

УЯЗВИМОСТИ ВЕБ-ПРИЛОЖЕНИЙ

2017

Оглавление

Введение	3
1. Методика	3
2. Резюме	4
3. Портрет участников исследования	5
4. Тенденции	6
5. Ручной анализ защищенности веб-приложений	7
5.1. Наиболее распространенные уязвимости	8
5.2. Анализ угроз и уровней защищенности	10
5.3. Статистика для различных отраслей экономики	13
5.4. Анализ различных средств разработки	16
5.5. Сравнение тестовых и продуктивных систем	19
5.6. Сравнение методов тестирования	20
6. Автоматизированный анализ защищенности	22
Заключение	25

ВВЕДЕНИЕ

Сфера применения веб-технологий расширяется из года в год. Практически каждая компания использует в своей деятельности веб-приложения — как для работы с клиентами, так и для обеспечения внутренних бизнес-процессов. И если функциональности веб-приложений уделяется значительное внимание, то вопросы их безопасности зачастую решаются в последнюю очередь, что негативным образом сказывается на уровне защищенности всего предприятия.

Уязвимости веб-приложений предоставляют злоумышленникам широкий простор для действий. Ошибки проектирования и администрирования позволяют атакующим получать важную информацию, а также нарушать функционирование веб-приложения, осуществлять атаки на отказ в обслуживании, проводить атаки на пользователей, проникать во внутреннюю сеть компании и получать доступ к критически значимым ресурсам.

В данном отчете представлена статистика, полученная экспертами Positive Technologies в ходе работ по анализу защищенности веб-приложений в 2016 году, а также ее сравнение с результатами предыдущих лет.

Представленное исследование позволяет понять, на какие недостатки защиты следует обратить внимание при разработке и эксплуатации приложений, какие угрозы несут в себе те или иные уязвимости и какие методы исследования безопасности наиболее эффективны. Также приводится оценка общего уровня защищенности веб-приложений и его динамика в последние годы.

1. МЕТОДИКА

В данном отчете рассмотрены результаты исследований 73 веб-приложений, для которых проводился углубленный анализ защищенности с наиболее полным покрытием проверок. В статистику вошли не только внешние веб-приложения, доступные из сети Интернет, но и предназначенные для внутреннего пользования. В данном исследовании не учитываются уязвимости, обнаруженные в ходе работ по тестированию на проникновение, инструментальному сканированию и анализу систем ДБО: информация о них представлена в отдельных аналитических отчетах¹.

Оценка защищенности проводилась как ручным способом методами черного, серого и белого ящика с использованием вспомогательных автоматизированных средств, так и в автоматизированном режиме с применением анализатора исходных кодов. Метод черного ящика заключается в проведении работ по оценке защищенности информационной системы от лица внешнего атакующего без предварительного получения какой-либо дополнительной информации о системе со стороны владельца. Метод серого ящика аналогичен методу черного ящика, при этом в качестве нарушителя в данном случае рассматривается пользователь, обладающий определенными привилегиями в системе. Метод белого ящика заключается в том, что для оценки защищенности информационной системы используются все необходимые данные о ней, включая исходный код приложений. Результаты ручного анализа защищенности представлены в первом разделе данного отчета, а результаты автоматизированного анализа — во втором.

Обнаруженные уязвимости классифицировались согласно соответствующим угрозам по системе Web Application Security Consortium Threat Classification (WASC TC v. 2), за исключением категорий Improper Input Handling и Improper Output Handling, поскольку они реализуются в рамках множества других атак. Кроме того, дополнительно мы выделили категории Insecure Session, Server Side Request Forgery и Clickjacking. Эти категории отсутствуют в классификации WASC, однако достаточно часто встречаются в исследуемых системах.

В категорию Insecure Session мы относим недостатки защиты сессии, например такие, как отсутствие флагов Secure и HttpOnly. Эти недостатки позволяют злоумышленнику перехватить значения Cookie пользователя при реализации различных атак.

¹ www.ptsecurity.com/ww-en/analytics/

Server-Side Request Forgery (подделка запроса со стороны сервера) — уязвимость, позволяющая выполнять произвольные HTTP-запросы от имени системы. Приложение, получив URL-адрес или HTTP-сообщение, осуществляет недостаточную проверку адреса назначения перед отправкой запроса. Используя данный недостаток, злоумышленник может отправлять запросы на серверы с ограниченным доступом (например, компьютеры в локальной сети), что приводит к разглашению важных данных, получению злоумышленником исходных кодов приложения, к отказу в обслуживании и т. п. В результате эксплуатации этой уязвимости злоумышленник может получить информацию о структуре сегментов сети, недоступных внешнему пользователю, обращаться к внутренним ресурсам, производить сканирование портов (сервисов) и т. п.

Clickjacking — разновидность атак на пользователей, заключающаяся в визуальном обмане. Ее принцип основан на том, что уязвимое приложение загружается во фрейме на страницу приложения, после чего маскируется под кнопку или какой-либо другой элемент. При клике по данному элементу пользователь выполняет задуманное злоумышленником действие в контексте уязвимого сайта. Уязвимость, позволяющая проводить эту атаку, возникает, когда приложение не возвращает специальный заголовок X-Frame-Options и тем самым разрешает показывать его во фреймах. Также в некоторых браузерах эта уязвимость может позволить выполнить атаку «Межсайтовое выполнение сценариев».

В настоящем отчете приведены только уязвимости, связанные с ошибками в коде и конфигурации веб-приложений. Другие распространенные проблемы информационной безопасности (к примеру, недостатки процесса управления обновлениями ПО) не рассматриваются.

Степень риска уязвимостей оценивалась согласно системе Common Vulnerability Scoring System (CVSS v. 3); на основе этой оценки выделялись качественные оценки высокого, среднего и низкого уровней риска.

2. РЕЗЮМЕ

Все веб-приложения уязвимы

Во всех исследованных веб-приложениях были обнаружены те или иные недостатки защищенности. При этом в 58% приложений обнаружены критически опасные уязвимости. В то же время видны и позитивные тенденции: общая доля сайтов, содержащих критически опасные уязвимости, уменьшилась по сравнению с 2015 годом на 12%.

Под угрозой пользователи

Практически все веб-приложения позволяют осуществлять атаки на пользователей. Кроме того, ряд веб-приложений осуществляет недостаточную защиту пользовательских данных. Так, доступ к персональным данным пользователей был получен в 20% приложений, которые обрабатывают пользовательскую информацию, включая сайты банков и государственных организаций.

Утечки — по-прежнему актуальная проблема

Приблизительно в каждой второй системе выявлена утечка важных данных, в том числе исходного кода и персональных данных пользователей. Раскрывают информацию о версии используемого ПО 63% веб-приложений.

Уязвимости веб-приложений — простой вектор проникновения в ЛВС

Каждое четвертое веб-приложение позволяет проводить атаки на ресурсы внутренней сети. Например, злоумышленник может получить доступ к файлам, сканировать устройства внутренней сети или проводить атаки на ресурсы сети. Также каждое четвертое веб-приложение содержит критически опасную уязвимость «Внедрение операторов SQL» и позволяет получить доступ к базе данных. Кроме того, эта уязвимость может дать возможность злоумышленнику прочитать произвольные файлы или создать новые, а также проводить атаки на отказ в обслуживании.

Промышленные компании наиболее уязвимы

Почти половина веб-приложений промышленных компаний характеризуется крайне низким уровнем защищенности. Во всех отраслях экономики, кроме финансовой, преобладают приложения, подверженные уязвимостям высокого уровня риска. В финансовой сфере 38% приложений содержат критически опасные уязвимости.

64% веб-приложений на основе ASP.NET содержат критически опасные уязвимости

Приблизительно каждое второе веб-приложение, созданное на базе языков PHP и Java, также содержит критически опасные уязвимости. При этом на одно веб-приложение, разработанное на языке PHP, приходится максимальное количество таких уязвимостей — 2,8 на систему.

Продуктивные системы более уязвимы, чем тестовые

Продуктивные системы в 2016 году являются менее защищенными. При ручном анализе критически опасные уязвимости выявляются в 50% тестовых систем и в 55% продуктивных систем. Уязвимостей высокого и среднего уровня риска в одной продуктивной системе выявлено в среднем в два раза больше, чем в одной тестовой.

Анализ исходного кода эффективнее «черного ящика»

В рамках ручного анализа доступ к исходному коду позволял выявить критически опасные уязвимости в 75% приложений, а при исследовании методом черного ящика критически опасные уязвимости были обнаружены в 49% систем.

Автоматизированный анализ выявляет уязвимости в кратчайшее время

В среднем на одно приложение автоматизированный анализатор исходного кода позволил выявить 4,6 уязвимости высокого уровня риска, 66,9 уязвимости среднего уровня риска и 45,9 уязвимости низкого уровня риска. Анализ исходного кода автоматизированными средствами показывает высокую эффективность, а скорость работы превосходит возможности ручного тестирования.

3. ПОРТРЕТ УЧАСТНИКОВ ИССЛЕДОВАНИЯ

Рассматриваемые веб-приложения принадлежат компаниям, относящимся к различным сферам деятельности — финансовым организациям, государственным учреждениям, СМИ, телекоммуникационным и промышленным компаниям; в отдельную категорию выделены интернет-магазины различной направленности.

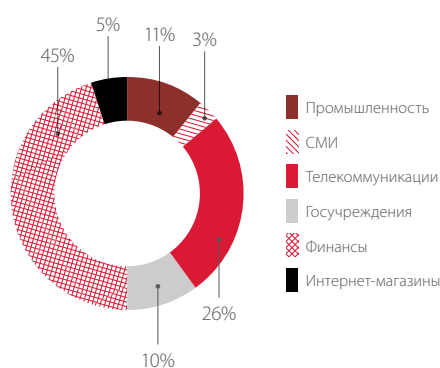


Рисунок 1. Портрет участников исследования

Среди исследуемых приложений преобладали продуктивные, т. е. внедренные в эксплуатацию и доступные для пользования, они составили почти две трети от общего количества систем.

