POSITIVE TECHNOLOGIES

СТАТИСТИКА УЯЗВИМОСТЕЙ ВЕБ-ПРИЛОЖЕНИЙ В 2017 ГОДУ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Веб-приложения используют самые разные организации, независимо от их сферы деятельности. Государственные учреждения предоставляют гражданам различные сервисы, СМИ размещают актуальные новости на своих площадках, IT- и телекоммуникационные компании рекламируют и продают услуги и продукты; всевозможные организации используют веб-приложения для обеспечения внутренних бизнес-процессов.

Особенность развития современных информационных технологий такова, что вопросы безопасности часто решаются по остаточному принципу. Мы регулярно видим новости о том, что веб-приложение той или иной организации было взломано: похищены критически важные данные, пользователи атакованы, руководство приносит извинения, компания подсчитывает убытки. Можно ли было избежать этого? Мы думаем, что да.

Каждый год специалисты Positive Technologies анализируют защищенность веб-приложений и систем ДБО, тестируют на проникновение корпоративную инфраструктуру и беспроводные сети, проводят инструментальный анализ и оценку осведомленности пользователей в вопросах ИБ для множества компаний в России и за рубежом. Данное исследование содержит результаты, полученные в ходе проведения работ по тестированию веб-приложений. Также приводится сравнительный анализ и статистика, собранная в ходе аналогичных исследований за 2016 год.

1. МЕТОДИКА

В данном исследовании представлены 23 веб-приложения, для которых в 2017 году проводился углубленный анализ с наиболее полным покрытием проверок. Результаты проектов по тестированию на проникновение, инструментальному сканированию и исследованию систем ДБО не вошли в статистику: эта информация представлена в других аналитических отчетах.

Оценка защищенности проводилась ручным способом методами черного, серого и белого ящика с использованием вспомогательных автоматизированных средств. Метод черного ящика заключается в проведении работ по оценке защищенности информационной системы от лица внешнего атакующего без предварительного получения какой-либо дополнительной информации о ней от владельца. Метод серого ящика аналогичен, но в качестве нарушителя рассматривается пользователь, обладающий определенными привилегиями в системе. При анализе методом белого ящика для оценки защищенности информационной системы используются все имеющиеся данные о ней, включая исходный код приложений.

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.

В настоящей статистике 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

Несанкционированный доступ к приложению возможен в каждой второй системе. В 2017 году наши эксперты демонстрировали возможность несанкционированного доступа к веб-приложению в 48% систем. При этом возможность получения полного контроля над исследуемыми приложениями была выявлена в 17% проектов.

Все веб-приложения по-прежнему уязвимы. Во всех исследованных системах наши эксперты выявили уязвимости. Критически опасные уязвимости были обнаружены в 52% приложений, а уязвимости средней степени риска — в каждом из них.

65% всех уязвимостей обусловлены ошибками в коде. Десятка самых распространенных уязвимостей 2017 года включает четыре критически опасные — «Внедрение SQL-кода», «Выполнение произвольного кода», «Выход за пределы назначенного каталога», «Внедрение внешних сущностей XML». Все эти уязвимости относятся к уязвимостям кода веб-приложений.

Утечка персональных данных в 44% приложений. Злоумышленник может получить персональные данные в 44% систем, в которых осуществляется их обработка. Речь идет, среди прочего, о финансовых учреждениях, интернет-магазинах и телекоммуникационных компаниях. При этом доля приложений, для которых существует угроза утечки критически важной информации, составляет 70%.

