется:

- по способу реагирования: статическая, так как СТУ проверяет *web*-ресурс на наличие известных уязвимостей;
- по способу сбора информации: системная, поскольку СТУ с высокой точностью может определить какая атака была совершена;
- подход к обнаружению основан на сигнатурном анализе обнаружения уязвимостей.

В качестве модели выявления атак в системе тестирования уязвимостей использована модель контекстного поиска.

В основу разработки информационной системы тестирования уязвимостей *web*-ресурсов взяты характеристики, классификации, методы и способы обнаружения, которые используются в системах

обнаружения вторжений. Система обнаружения вторжений (*IDS*) – специализированная система, используемая для идентификации того факта, что была предпринята попытка вторжения, вторжение происходит или произошло, а также для возможного реагирования на вторжение в информационные системы и сети [4]. Системами обнаружения вторжений называют множество различных программных и аппаратных средств, объединяемых одним общим свойством – они занимаются анализом *web*–ресурсов, обнаруживают подозрительные или просто нетипичные события, регистрируют информацию об этих событиях, предпринимают некоторые самостоятельные действия идентификации их причин.

В системах обнаружения вторжений выделяют несколько методов обнаружения атак. Поведенческая система обнаружения вторжений использует информацию о нормальном поведении контролируемой системы. Интеллектуальная система обнаружения вторжений использует информацию об атаках. Также системы обнаружения вторжений могут быть активными – это означает, что система обнаружения вторжений выявляет лазейки, которыми могут воспользоваться потенциальные нарушители и устраняет их. Пассивные системы выдают предупреждения: информацию об уязвимости и совершённой атаке. Расположение источников результата аудита подразделяет *IDS* в зависимости от вида исходной информации, которую они анализируют. Входными данными для них могут быть регистрационные файлы или сетевые пакеты.

Системы обнаружения вторжений могут быть классифицированы различными способами. Классификация основывается на источнике данных, поведении системы, архитектуре, способов защиты системы и обнаружения атаки. По способам реагирования различают статические и динамические системы обнаружения вторжений. Статические средства осуществляют анализ среды, поиск ошибок в конфигурациях, а Динамические системы обнаружения вторжений осуществляют мониторинг в реальном времени всех действий, происходящих в системе, просматривая файлы аудита или сетевые пакеты, передаваемые за определённый промежуток времени. Они реализуют анализ в реальном времени и позволяют постоянно следить за безопасностью системы. По способу сбора информации различают сетевые и системные системы обнаружения вторжений. Сетевая система обнаружения вторжений может запускаться либо на отдельном компьютере, который контролирует свой собственный трафик, либо на выделенном компьютере, прозрачно просматривающим весь трафик в сети. Системы обнаружения вторжений, которые устанавливаются на хосте и обнаруживают злонамеренные действия на нём называются хостовыми или системными [6].

По способу выявления атаки системы обнаружения вторжений принято делить на две категории: обнаружение аномального поведения (*anomaly-base*) и сигнатурный анализ (*signature-base*). Метод обнаружения вторжений на основе аномалий базируется на выявлении трафика, который сильно отличается от нормального поведения системы. Примером аномального поведения может служить большое число соединений за короткий промежуток времени. Однако аномальное поведение не всегда является атакой.

В основу сигнатурного анализа входит описание атаки в виде сигнатуры (*signature*) и поиска данной сигнатуры в контролируемом пространстве (*web*-ресурсе).

Сигнатурный анализ – это способ обнаружения вредоносной активности в сети, основанный на реализации процесса сравнения значения хеш–функции проверяемого объекта с базой известных вредоносных сигнатур. После идентификации угрозы она фиксируется и пользователю передается тревожное сообщение [5].