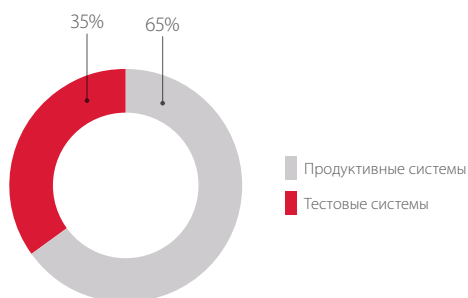


Рисунок 2. Доли продуктивных и тестовых систем

В этом году среди средств разработки веб-приложений преобладают PHP, Java, по сравнению с прошлым годом значительно увеличилась доля приложений, созданных на основе технологии ASP.NET. Иные средства разработки (Ruby, Python) объединены в категорию «Другие» и составляют всего 7%.

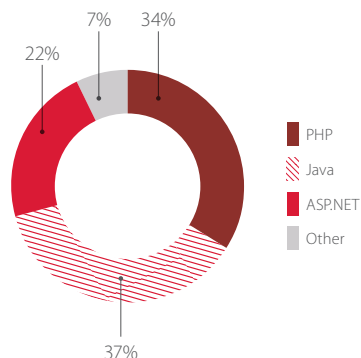


Рисунок 3. Средства разработки веб-приложений

4. ТЕНДЕНЦИИ

Все веб-приложения, исследованные в рамках анализа защищенности ручными методами или с использованием автоматизированных средств, были подвержены уязвимостям той или иной степени опасности. При этом всего в 1% приложений были выявлены уязвимости только низкой степени опасности. Позитивные изменения состоят в значительном сокращении (с 70 до 58%) доли веб-приложений, содержащих уязвимости высокого уровня риска. Частично на статистические данные повлиял тот факт, что компании, которые в прошлом году проводили анализ защищенности, учли результаты проверок при разработке новых веб-приложений. Причем в первую очередь внимание уделялось устранению критически опасных уязвимостей.

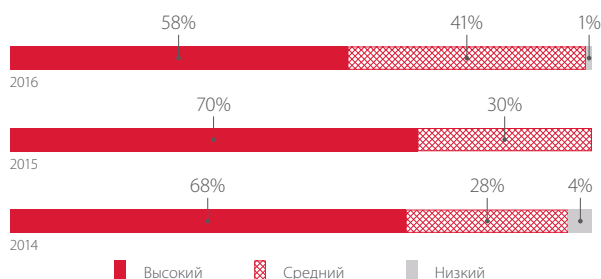


Рисунок 4. Доля уязвимых сайтов по максимальной степени риска уязвимостей

В целом негативная тенденция наблюдалась в течение предыдущих трех периодов исследований, и если в 2015 году мы наблюдали только замедление роста числа приложений, подверженных уязвимостям высокого уровня риска, то в 2016 году произошел спад. Однако стоит обратить внимание, что критически опасные ошибки все еще выявляются более чем в половине приложений, для которых проводится анализ защищенности.

Практически во всех исследованных приложениях были обнаружены уязвимости среднего уровня риска. Это число незначительно колеблется каждый год в пределах 90—100%. Возросла доля веб-приложений, подверженных уязвимостям низкого уровня риска.

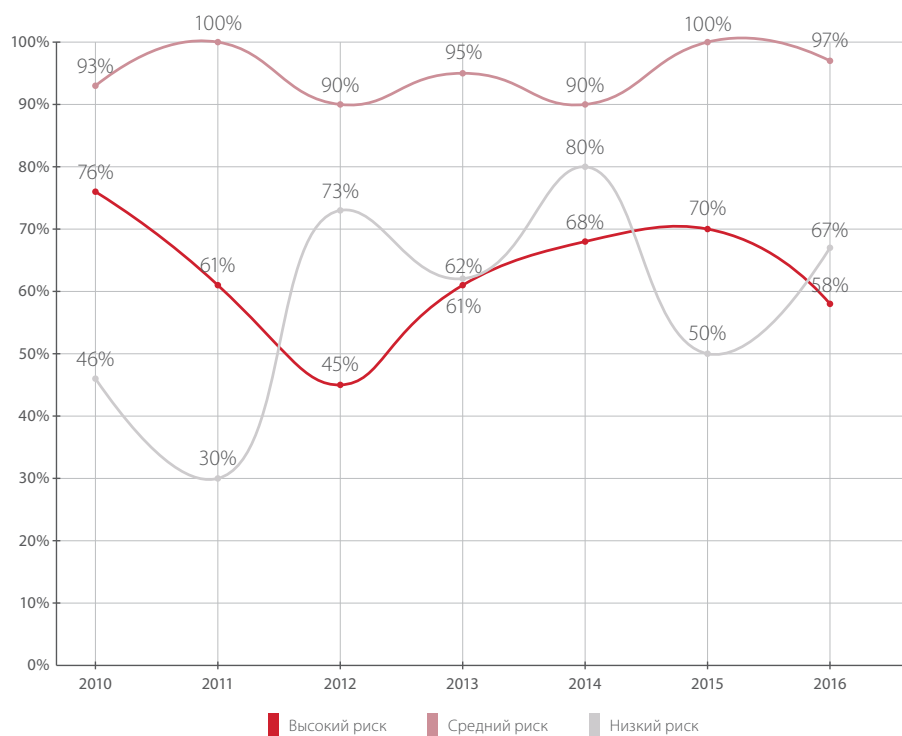


Рисунок 5. Доли сайтов с уязвимостями различной степени риска

5. РУЧНОЙ АНАЛИЗ ЗАЩИЩЕННОСТИ ВЕБ-ПРИЛОЖЕНИЙ

Среди всех уязвимостей, выявленных в рамках ручного анализа веб-приложений, одну десятую часть составляют уязвимости высокого уровня риска, а большая часть (81%) относится к среднему уровню опасности. По сравнению с прошлым годом доля критически опасных уязвимостей существенно сократилась, однако это связано преимущественно с тем, что в 2016 году на одну систему выявлено значительно больше уязвимостей среднего уровня риска.

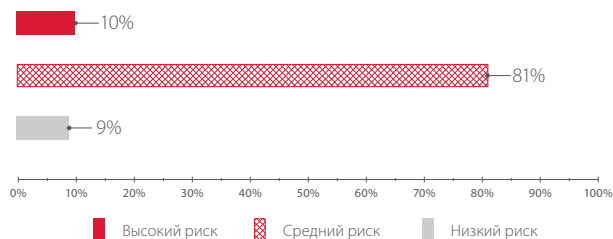


Рисунок 6. Доля уязвимостей различной степени риска (ручное тестирование)

Недостатки безопасности найдены во всех исследованных веб-приложениях. Критически опасные уязвимости обнаружены более чем в половине приложений, для которых проводились ручные проверки (54%), 44% приложений подвержены уязвимостям среднего уровня риска и ниже, и лишь в 2% приложений были выявлены уязвимости только низкой степени опасности.

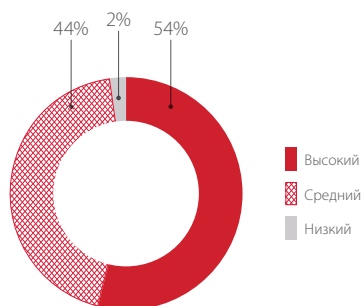


Рисунок 7. Распределение систем по максимальной степени риска обнаруженных уязвимостей (ручное тестирование)

В среднем на одну систему при ручном анализе защищенности выявлено 17 уязвимостей среднего уровня риска и по две уязвимости высокого и низкого уровня риска.

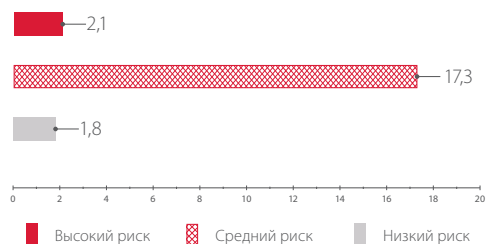


Рисунок 8. Среднее число уязвимостей на одну систему (ручное тестирование)

5.1. НАИБОЛЕЕ РАСПРОСТРАНЕННЫЕ УЯЗВИМОСТИ

В 2016 году половина уязвимостей, вошедших в десятку самых распространенных, позволяет совершать атаки на пользователей веб-приложений.

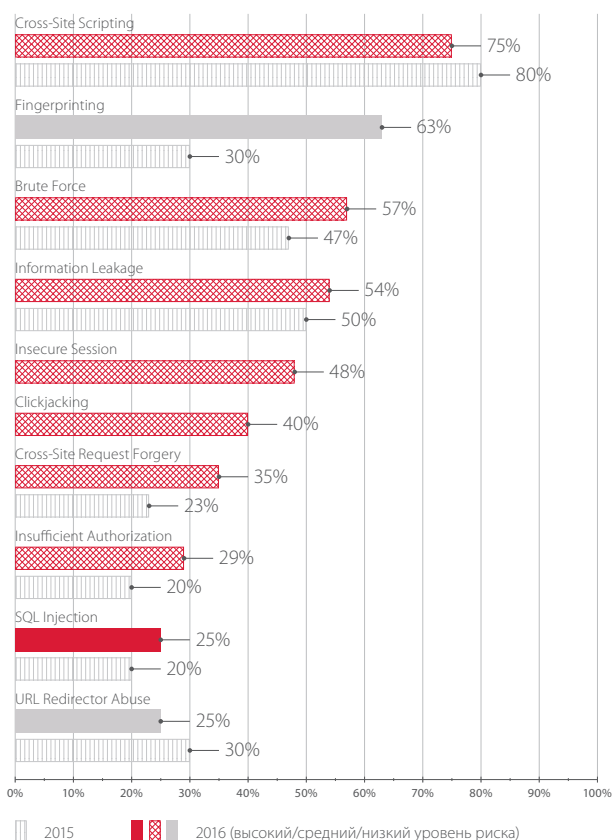


Рисунок 9. Наиболее популярные уязвимости, выявленные ручным тестированием (доля систем)

Как и в прошлом году, на первой строчке рейтинга находится уязвимость среднего уровня риска «Межсайтовое выполнение сценариев» (Cross-Site Scripting), которая встречается в 75% исследованных систем. В результате эксплуатации данной уязвимости злоумышленник может внедрить в браузер пользователя произвольные HTML-теги, включая сценарии на языке JavaScript и других языках, и таким образом получить значение идентификатора сессии атакуемого и совершить иные неправомерные действия, например фишинговые атаки.