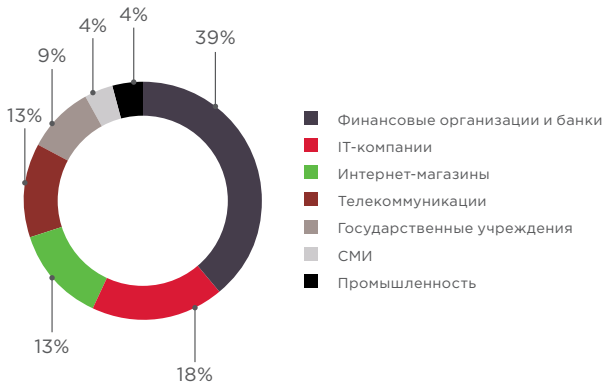
60% веб-приложений, разработанных на PHP, содержат критически опасные уязвимости. При этом примерно по две критически опасные уязвимости в среднем на одну систему удавалось выявить вне зависимости от применяемой технологии разработки. И независимо от средства разработки почти в каждом приложении присутствуют уязвимости из рейтинга наиболее распространенных.

Тестовые приложения содержат больше уязвимостей, чем продуктивные. В среднем в них удавалось обнаружить в пять раз больше критически опасных уязвимостей.

Наличие исходного кода повышает эффективность анализа. В рамках ручного анализа доступ к исходному коду позволял выявить критически опасные уязвимости в 100% приложений, а при исследовании методом черного ящика критически опасные уязвимости были обнаружены в 35% систем.

3. ПОРТРЕТ УЧАСТНИКОВ

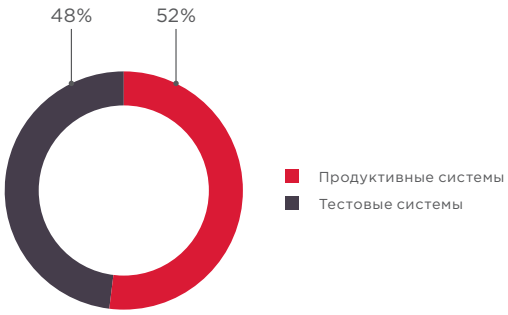
В 2017 году проводился анализ защищенности веб-приложений, представляющих компании из различных сфер деятельности — финансовые организации, интернет-магазины, промышленные, телекоммуникационные и IT-компании, государственные учреждения.



Портрет участников

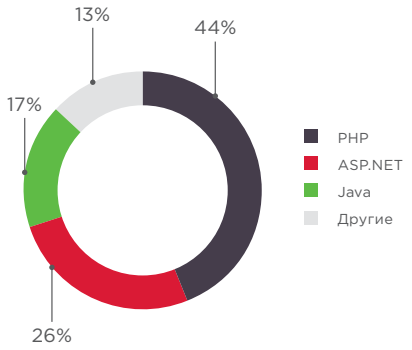
Далее все статистические данные приводятся без привязки к отрасли.

Анализ защищенности осуществлялся как в отношении продуктивных систем (52%), так и тестовых (48%), находящихся в финальной стадии разработки или в стадии приемки в эксплуатацию.



Доли продуктивных и тестовых систем

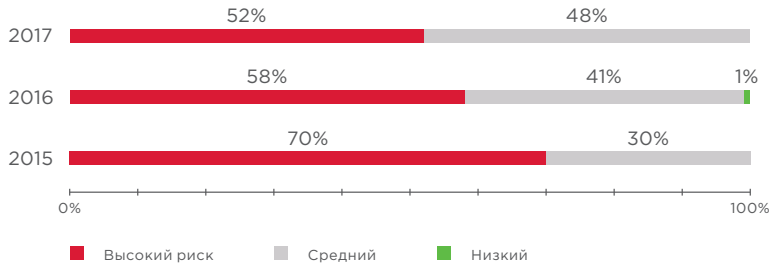
Значительную долю составили веб-приложения, разработанные на базе PHP (44%). Приложения на базе Java и ASP.NET составили 17% и 26% соответственно. Выросла доля приложений, разработанных с использованием других средств разработки — Python, Node.js, Ruby on Rails: с 7% до 13%.