Анализ сигнатур был первым методом, примененным для обнаружения вторжений. Во входящем пакете просматривается байт за байтом и сравнивается с сигнатурой (подписью) – характерной строкой программы, указывающей на характеристику вредного трафика. Такая подпись может содержать ключевую фразу или команду, которая связана с атакой. Если совпадение найдено, объявляется тревога. В качестве сигнатуры атаки может выступать шаблон действий или строка символов, характеризующие аномальную деятельность. Эти сигнатуры хранятся в БД, аналогичной той, которая используется в антивирусных системах. Данная технология обнаружения атак очень похожа на технологию обнаружения вирусов, при этом система может обнаружить все известные атаки. Однако системы данного типа не могут обнаруживать новые, еще неизвестные виды атак [6].

Эти системы превосходят все другие при отлове хакеров на первичном этапе: простые атаки имеют привычку использовать некие предварительные действия, которые легко распознать. Также анализ, основанный на сигнатуре, точно и быстро сообщает, что в системе все нормально, если это

действительно так. Наличие в системе обнаружения сигнатур известных атак даёт высокий процент обнаружения вторжений.

Сигнатурой атаки называют характерные признаки компьютерного вируса, используемые для их обнаружения. В качестве сигнатуры атаки могут выступать: строка символов, семантическое выражение на специальном языке, формальная математическая модель. Выделением сигнатур занимаются эксперты в области компьютерной вирусологии, которые способны выделить код вируса из кода программы и сформулировать его характерные свойства в наиболее удобной для поиска форме.

Алгоритм работы сигнатурного метода основан на поиске сигнатур атак в исходных данных, собранных сетевыми и хостовыми датчиками системы. При обнаружении искомой сигнатуры система тестирования уязвимостей фиксирует факт информационной атаки, которая соответствует найденной сигнатуре. Количество сигнатур не равно количеству обнаруживаемых вирусов, так как часто для обнаружения семейства похожих вирусов используется одна и та же сигнатура.

Наиболее распространенной сигнатурной моделью процесса выявления атак является модель контекстного поиска определенного множества символов в исходных данных. Контекстный поиск информации в настоящее время активно используется различными подразделениями правоохранительных органов в целях выявления, расследования и профилактики преступлений. В качестве исходных данных выступает информация, накопленная в «логах». Для определения множества символов, поиск которых должен быть проведен в исходных данных, используются регулярные выражения: [], ^, ·, *, +, $.

При помощи регулярных выражений можно производить контекстный поиск любой сложности, что даёт возможность эффективно выявлять известные типы атак, сопоставляя каждую из них определённой сигнатуре. Пример сигнатур атак, которая используются в системе тестирования уязвимостей: (?:"[^"]*[^-]?>)|(?:[^\w\s]\s*\/>)|(?:>") .

Разработка структуры и подходов в предлагаемой системе тестирования уязвимостей

Основные компоненты архитектуры разрабатываемой системы тестирования уязвимостей представлены на рисунке (рис. 2).

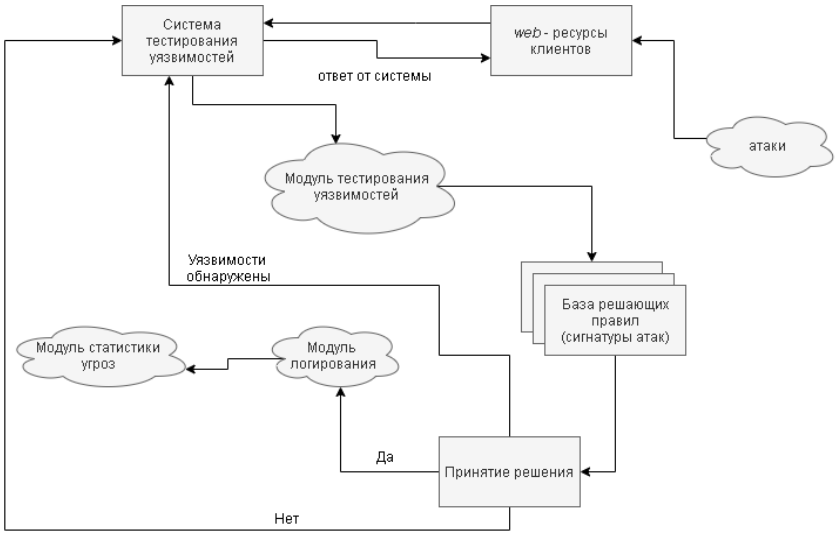

Рис. 2. Архитектура системы тестирования уязвимостей

Система тестирования уязвимостей выявляет распространённые типы атак:

XSS (*Cross–Site Scripting*, «межсайтовый скриптинг; *CSRF* (*Cross Site Request Forgery*, «межсайтовая подделка запроса); *LFI* (*Local File Inclusion*, «включение локальных файлов»); *SQLI* (*SQL injection*, внедрение *SQL*-кода; *Format–string* («форматирование строки»); *DoS* (*Denial of Service*, «отказ в обслуживании»).

Система тестирования уязвимостей содержит базу данных, в которой находится 78 правил для определения различных типов уязвимостей. Администратор системы тестирования уязвимости может добавлять новые правила в систему, тем самым обеспечивая актуальный поиск уязвимостей в *web*–ресурсах.

Структурная схема взаимодействия клиентской и серверной части представлена на рисунке ниже (рис. 3). «Клиентская часть» (браузер) посылает запрос на *web*-сервер в «серверной части», далее происходит взаимодействие с *php*-интерпретатором. Затем происходит обращение к СУБД (система управления базами данных) и к БД системы тестирования уязвимостей.

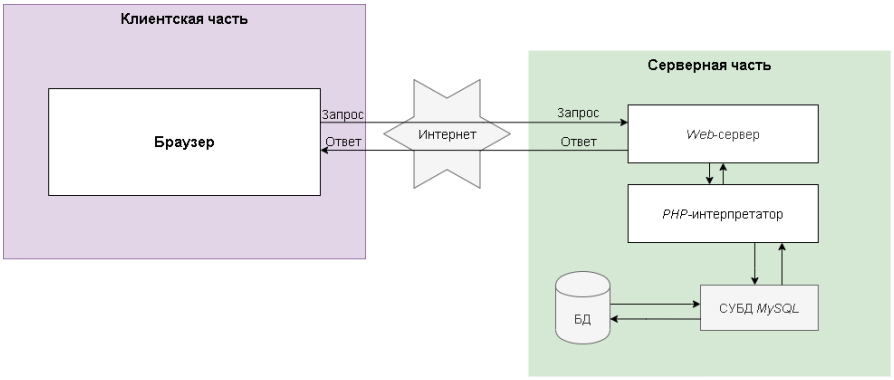

Рис. 3. Структурная схема взаимодействия клиентской и серверной части в СТУ

Клиентская часть приложения написана на языках *html, css, javascript,* с применением библиотек: *jQuery, bootstrap, apexcharts.* Серверная часть написана на *php*.

Если пользователь зарегистрирован в системе, то он может воспользоваться функциями системы в пределах своей роли. Если пользователь не зарегистрирован, то он может зарегистрироваться и авторизоваться, иначе пользователь может воспользоваться только авторизацией. Сигнатурный метод анализа уязвимостей основан на поиске сигнатур атак в исходных данных, собранных системой тестирования уязвимостей. При обнаружении искомой сигнатуры СТУ уведомляет о факте атаки, которая соответствует найденной сигнатуре. Система выделяет одну угрозу, соответствующую нескольким правилам, чтобы при одинаковых тегах угрозы не повторялись. Логи – это данные, которые система получает при проведении тестирования уязвимостей. Администратор может ознакомиться с полями формы, нажав на «Просмотреть значения». Администратор системы может просматривать, кто зарегистрирован в системе тестирования уязвимостей, а также логин и роль пользователя. Для того, чтобы протестировать уязвимости, необходимо ввести запрос или выбрать одну угрозу из списка предложенных угроз. Также можно выбрать язык, на котором будут транслироваться обнаруженные уязвимости. Пользователи системы могут просматривать статистику угроз по сайтам. Клиент видит статистику только по своим сайтам, администратор системы тестирования уязвимостей видит статистику по всем сайтам, включая сайты клиентов (рис. 4).