Эксперты Positive Technologies в прошедшем году проводили также исследования атак злоумышленников на веб-приложения, в том числе составили рейтинг наиболее популярных атак². Исходными данными для статистического отчета послужили результаты множества пилотных проектов по внедрению межсетевого экрана уровня приложений PT Application Firewall среди российских и зарубежных компаний.

² <https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Web-Application-Vulnerability-2016-eng.pdf>

Для того чтобы взломать сайт или атаковать его пользователей, злоумышленники пробуют эксплуатировать различные уязвимости приложения — как ошибки, допущенные при разработке, так и недостатки администрирования. По результатам исследований было установлено, что в 58% приложений, для которых проводились пилотные проекты, злоумышленники предпринимали попытки атаковать пользователей путем эксплуатации самой распространенной в этом году уязвимости «Межсайтовое выполнение сценариев».

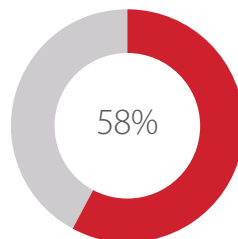


Рисунок 10. Атаки «Межсайтовое выполнение сценариев» (доля систем)

Веб-приложения, содержащие ошибки, связанные с раскрытием информации о версии ПО (Fingerprinting), занимают второе место в рейтинге и встречаются в 63% систем. Кроме того, более чем в половине систем (54%) выявлена утечка важных данных, в том числе исходного кода и персональных данных.

Отсутствие защиты от перебора учетных данных (Brute Force) остается на третьем месте, но доля приложений, уязвимых для такого перебора, увеличилась на 10%.

На пятом и шестом местах находятся недостатки защиты пользовательских сессий и отсутствие защиты от атак типа Clickjacking. Сравнение с прошлым периодом исследований для этих недостатков не приводится, поскольку в отдельные категории они были выделены лишь в этом году. В то время как разработчики стали более внимательно относиться к устранению критически опасных уязвимостей, несущих угрозу непосредственно для владельцев веб-приложений, на первый план в этом году выходят ошибки, затрагивающие пользователей. В 35% систем выявлена уязвимость, которая также позволяет проводить атаки на пользователей, — «Подделка межсайтовых запросов» (Cross-Site Request Forgery).

Как уже отмечалось, общая доля сайтов, содержащих уязвимости высокого уровня риска, уменьшилась, и в этом году в рейтинге оказалась всего одна критически опасная уязвимость — «Внедрение операторов SQL», которой, тем не менее, подвержены 25% веб-приложений. По итогам исследований атак на веб-приложения в 2016 году попытка эксплуатации этой уязвимости находится на первом месте в рейтинге наиболее популярных атак и встречается в 84% систем.

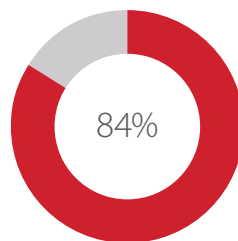


Рисунок 11. Атаки «Внедрение операторов SQL» (доля систем)

Уязвимости клиентской части в 2016 году составили 59% от общего числа. Среди них «Межсайтовое выполнение сценариев», «Подделка межсайтовых запросов», недостатки защиты сессии и другие проблемы безопасности, позволяющие совершать атаки на клиентов веб-приложения. 41% выявленных уязвимостей относятся к серверной части приложений, например «Утечка информации» или «Недостаточная авторизация».

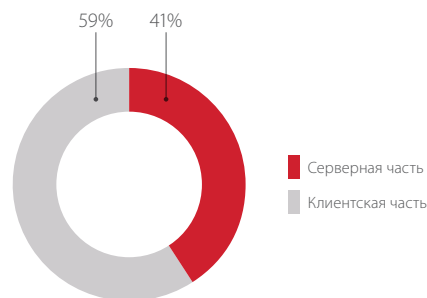


Рисунок 12. Уязвимости по объекту атаки

Большинство обнаруженных уязвимостей (73%) содержатся в программном коде и связаны с ошибками при разработке веб-приложений — как, например, «Внедрение операторов SQL». Некорректные параметры конфигурации веб-серверов составляют около четверти от общего числа недостатков безопасности.

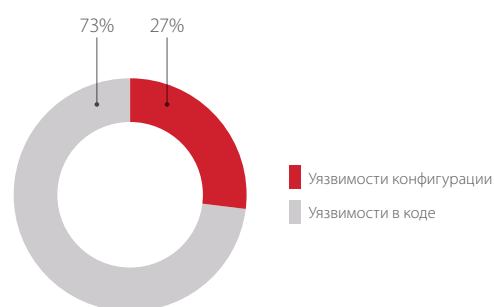


Рисунок 13. Типы уязвимостей

5.2. АНАЛИЗ УГРОЗ И УРОВНЕЙ ЗАЩИЩЕННОСТИ

В зависимости от последствий, которые могут вызвать имеющиеся уязвимости, введена градация уровней защищенности веб-приложений — от крайне низкого до приемлемого. Под крайне низкой степенью защиты мы понимаем наличие критически опасных уязвимостей, например позволяющих выполнять команды ОС сервера любому внешнему злоумышленнику или приводящих к разглашению особо чувствительной информации. В целом при наличии уязвимостей высокой степени риска уровень защищенности системы может варьироваться от крайне низкого до уровня ниже среднего.

Общий уровень защищенности веб-приложений остается достаточно низким. По результатам проведенного анализа уровень защищенности 16% систем был оценен экспертами как крайне низкий.

Низким уровнем защищенности характеризуется каждое третье из исследованных веб-приложений (32%). Достаточной степенью защиты обладают всего 5% приложений.

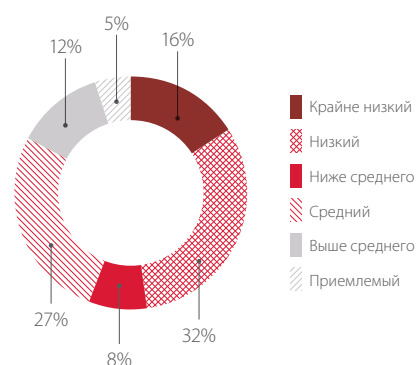


Рисунок 14. Уровень защищенности веб-приложений

Наименее защищенными в 2016 году оказались веб-приложения интернет-магазинов различного рода, промышленных и телекоммуникационных компаний: уровень защищенности более чем половины из них оценивается как низкий или крайне низкий. При этом крайне низкой степени защиты обладали более трети веб-приложений компаний из сфер электронной коммерции (34%) и промышленности (43%). Немногом лучше ситуация с банками и государственными организациями. Достаточный уровень защиты был отмечен лишь в 15% веб-приложений в сфере телекоммуникаций. Для приложений СМИ статистика не приводится, поскольку их выборка недостаточна для объективной оценки.

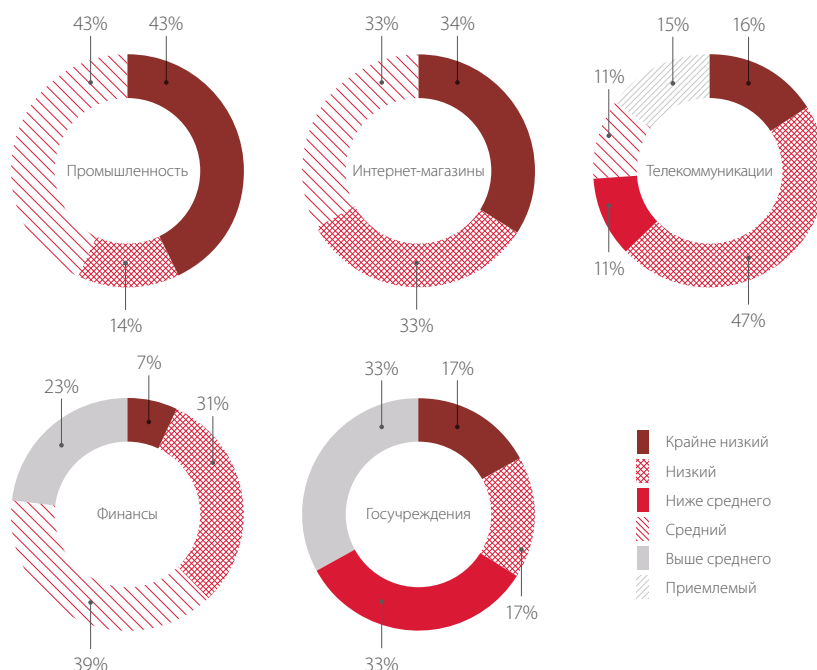


Рисунок 15. Уровень защищенности веб-приложений по отраслям

Самой распространенной угрозой в 2016 году оказались атаки на пользователей веб-приложений: такие атаки возможны практически во всех веб-приложениях (94%). Четверть веб-приложений, как было показано в предыдущем разделе, содержит уязвимости, пользуясь которыми злоумышленник может получить доступ к базам данных. Столько же веб-приложений (25%) могут стать вектором проникновения во внутреннюю сеть компании, в частности они позволяют сканировать устройства, получать информацию о структуре сети, отправлять запросы к локальным узлам. Приблизительно каждое пятое приложение (19%) дает возможность выполнять произвольные команды ОС на сервере.