Средства разработки (доли веб-приложений)

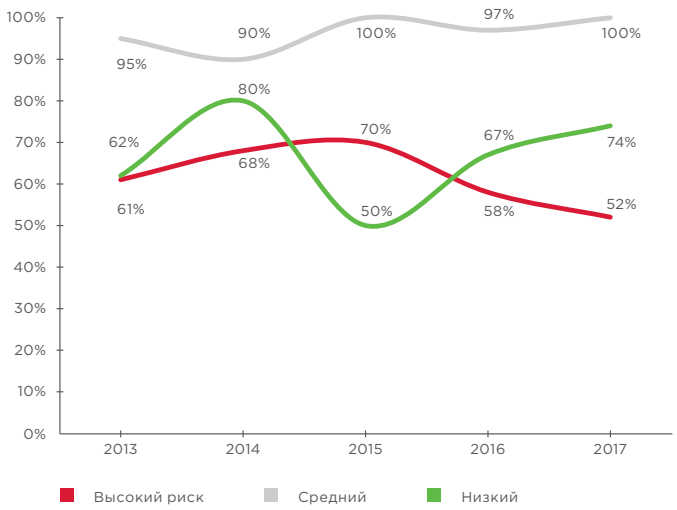
4. ТЕНДЕНЦИИ

Каждое веб-приложение, исследованное в рамках анализа защищенности, содержало уязвимости той или иной степени опасности. При этом наметилась позитивная тенденция: доля веб-приложений, содержащих критически опасные уязвимости, снижается второй год подряд. В 2017 году в 52% приложений были выявлены уязвимости высокой степени риска.



Доля уязвимых сайтов в зависимости от максимальной степени риска уязвимостей

При этом в каждом из исследованных приложений были выявлены уязвимости среднего уровня риска. Этот показатель на протяжении нескольких лет находится в пределах 90–100%. Второй год подряд наблюдается рост доли приложений, содержащих уязвимости низкой степени риска: в 2017 году они обнаружены в 74% исследованных систем.



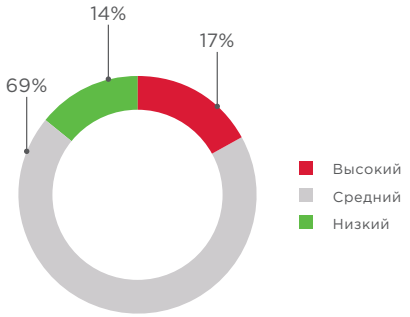
Динамика изменения доли сайтов с уязвимостями различной степени риска

В ходе работ по анализу защищенности в 2017 году в 21% веб-приложений было выявлено использование устаревшего ПО, которое содержало уязвимости различной степени риска вплоть до критической. Отметим, что данные уязвимости в настоящей статистике не учитывались, так как не относятся непосредственно к уязвимостям веб-приложения, однако их эксплуатация может привести к реализации серьезных угроз в отношении атакуемой системы. Наиболее часто эксперты выявляли использование устаревших версий веб-серверов, а также систем управления содержимым. Например, в ходе исследования одного из сайтов было установлено, что любой внешний злоумышленник мог воспользоваться публичным эксплойтом для проведения атаки типа «Отказ в обслуживании»[1], так как в системе использовалась устаревшая версия веб-сервера Apache, содержавшая уязвимость 2011 года CVE-2011-3192. Другой пример: в приложении был выявлен факт использования уязвимой версии ПО ImageMagick для конвертации изображений в личном кабинете. В ходе работ эксперты с помощью общедоступного инструмента[2] произвели эксплуатацию уязвимости CVE-2017-15277, позволяющую получить часть памяти операционной системы на удаленном узле. В результате были получены сведения о каталогах и файлах в атакуемой системе.

5. РУЧНОЙ АНАЛИЗ ЗАЩИЩЕННОСТИ ВЕБ-ПРИЛОЖЕНИЙ

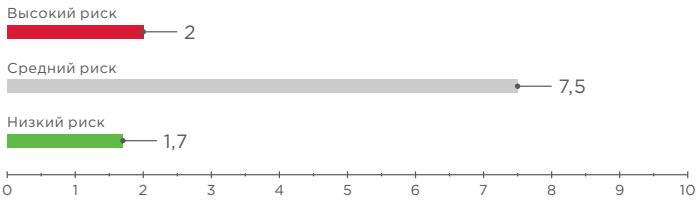
В 2016 году мы отмечали, что доля критически опасных уязвимостей значительно снизилась, но в этом году положительная тенденция не сохранилась, доля таких уязвимостей выросла и составила 17%. Большую часть (69%) вновь составили уязвимости среднего уровня опасности, и 14% были классифицированы как уязвимости низкой степени риска.