Названия угрозы	Количество появления
Detects common comment types	746
Detects possible includes and typical script methods	38
Detects very basic XSS probings	39
Detects obfuscated script tags and XML wrapped HTML	84
Detects possibly malicious html elements including some attributes	84
Detects JavaScript location/document property access and window access obfuscation	48

Рис. 4. Статистика угроз по сайтам

Для защиты сайта клиента в активном режиме написан сценарий, который после передачи входных данных с сайта клиента в систему тестирования уязвимостей, при случаях обнаружения потен-

циальных угроз может избавить сайт клиента от уязвимых полей. Основные требования безопасности информационной системы тестирования уязвимостей *web*–ресурсов определены по ГОСТ Р ИСО/МЭК 15408 [1, 2, 3].

После окончания тестирования осуществляется: выработка конкретных рекомендаций по устранению выявленных недостатков и повышению уровня защищенности *web*-ресурса организации; консультации и участие специалистов по устранению выявленных уязвимостей и недостатков; проведение последующих тестов и испытаний (после устранения уязвимостей и недостатков) для подтверждения эффективности реализованных мер.

Апробация

Работа системы тестирования уязвимостей была апробирована на *web*-ресурсах: обучающей платформы 25 *Minute Languages* [7] и *web*-ресурсе Государственного университета «Дубна» [8].

На сегодняшний день многие компании имеют свои *web*–ресурсы в сети Интернет. Одним из таких является обучающая платформа 25 *Minute Languages*. В функционал *web*-ресурса входит информация об организации, описание преимуществ при выборе данного *web*-ресурса для обучения, предоставляется выбор языка обучения, отзывы клиентов 25 *Minute Languages,* преподаватели, раздел регистрации, содержащий информацию о персональных данных, которая согласно законодательству Российской Федерации, подлежит защите от разглашения (Федеральный закон № 152). Данной компании необходимо соблюдать информационную безопасность в связи с тем, что на её *web*-ресурсе сконцентрированы персональные данные клиентов, а также конфиденциальная информация.

Информационная безопасность становится все более востребованной, так как разработчики зачастую игнорируют аспекты безопасности, удешевляя проекты при разработке и в итоге это влечёт за собой массу проблем (например, если злоумышленник получит доступ к персональным данным клиентов, то при публикации этих данных, компания может потерять доверие к себе).

В качестве примера продемонстрировано тестирование *XSS* уязвимости на форме авторизации *web*-ресурса обучающей платформы 25 *Minute Languages*. Для проведения тестирования необходимо выбрать предложенный системой тестирования уязвимостей *XSS* запрос *<select>login* + ‘-‘*</select>* и разместить его в поле «пароль» формы авторизации на обучающей платформе 25 *Minute Languages*. В системе тестирования уязвимостей были отражены найденные уязвимости (рис. 5).