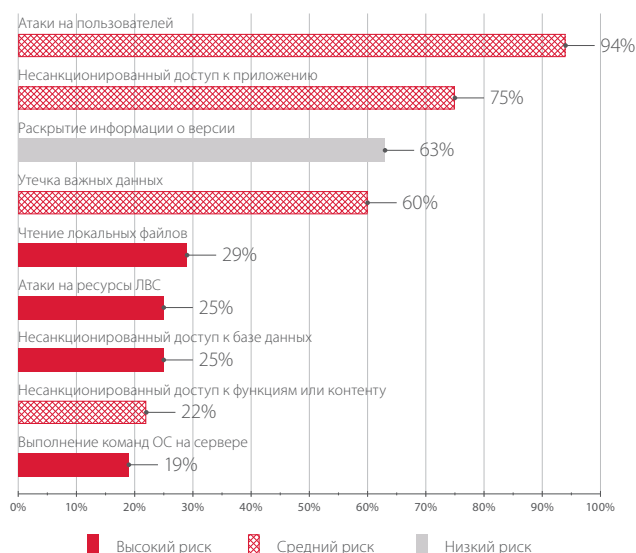


Рисунок 16. Самые распространенные угрозы

Необходимо добавить, что в перечень проверок, осуществляемых в рамках анализа защищенности, не входит тестирование на отказ в обслуживании. Тем не менее в ряде приложений были выявлены уязвимости, которые позволяют злоумышленнику осуществлять подобные атаки.

В основном совершать атаки на пользователей приложений позволяли такие уязвимости, как «Межсайтовое выполнение сценариев», «Подделка межсайтовых запросов», «Открытое перенаправление», а также недостаточная защита сессий и отсутствие защиты от атак типа Clickjacking. Эти ошибки вошли в топ-10 самых распространенных уязвимостей в этом году.

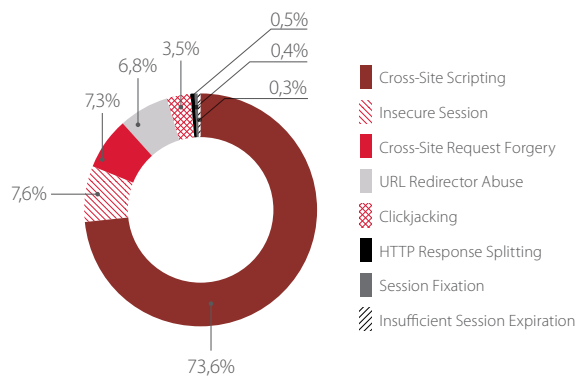


Рисунок 17. Соотношение уязвимостей, позволяющих проводить атаки на пользователей

Злоумышленник может получить несанкционированный доступ к 75% приложений. Причиной этого служат в основном слабая парольная политика, отсутствие защиты от перебора учетных данных и возможность проведения атак на пользователей.

Третью и четвертую строчки рейтинга занимают угрозы, связанные с утечкой информации. Раскрытие информации о версии приложения является недостатком низкого уровня риска, однако при использовании устаревшего ПО злоумышленник может использовать известные уязвимости, для большинства из которых можно найти эксплойты в открытом доступе.

Отдельно отметим, что в 8% систем в результате эксплуатации различных уязвимостей был получен доступ к исходному коду веб-приложений. Анализ исходного кода позволяет злоумышленникам выявить другие уязвимости приложения и спланировать дальнейшее развитие вектора атаки. Кроме того, в исходном коде приложения может содержаться чувствительная информация для доступа к критически важным ресурсам.

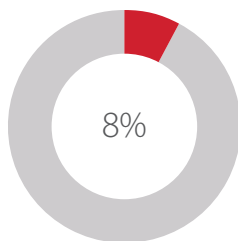


Рисунок 18. Доля систем, в которых злоумышленник может получить доступ к исходному коду

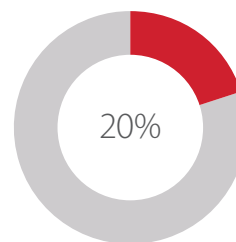


Рисунок 19. Доля систем, в которых злоумышленник может получить доступ к персональным данным пользователей

Под угрозой также находятся и персональные данные пользователей, доступ к которым возможен в 20% систем, обрабатывающих такие данные, включая сайты банков и государственных организаций. Злоумышленник может получить сведения о пользователях как вследствие утечки информации, так и в результате эксплуатации других уязвимостей, например внедрения операторов SQL.

Если рассмотреть критически опасные угрозы в зависимости от отрасли, то можно отметить, что в приложениях госучреждений, финансовых и телекоммуникационных компаний встречаются все угрозы высокого уровня риска. Угроза доступа к СУБД и выполнения команд ОС распространена по большей части среди веб-приложений интернет-магазинов и промышленных компаний.

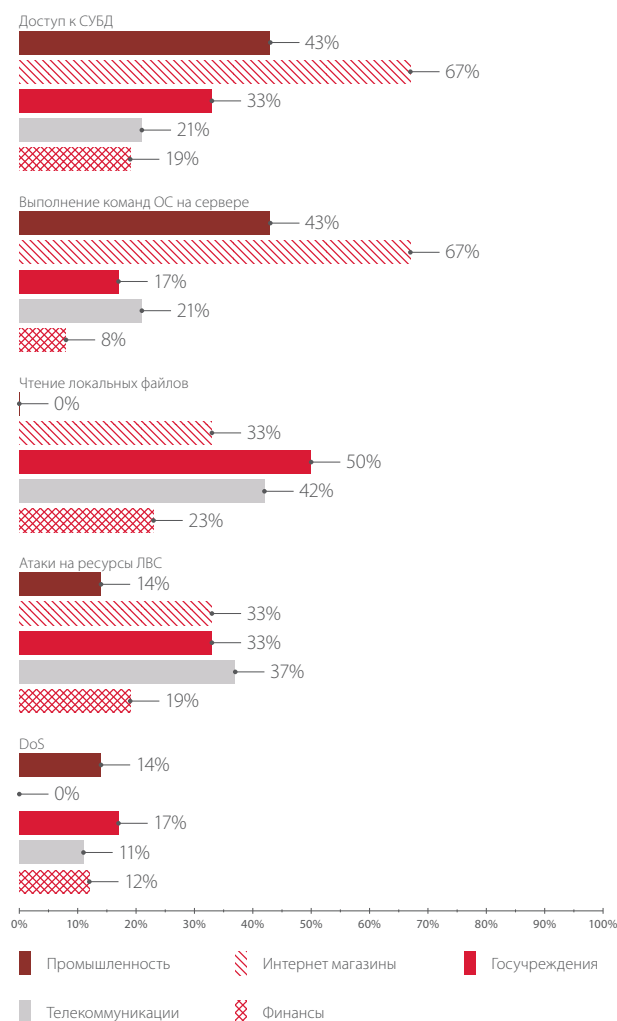


Рисунок 20. Критически опасные угрозы по отраслям

5.3. СТАТИСТИКА ДЛЯ РАЗЛИЧНЫХ ОТРАСЛЕЙ ЭКОНОМИКИ

В данном разделе представлена статистика веб-приложений сферы телекоммуникаций, финансового сектора, электронной коммерции, государственных учреждений, а также промышленности. В текущем разделе не рассматриваются веб-приложения СМИ, поскольку их выборка недостаточна для объективной оценки.

Как показывают полученные результаты, во всех отраслях, за исключением финансовой, преобладают веб-приложения, подверженные уязвимостям высокой степени опасности. Так, уязвимости высокого уровня риска найдены в 74% веб-приложений телекоммуникационных компаний, 67% приложений государственных организаций и интернет-магазинов, 57% приложений промышленных компаний. В финансовой сфере таким уязвимостям оказались подвержены всего треть веб-приложений (38%).

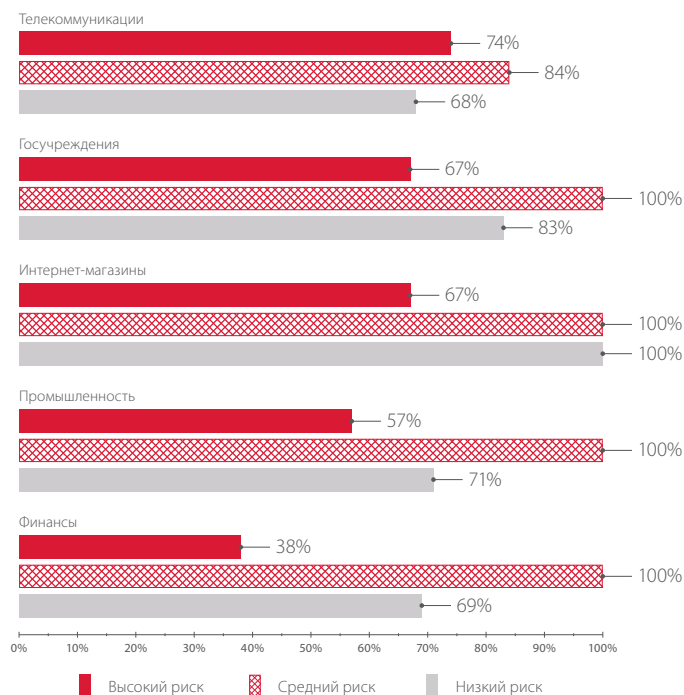


Рисунок 21. Доли веб-приложений с уязвимостями различной степени риска

Уязвимости средней степени опасности были обнаружены во всех исследованных веб-приложениях, кроме некоторых приложений телекоммуникационных компаний. Можно заметить, что в сфере телекоммуникаций присутствуют как множество веб-приложений, содержащих критически опасные уязвимости, так и небольшой процент относительно защищенных веб-приложений, где были выявлены лишь незначительные недостатки.