1 exploit-db.com/exploits/17696/
2 github.com/neex/gifoeb



Доля уязвимостей различной степени риска

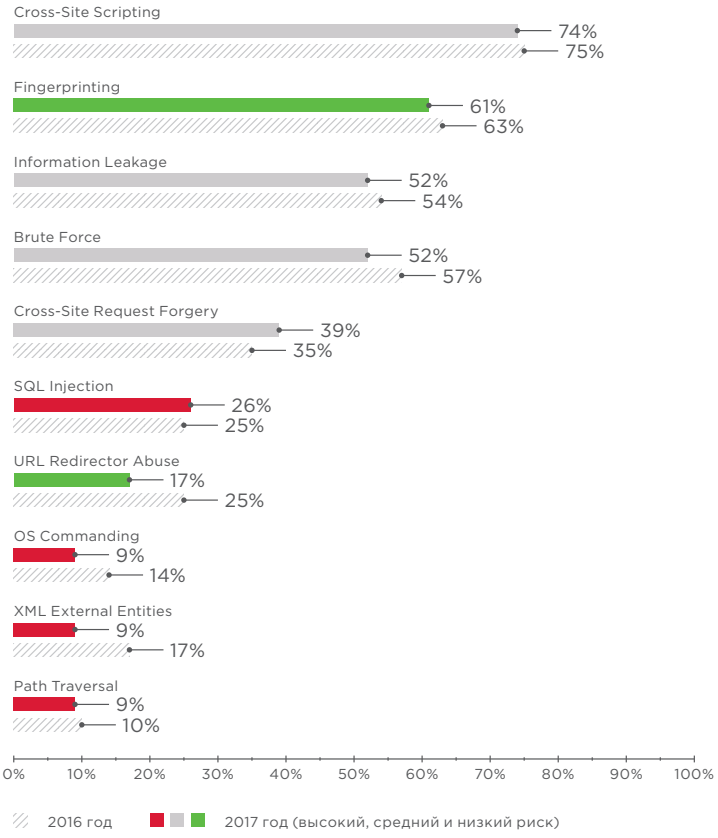
На прежнем уровне осталось среднее число критически опасных уязвимостей, приходящихся на одно веб-приложение, — 2 против 2,1 в 2016 году. Несмотря на то, что в этом году во всех приложениях были выявлены уязвимости средней степени опасности, среднее их количество снизилось с 17,3 до 7,5 на одну систему. С прошлого года количество уязвимостей низкой степени опасности практически не изменилось (1,7 против 1,8).



Среднее число уязвимостей на одну систему

5.1. Наиболее распространенные уязвимости

На рисунке ниже представлен рейтинг уязвимостей, которые чаще других наши эксперты выявляли в ходе работ по ручному анализу защищенности веб-приложений. Стоит отметить, что самые распространенные уязвимости сохранили свои позиции.

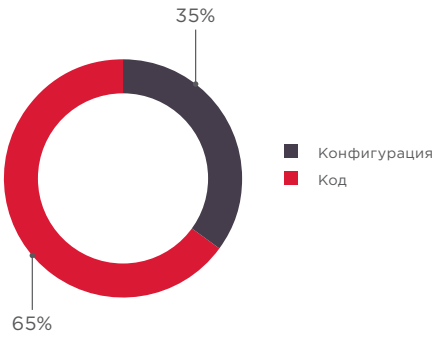


Наиболее распространенные уязвимости, выявленные в рамках ручного тестирования (доля систем)