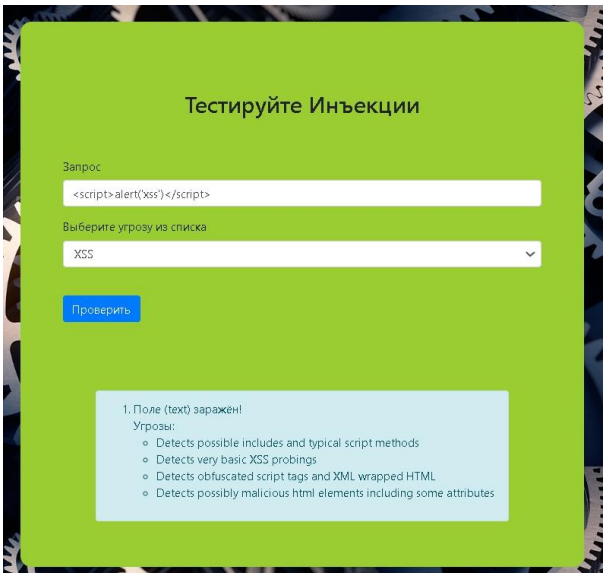

Рис. 5. Результат найденных уязвимостей

Разработанная информационная система тестирования уязвимостей также была апробирована на *web*-ресурсе Государственного университета «Дубна». Данный *web*-ресурс содержит информацию о конфиденциальных персональных данных, логинах и паролях. При тестировании этого *web*-ресурса было обнаружено, что ресурс имеет достаточный уровень защищённости. Во время эксплуатации инъекций было получено уведомление «Уязвимости не найдены». В связи с тем, что хостинг, на

котором размещена информационная система тестирования уязвимостей *web*-ресурсов не имеет *ssl*-сертификата, *web*-ресурс Государственного университета «Дубна» блокировал скрипт с результатами тестирования.

Заключение

Разработанная информационная система тестирования уязвимостей позволяет в реальном времени определять уязвимости *web*-ресурсов, выявлять лазейки, которыми могут воспользоваться потенциальные нарушители и систематизировать полученные данные в «логах». Наличие подобной информации позволяет повысить уровень безопасности организаций и предприятий. Система тестирования уязвимостей обладает высоким быстродействием и 100% обнаружением уязвимостей при тестировании системы на определённых *web*-ресурсах, расположенных в сети Интернет. СТУ успешно размещена на хостинге *SmartApe*, который соответствует федеральному закону №152 и готова к использованию для решения основных поставленных задач. Система тестирования уязвимостей представляет собой развивающийся ресурс, в котором можно расширять базу решающих правил, для нахождения распространённых типов атак.

Список литературы

1. ГОСТ Р ИСО/МЭК 15408-1-2008. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий». Часть 1. Введение и общая модель : дата введения 01.10.2009 / Федер. агентство по техн. регулированию и метрологии. – М. : Стандартинформ, 2009. – 36 с.
2. ГОСТ Р ИСО/МЭК 15408-2-2013. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий». Часть 2. Функциональные требования безопасности : дата введения 01.09.2014 / Федер. агентство по техн. регулированию и метрологии. – М. : Стандартинформ, 2014. – 156 с.
3. ГОСТ Р ИСО/МЭК 15408-3-2013. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий». Часть 3. Компоненты доверия к безопасности : дата введения 01.09.2014 / Федер. агентство по техн. регулированию и метрологии. – М. : Стандартинформ, 2014. – 145 с.
4. ГОСТ Р ИСО/МЭК 27033-1-2011. Информационная технология (ИТ). Методы и средства обеспечения безопасности. Безопасность сетей. Часть 1. Обзор и концепции. Термины и определения : дата введения – 01.01.2012 / Федер. агентство по техн. регулированию и метрологии. – М. : Стандартинформ, 2012. – 67 с.
5. Коноваленко, С. А. Сравнительный анализ функциональных возможностей программно-аппаратных систем обнаружения компьютерных атак / С.А. Коноваленко, С.А. Беседин, А.А. Соновский. – 2020. – URL : https://moluch.ru/conf/stud/archive/363/15679/ (дата обращения: 14.05.2020).
6. Шаньгин В. Ф. Информационная безопасность компьютерных систем и сетей: учебное пособие. – М. : ФОРУМ, 2017. – 416 с.
7. Обучающая платформа 25 Minute Languages : сайт. – URL : http://speak25.com/ (дата обращения: 31.05.2020).
8. Государственный университет «Дубна» : сайт. – URL : https://www.uni-dubna.ru/ (дата обращения: 08.06.2020).