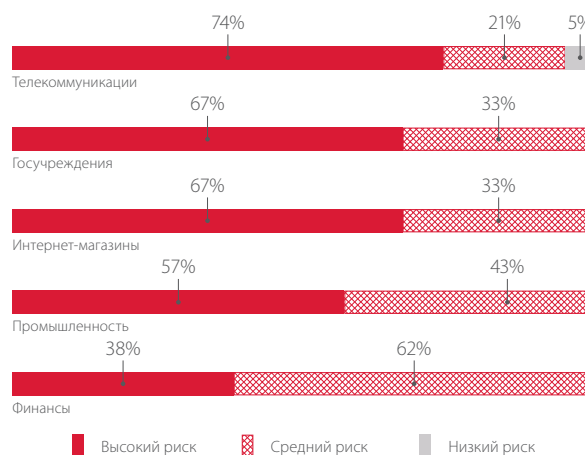


Рисунок 22. Максимальная степень риска (доли веб-приложений)

Если же рассмотреть среднее число уязвимостей различного уровня риска, обнаруженных в веб-приложениях, то на первом месте по количеству критически опасных уязвимостей окажутся веб-приложения государственных компаний — 6,2 уязвимости на систему. Напомним, что в прошлом году этот показатель составлял меньше единицы (0,7 уязвимости). Если в предыдущие годы проверки проводились лишь для наиболее важных государственных систем, при разработке которых безопасность является одним из главных требований, то в настоящее время все большее число государственных организаций начинает обращать внимание на вопросы безопасности, проводить анализ существующих систем, в том числе и веб-приложений, уровень защищенности которых оказывается достаточно низким.

Также высокое число критически опасных уязвимостей присутствует в веб-приложениях компаний, занимающихся электронной коммерцией. Эти веб-приложения содержат и наибольшее количество уязвимостей среднего уровня риска: 39,3 уязвимости на систему в среднем выявляется для интернет-магазинов и 27,5 уязвимости — для государственных организаций.

Около двух уязвимостей критического уровня риска в среднем можно обнаружить в веб-приложениях промышленных и телекоммуникационных компаний, а наиболее защищенными по данному критерию являются веб-приложения финансового сектора — всего 0,8 уязвимости высокого уровня риска на приложение.

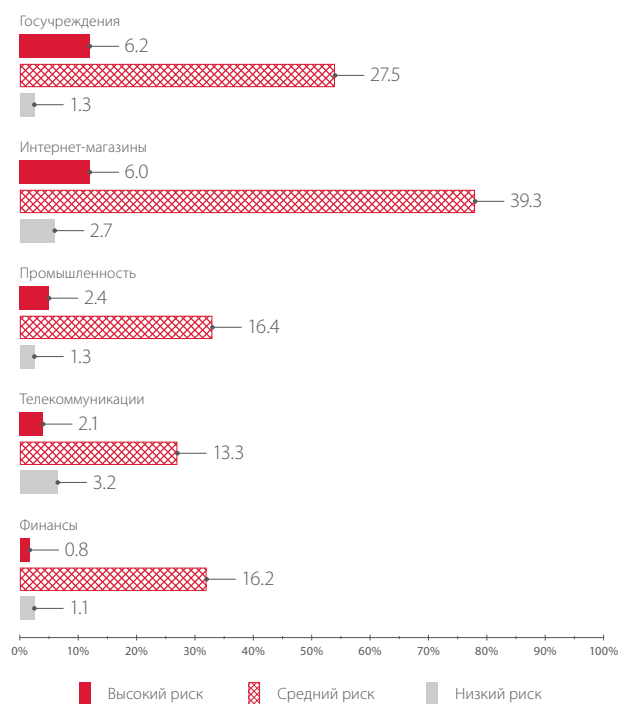


Рисунок 23. Среднее число уязвимостей различного уровня риска на одно приложение

Среди уязвимостей высокого уровня риска в 2016 году наиболее часто встречалась уязвимость «Внедрение операторов SQL», она входит в десятку самых популярных уязвимостей и содержится в веб-приложениях всех отраслей. Распространены и такие опасные уязвимости, как «Внедрение внешних сущностей XML», «Выполнение команд ОС» и «Выход за пределы назначенного каталога». В веб-приложениях телекоммуникационных компаний и финансовых организаций встречаются все перечисленные ошибки, однако это может быть связано с тем, что приложения из этих отраслей составили большую часть выборки.

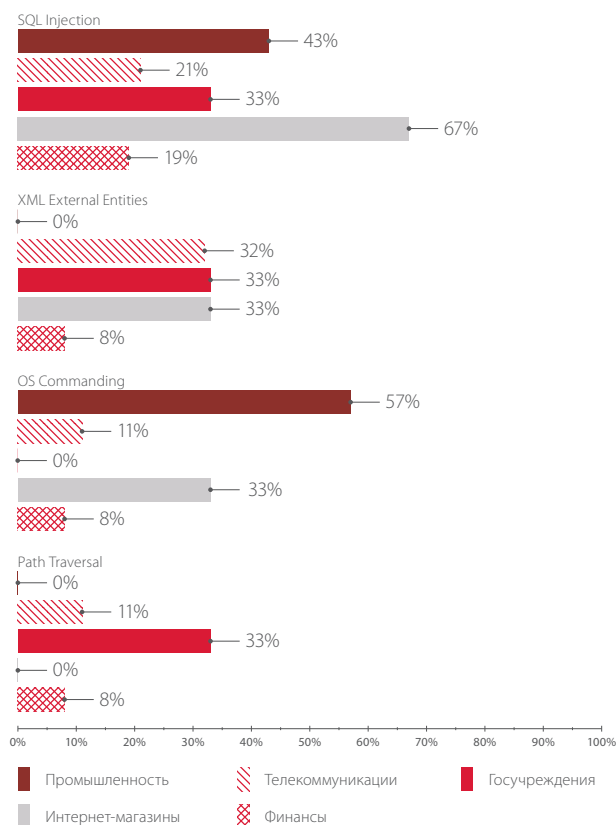


Рисунок 24. Доли отраслевых сайтов с распространенными уязвимостями

5.4. АНАЛИЗ РАЗЛИЧНЫХ СРЕДСТВ РАЗРАБОТКИ

Как и в предыдущем году, все исследованные приложения, независимо от средства разработки, содержат уязвимости как минимум среднего уровня риска. В приведенной здесь статистике рассматриваются сайты на PHP, Java и ASP.NET. Приложения, написанные на других, менее популярных языках, присутствовали в выборке в малом количестве, поэтому статистика по ним не приводится - однако во всех приложениях, разработанных с использованием этих средств, обнаружены критически опасные уязвимости, и лишь в одном выявлены только уязвимости низкого уровня риска.

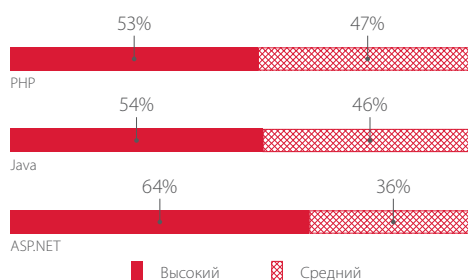


Рисунок 25. Доля веб-приложений по максимальному уровню риска уязвимостей

В 2016 году распределение приложений, созданных с использованием языков программирования PHP и Java, по уровню опасности выявленных уязвимостей оказалось практически одинаковым: все приложения подвержены уязвимостям среднего уровня риска, а более половины приложений содержат критически опасные уязвимости.

Наиболее высокий процент приложений, содержащих критически опасные уязвимости (64%), наблюдается среди систем, разработанных на базе технологии ASP.NET. При этом доля приложений, в которых были обнаружены уязвимости среднего и низкого уровня риска, несколько ниже, чем для PHP и Java: 93% и 50% соответственно. В то же время на одно приложение ASP.NET в среднем приходится меньше критически опасных уязвимостей, чем для приложений на основе PHP и Java.

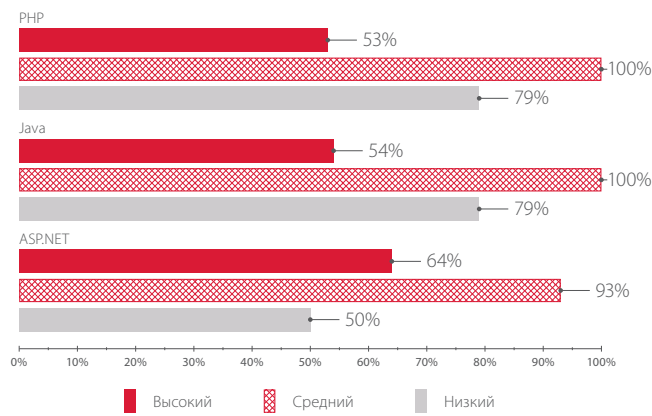


Рисунок 26. Доли систем с уязвимостями разной степени риска

Как уже упоминалось, по сравнению с прошлыми периодами исследований заметно сократилось количество критически опасных уязвимостей. В среднем на одну систему приходится около 2 уязвимостей высокого уровня риска, максимальное число таких уязвимостей (2,8) содержат приложения, разработанные на языке PHP. Число уязвимостей среднего уровня опасности, напротив, увеличилось в сравнении с показателями предыдущего года (для языков PHP и Java), при этом приложения, созданные на базе языка Java, содержат почти в два раза больше таких уязвимостей, чем остальные приложения.