Регулярно на первой строчке этого рейтинга находится уязвимость среднего уровня риска «Межсайтовое выполнение сценариев» (Cross-Site Scripting): она была выявлена в 74% систем. Стоит отметить, что среди наиболее распространенных присутствуют и другие уязвимости, позволяющие атаковать пользователей веб-приложений, в том числе «Подделка межсайтового запроса» (Cross-Site Request Forgery) и «Открытое перенаправление» (URL Redirector Abuse). В рейтинге присутствуют сразу четыре уязвимости высокой степени опасности. В каждом четвертом приложении эксперты демонстрировали возможность эксплуатации уязвимости «Внедрение SQL-кода»: злоумышленник мог получить чувствительную информацию из СУБД, включая учетные данные пользователей. В 9% приложений выявлялись такие опасные уязвимости, как «Выполнение произвольного кода» (OS Commanding), «Внедрение внешних сущностей XML» (XML External Entities), «Выход за пределы назначенного каталога» (Path Traversal).

Уязвимости серверной и клиентской частей приложения разделились поровну. Среди серверных уязвимостей, например, «Утечка информации», «Недостаточная защита от атак, направленных на перехват данных». К уязвимостям клиентской части относились, среди прочих, «Межсайтовое выполнение сценария», «Подделка межсайтового запроса».

Значительная доля обнаруженных ошибок (65%) были допущены при разработке приложения и содержатся в программном коде систем; некорректные параметры конфигурации веб-серверов составили около трети от общего числа недостатков безопасности.

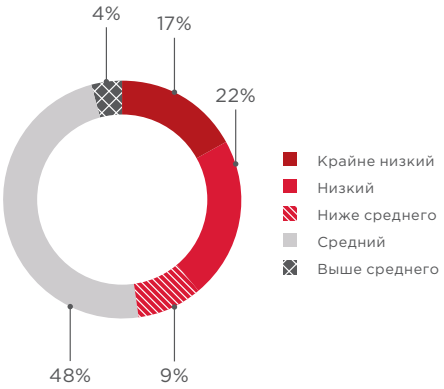


Источник уязвимостей

Отметим, что большинства выявленных проблем в безопасности веб-приложений можно было избежать. Для этого необходимо придерживаться принципов безопасной разработки при создании веб-приложения, включая анализ защищенности кода еще в процессе его написания.

5.2. Анализ угроз и уровней защищенности

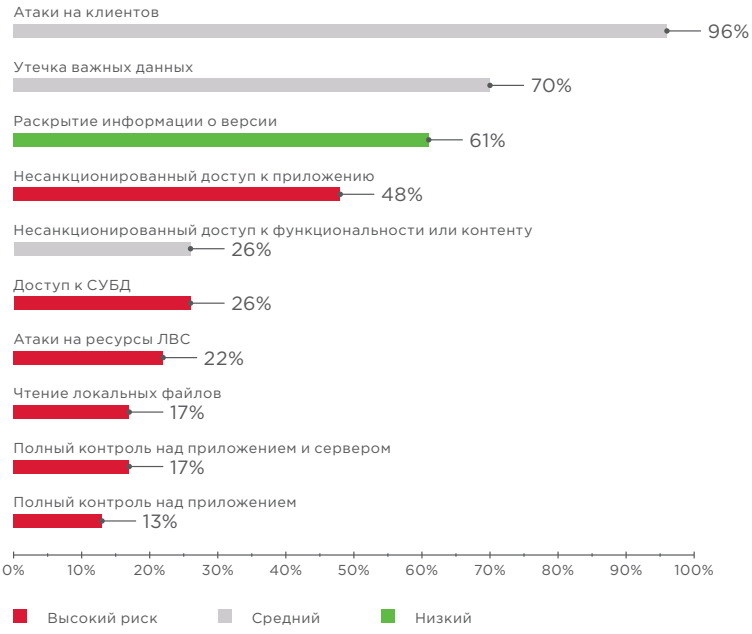
В рамках работ по анализу веб-приложений оценивается уровень защищенности каждой системы: от крайне низкого до приемлемого. Под крайне низкой степенью защищенности мы понимаем наличие множественных критически опасных уязвимостей, например позволяющих выполнять команды ОС сервера любому внешнему злоумышленнику или приводящих к разглашению особо чувствительной информации. В зависимости от числа критически опасных уязвимостей и сложности их эксплуатации уровень защищенности системы может варьироваться от крайне низкого до уровня ниже среднего. В 2017 году почти половина приложений (48%) были оценены как среднезащищенные. Оценку уровня защищенности от «ниже среднего» до «крайне низкого» получили 48% систем; аналогичный показатель в 2016 году составил 56%. Но при этом для 17% приложений уровень защищенности был оценен экспертами как крайне низкий, что соответствует уровню прошлого года (16%). Ни для одного из рассматриваемых веб-приложений уровень защищенности не получил оценку «приемлемый». Общий уровень защищенности веб-приложений по-прежнему остается достаточно низким.