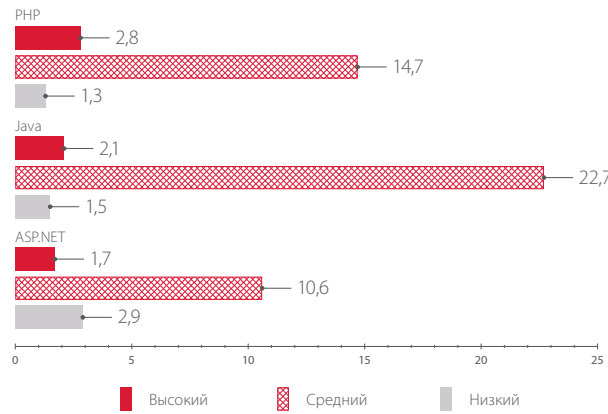


Рисунок 27. Среднее количества уязвимостей на одну систему в зависимости от средства разработки

Рейтинг самых распространенных недостатков в зависимости от средств разработки представлен в таблице 1.

PHP	Доля сайтов	ASP.NET	Доля сайтов	Java	Доля сайтов
Cross-Site Scripting	79%	Cross-Site Scripting	79%	Cross-Site Scripting	64%
Information Leakage	79%	Fingerprinting	79%	Insecure Session	57%
Brute Force	74%	Brute Force	75%	Cross-Site Request Forgery	50%
Fingerprinting	74%	Information Leakage	50%	URL Redirector Abuse	43%
Insecure Session	53%	Clickjacking	50%	Deserialization of Untrusted Data	29%
Clickjacking	42%	Cross-Site Request Forgery	42%	Information Leakage	29%
Insufficient Authorization	32%	Insecure Session	42%	SQL Injection	29%
SQL Injection	26%	Insufficient Authorization	33%	Clickjacking	21%
OS Commanding	26%	SQL Injection	29%	Insufficient Authorization	21%
URL Redirector Abuse	26%	XML External Entities	21%	XML External Entities	14%

Уровень риска: ■ Высокий ■ Средний ■ Низкий

Таблица 1. Наиболее распространенные уязвимости (по средствам разработки)

Самой распространенной уязвимостью для всех приложений оказалась уязвимость «Межсайтовое выполнение сценариев», она обнаружена более чем в 60% приложений для всех языков программирования. Распространены также недостатки, связанные с раскрытием чувствительных данных: «Утечка информации» и «Раскрытие информации о версии ПО».

Для языков PHP и Java по сравнению с прошлым годом сократилась доля приложений, содержащих уязвимости высокого уровня риска. Так, например, в приведенном рейтинге отсутствует популярная в прошлые годы уязвимость «Выход за пределы назначенного каталога».

Тем не менее от 26% до 29% приложений в каждой категории содержат критически опасную уязвимость «Внедрение операторов SQL», более четверти приложений (26%) на базе языка PHP содержат уязвимость «Выполнение команд ОС», а для остальных средств разработки в десятку наиболее распространенных недостатков входит «Внедрение внешних сущностей XML».

Независимо от средства разработки та или иная часть приложений подвержена уязвимостям из общего рейтинга наиболее распространенных недостатков, что отражено на приведенных ниже рисунках.

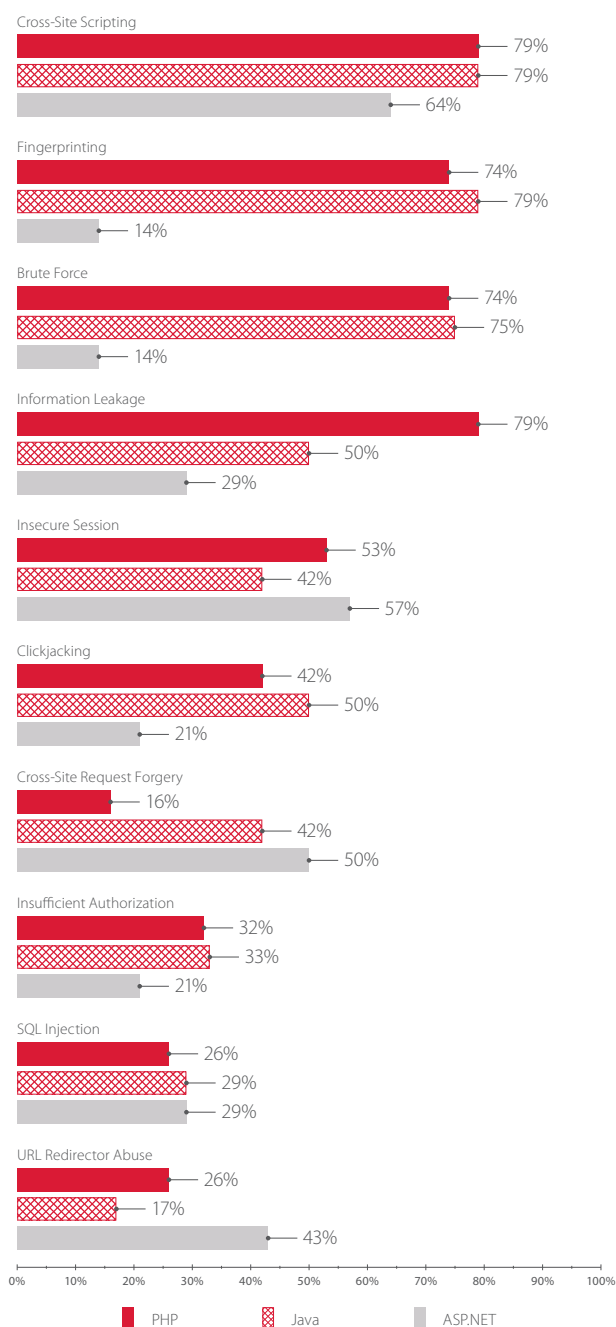


Рисунок 28. Доля веб-приложений, подверженных наиболее распространенным уязвимостям, по средствам разработки

5.5. СРАВНЕНИЕ ТЕСТОВЫХ И ПРОДУКТИВНЫХ СИСТЕМ

В 2016 году продуктивные системы оказались более уязвимыми, чем тестовые. Каждое второе веб-приложение на стадии разработки содержит критически опасные ошибки, а среди систем, находящихся в процессе эксплуатации, такие ошибки выявляются уже в 55% приложений.

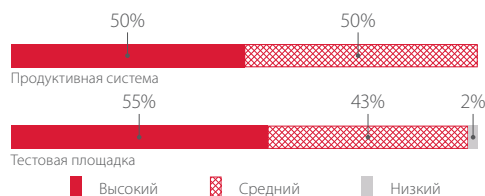


Рисунок 29. Максимальный уровень риска уязвимостей (доли систем)

Продуктивные системы являются менее защищенными и по количеству обнаруженных уязвимостей. Ошибок высокой и средней степени риска в продуктивных системах выявлено в среднем в два раза больше, чем на тестовых площадках. Можно объяснить такое распределение тем, что компании, внедряющие процессы обеспечения безопасности, в том числе тестирование приложений на стадии разработки, в целом более ответственно относятся к вопросам безопасности. Также следует учесть, что некоторые уязвимости возможно выявить только в полностью сконфигурированной и готовой к использованию системе; кроме того, новые ошибки могут возникнуть и на этапе внедрения.

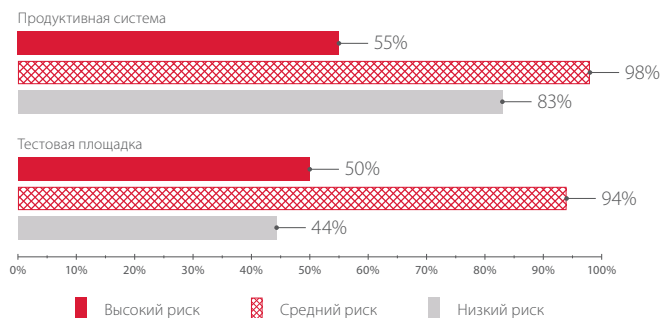


Рисунок 30. Доли веб-приложений с уязвимостями разной степени риска

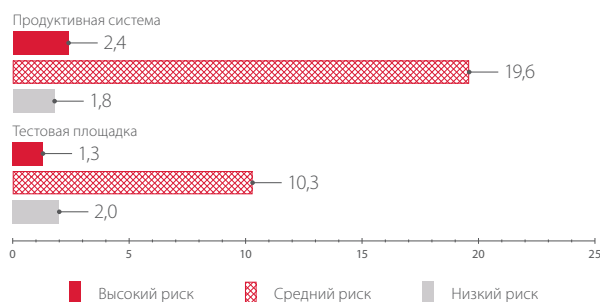


Рисунок 31. Среднее количество уязвимостей на одну систему

Подобные результаты свидетельствуют о том, что необходимо внедрять процессы обеспечения безопасности веб-приложений на протяжении всего жизненного цикла: как на этапе разработки, так и при внедрении и дальнейшем использовании.

5.6. СРАВНЕНИЕ МЕТОДОВ ТЕСТИРОВАНИЯ

В рамках ручного анализа защищенности использовались методы черного, серого и белого ящика. Так как эти методы применялись в отношении разных систем, нельзя напрямую сравнить их результаты, но можно использовать эти значения для общей оценки эффективности методов тестирования. Для большинства систем (81%) исследования проводились методами черного и серого ящика без доступа к исходному коду.

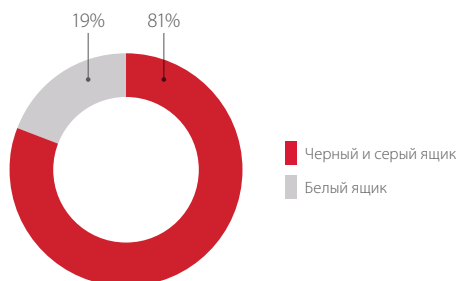


Рисунок 32. Доля приложений по методу тестирования

Как видно из результатов исследования, в 75% веб-приложений, для которых проводился анализ исходного кода, были выявлены критически опасные уязвимости. Для тех приложений, где исходный код не был доступен, этот показатель составляет 49%. Уязвимости среднего уровня риска были обнаружены практически во всех приложениях как при использовании метода черного ящика (98%), так и в результате анализа исходного кода (92%). Таким образом, анализ приложения методом белого ящика в большинстве случаев оказывается более эффективным, но тем не менее внешний злоумышленник, не обладающий сведениями о системе, также с высокой долей вероятности сможет обнаружить уязвимости различной степени риска, в том числе критически опасные.

Помимо этого, стоит принять во внимание тот факт, что в результате эксплуатации различных уязвимостей злоумышленник может получить доступ и к исходному коду приложения, как было показано выше.

Необходимо учитывать, что в рамках анализа защищенности веб-приложения методом черного или серого ящика проводятся только те проверки, которые не повлияют на функционирование приложения и не вызовут отказ в обслуживании. Злоумышленник же вряд ли станет руководствоваться этим принципом.

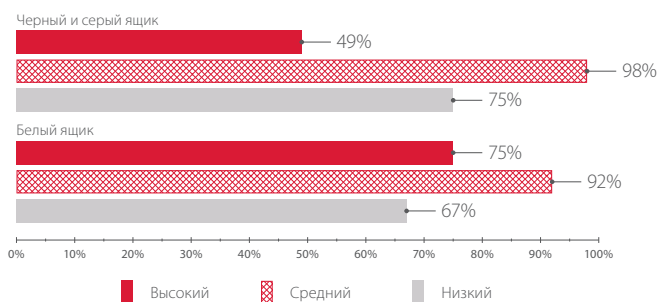


Рисунок 33. Доли систем с уязвимостями разной степени риска в зависимости от метода тестирования

В среднем при наличии доступа к исходному коду в одном приложении специалисты Positive Technologies выявляли 2,8 уязвимости высокого уровня риска, в то время как методом черного ящика были обнаружены 1,9 уязвимости на систему. В связи с тем, что существенная часть работ по анализу исходного кода проводилась в отношении тестовых версий новых приложений для компаний, которые уже проводили тестирование своих систем годом ранее, разница не столь велика, как в прошлый период исследований. Эффективность метода белого ящика можно подтвердить и на основе результатов автоматизированного анализа исходного кода, которые представлены в следующем разделе.

Анализ исходного кода позволял также выявить в три раза больше уязвимостей низкого уровня риска, чем при использовании метода черного ящика. Для уязвимостей среднего уровня опасности результаты отличаются незначительно.

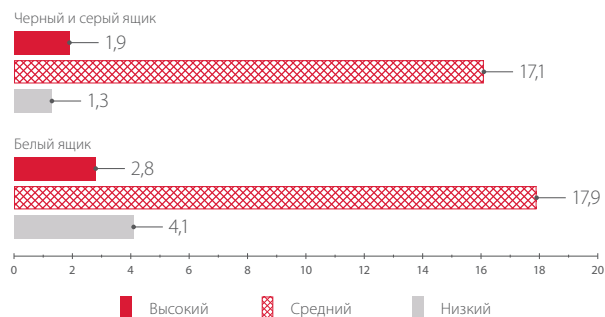


Рисунок 34. Среднее число уязвимостей на одну систему в зависимости от метода тестирования

Метод белого ящика, как и в прошлые периоды исследований, оказался эффективнее при выявлении критически опасных уязвимостей. Например, анализ исходного кода позволял в четыре раза чаще выявлять уязвимости «Внедрение внешних сущностей XML». Кроме того, представленные результаты показывают, что недостатки защиты сессии, «Подделка межсайтовых запросов» и «Открытое перенаправление» также были обнаружены в основном при использовании метода белого ящика.

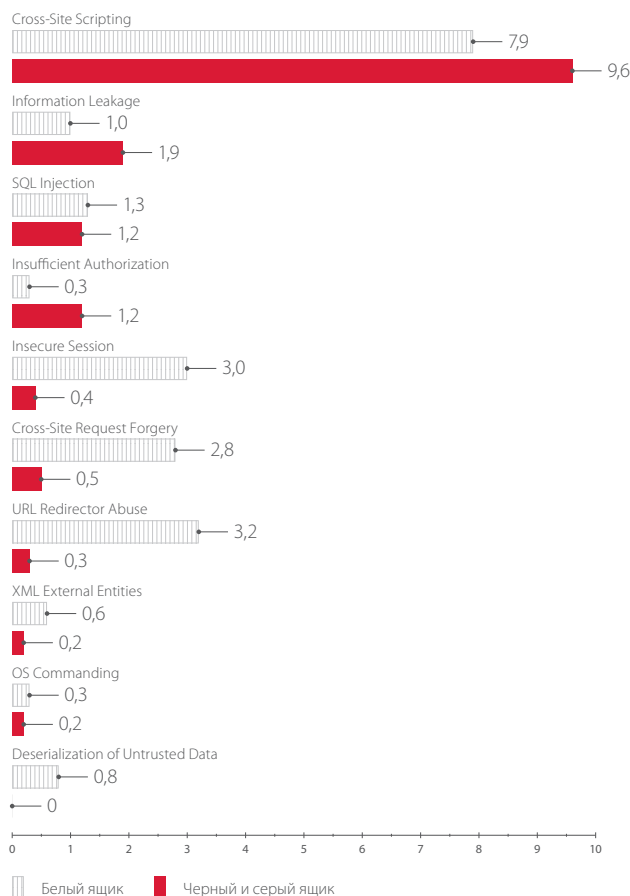


Рисунок 35. Среднее число уязвимостей определенного типа на одну систему

6. АВТОМАТИЗИРОВАННЫЙ АНАЛИЗ ЗАЩИЩЕННОСТИ

В данном разделе будут рассмотрены веб-приложения, для которых проводился анализ исходного кода с использованием автоматизированного анализатора. Проверки исходного кода ручными методами и с помощью автоматизированных средств проводились в отношении разных систем, поэтому их результаты нельзя сравнивать между собой.

Все приложения, вошедшие в данную выборку, являлись тестовыми, причем часть приложений находилась на ранней стадии разработки. Уязвимости, обнаруженные в ходе автоматизированного анализа и включенные в представленную статистику, были подтверждены вручную на тестовых стендах.

В разделе используется классификация уязвимостей, которая применяется в автоматизированном сканере защищенности. Эта классификация отличается от предложенной WASC, в частности, более детальной проработкой недостатков, которые в классификации WASC объединены в категории общих недостатков конфигурации приложения, настроек доступа к файловой системе и др.

По сравнению с прошлым годом для веб-приложений, исследуемых автоматизированными средствами, также наблюдаются позитивные изменения. Среди всех выявленных уязвимостей более четверти являются критически опасными (28,5%), тогда как в прошлом году такие уязвимости составляли около 40%. Также как и в случае ручных проверок, на итоговые результаты влияет тот факт, что компании, проводившие тестирование приложений в прошлом году, более внимательно относятся к вопросам безопасности при разработке новых приложений.

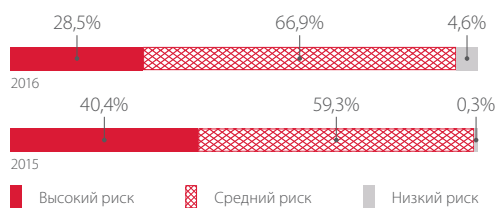


Рисунок 36. Распределение уязвимостей по уровню риска (автоматизированное тестирование)

Во всех исследованных веб-приложениях были обнаружены уязвимости не ниже среднего уровня риска. Критически опасные уязвимости, также как и в 2015 году, выявлены в подавляющем большинстве приложений (89%).

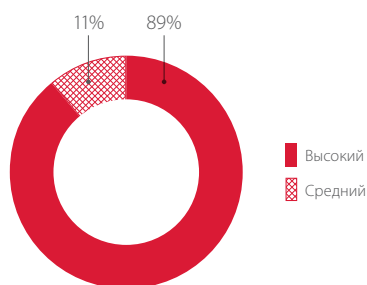


Рисунок 37. Распределение систем по максимальной степени риска уязвимостей (автоматизированное тестирование)

Уязвимости среднего уровня риска обнаружены во всех исследованных системах.

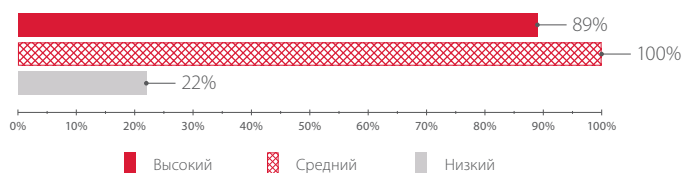


Рисунок 38. Доли систем с уязвимостями различной степени риска (автоматизированное тестирование)

Методами автоматизированного анализа исходного кода в среднем в одном приложении было обнаружено 4,6 уязвимости высокого уровня риска, 66,9 уязвимости среднего уровня риска и 45,9 уязвимости низкого уровня риска. Более того, в двух исследованных приложениях были обнаружены сотни критически опасных уязвимостей и около двух тысяч уязвимостей средней степени опасности, однако эти приложения были исключены из приведенной статистики для отображения более объективных значений. Тем не менее эти данные позволяют получить представление об эффективности использования автоматизированных средств анализа и о том, насколько применение подобных средств может повысить уровень защищенности разрабатываемых веб-приложений. Анализ исходного кода, в отличие от метода черного ящика, позволяет выявить все «точки входа», то есть все возможные варианты эксплуатации каждой уязвимости, что позволяет устранить уязвимость полностью.

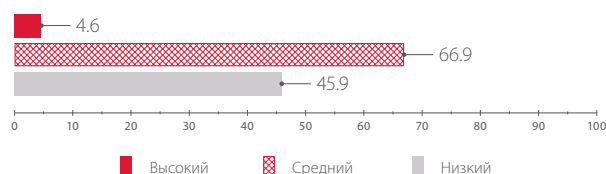


Рисунок 39. Среднее число уязвимостей на одну систему (автоматизированный анализ)

Все исследуемые системы оказались подвержены уязвимости «Межсайтовое выполнение сценариев». Как и в случае ручного тестирования, это самая распространенная уязвимость в рассматриваемых приложениях. Пример обнаружения такой уязвимости представлен на рисунке ниже. Приложение не осуществляет проверку передаваемых пользователем данных, чем может воспользоваться злоумышленник и передать, например, сценарий на языке JavaScript, чтобы осуществить атаку на пользователей приложения.

Межсайтовое выполнение сценариев		Описание уязвимости
Уязвимый код:	39 value=<%= request.getParameter(ContextParamId.ID [PARAM]) %>/'>	
Функция:	javax.servlet.jsp.JspWriter.print(java.lang.String)	
Уязвимый файл:	[redacted].jsp	
Входной файл:	[redacted].jsp	
Запрос:	GET [redacted]Block2Form.jsp? ID=[redacted]PARAM=%2F%3E%3Cscript%3Ealert%281%29%3C%2Fscript%3E HTTP/1.1 Host:[redacted].ru Connection: close	
Условие:	(((org.apache.struts.taglib.html.FormTag) _005jspx_005ftagPool_005html_005fform_0026_005fonsubmit_005fmethod_005faction.get(class org.apache.struts.taglib.html.FormTag)).doStartTag() != javax.servlet.jsp.tagext.Tag.SKIP_BODY == true)	
OWASP - A3	CWE-79	

Рисунок 40. Пример обнаружения уязвимости «Межсайтовое выполнение сценариев»

Используемый в наших исследованиях анализатор кода позволяет автоматически создавать эксплойты для проверки наличия уязвимости, в данном случае эксплойт был составлен для отправки запроса методом GET.

Наиболее распространенными уязвимостями высокого уровня риска являются недостатки, связанные с разграничением доступа к файлам. Установлено, что почти половина исследованных веб-приложений позволяет создавать и модифицировать произвольные файлы, что в свою очередь может привести к выполнению команд ОС, например если злоумышленник создаст файл с расширением .php. Практически во всех этих приложениях такие ошибки встречаются в совокупности с возможностью чтения и удаления произвольных файлов. Пример обнаружения уязвимости в исходном коде представлен на рисунке. Уязвимость позволяет злоумышленнику выйти за пределы назначенного каталога и прочитать произвольные файлы на сервере.

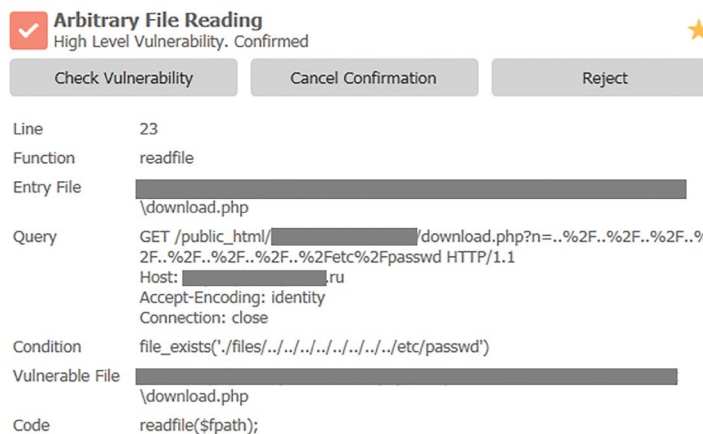


Рисунок 41. Пример обнаружения уязвимости «Чтение произвольных файлов»

В исходном коде ряда веб-приложений была обнаружена и критически опасная уязвимость «Внедрение операторов SQL», которая также связана с недостаточной фильтрацией входных данных. Эта уязвимость позволяет не только получать информацию из базы данных, но также в некоторых случаях читать произвольные файлы, создавать новые, проводить атаки, направленные на отказ в обслуживании. Ниже приведены пример уязвимого кода, выявленного анализатором, и эксплойт для проверки возможности эксплуатации уязвимости.

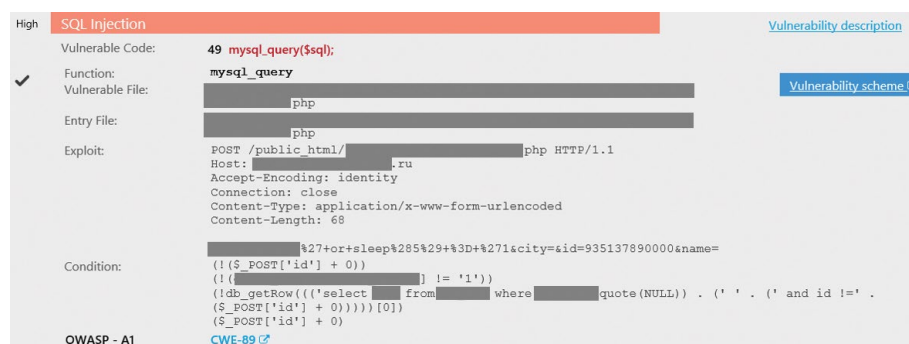


Рисунок 42. Пример обнаружения уязвимости «Внедрение операторов SQL»

Менее распространена в 2016 году, но также встречается критически опасная уязвимость «Внедрение внешних сущностей XML», которая может быть использована злоумышленником для чтения произвольных файлов или проведения атак на ресурсы внутренней сети. На рисунке представлен пример такой уязвимости.

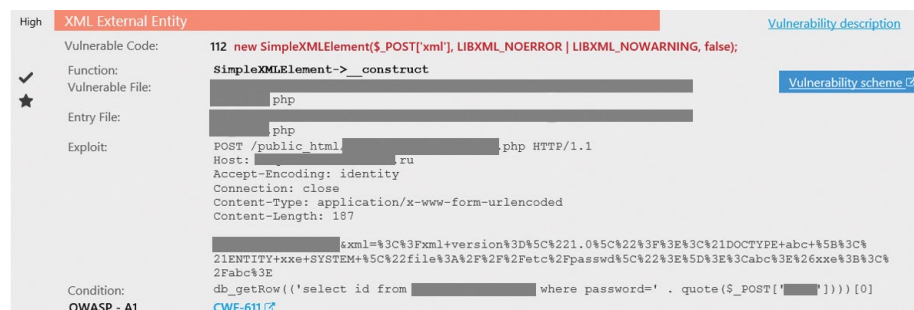


Рисунок 43. Пример обнаружения уязвимости «Внедрение внешних сущностей XML»

Также были выявлены недостатки в коде приложений (жестко заданный пароль, использование односторонней хеш-функции без соли, статический генератор случайных чисел) и другие ошибки.

В целом полученные результаты свидетельствуют о том, что анализ защищенности веб-приложения необходимо осуществлять на всех стадиях жизненного цикла, причем тестирование исходного кода приложения автоматизированными средствами позволяет выявить максимальное число ошибок программирования в кратчайшее время, в том числе и ошибки высокого уровня риска, которые ведут к критически опасным угрозам и могут быть выявлены злоумышленником во время эксплуатации приложения.

ЗАКЛЮЧЕНИЕ

Результаты исследования показывают, что несмотря на ярко выраженные позитивные тенденции, общий уровень защищенности веб-приложений остается достаточно низким. Более чем в половине веб-приложений выявляются критически опасные уязвимости, причем при наличии у злоумышленника доступа к исходному коду этот показатель резко возрастает. Обнаруженные уязвимости позволяют нарушителю получить большое количество чувствительной информации, например исходный код приложения или персональные данные пользователей, в том числе на сайтах банков и государственных учреждений. Не защищены от атак и сами пользователи: практически все приложения дают злоумышленнику возможность провести на них атаки.

Кроме того, уязвимости веб-приложений являются одним из наиболее доступных злоумышленнику векторов проникновения во внутреннюю сеть компании, среди исследованных систем около четверти могут стать причиной несанкционированного доступа ко внутренним ресурсам.

Анализ исходного кода показывает намного более высокие результаты, чем исследование защищенности без доступа к коду приложения. Кроме того, тестирование исходного кода в процессе разработки позволяет значительно повысить защищенность конечного приложения. Для анализа исходного кода на различных стадиях разработки целесообразно применять автоматизированные средства, поскольку скорость работы анализатора превосходит возможности ручного анализа.

Веб-приложения, находящиеся в процессе эксплуатации, оказались более уязвимыми, чем тестовые: это свидетельствует о том, что необходимо проводить анализ защищенности как на стадии разработки, так и после внедрения в эксплуатацию. В качестве превентивной меры защиты рекомендуется использовать межсетевой экран уровня приложений (web application firewall).

В целом вопросам безопасности веб-приложений все еще уделяется недостаточно внимания, хотя в 2016 году мы видим изменения к лучшему. Необходимо внедрять процессы обеспечения безопасности на протяжении всего цикла жизни веб-приложения, что относится как к разработчикам кода приложений, так и к администраторам систем, обеспечивающим их функционирование. Только при условии принятия всех необходимых мер защиты в комплексе, в том числе внедрении процессов безопасной разработки, применении превентивных способов защиты и регулярного исследования безопасности веб-приложений, можно снизить возможные риски и обеспечить приемлемый уровень защиты.

О компании

Positive Technologies — один из лидеров европейского рынка систем анализа защищенности и соответствия стандартам, а также защиты веб-приложений. Деятельность компании лицензирована Минобороны России, ФСБ России и ФСТЭК России, продукция сертифицирована ФСТЭК России и в системе добровольной сертификации «Газпромсерт». Организации во многих странах мира используют решения Positive Technologies для оценки уровня безопасности своих сетей и приложений, для выполнения требований регулирующих организаций и блокирования атак в режиме реального времени. Благодаря многолетним исследованиям специалисты Positive Technologies заслужили репутацию экспертов международного уровня в вопросах защиты SCADA- и ERP-систем, крупнейших банков и телеком-операторов.