Уровень защищенности веб-приложений

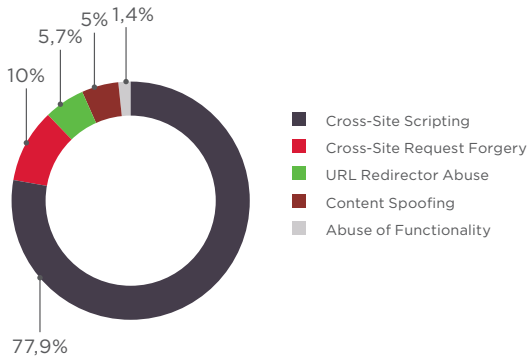
Распределение реализуемых угроз сохранило тенденцию 2016 года. Атаки на пользователей веб-приложения могут быть совершены в 96% систем. Выросла доля потенциальных утечек критически важной информации, в том числе персональных данных. Несанкционированный доступ к приложению может быть получен примерно в каждом втором случае (48%). Каждое четвертое приложение может стать вектором проникновения во внутреннюю сеть.

Отдельно отметим такие угрозы, как «Полный контроль над приложением и сервером» и «Полный контроль над приложением». Для реализации первой необходимо было выявить уязвимость, эксплуатация которой позволяет выполнять команды на сервере атакуемого приложения, например «Выполнение произвольного кода» или «Загрузка произвольных файлов». Такие недостатки удалось выявить в 17% приложений. Под «Полным контролем над приложением» подразумевалась возможность получения максимальных привилегий в системе без получения контроля над сервером компании, например вследствие хранения доступной копии базы данных с учетной записью администратора веб-приложения в открытом виде. Недостатки такого рода были обнаружены в 13% систем.



Наиболее распространенные угрозы (доля систем)

Отдельно рассмотрим уязвимости, позволяющие реализовать атаки на пользователей веб-приложений. Значительную долю составили уязвимости типа «Межсайтовое выполнение сценариев» (77,9%). Также были выявлены «Подделка межсайтового запроса» (10%), «Открытое перенаправление» (5,7%). Как мы уже отмечали, эти уязвимости входят в топ-10 самых распространенных уязвимостей 2017 года.



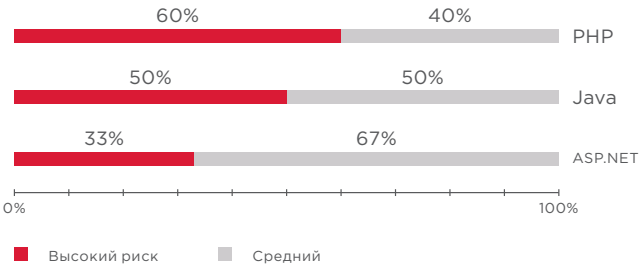
Соотношение уязвимостей, позволяющих проводить атаки на пользователей

В 4% веб-приложений эксперты демонстрировали доступ к исходному коду веб-приложений. Получение исходного кода позволяет злоумышленнику выявлять другие уязвимости атакуемой системы, планировать и проводить новые атаки.

Доля систем, в которых возможен доступ к персональным данным, значительно выросла по сравнению с 2016 годом. Эксперты демонстрировали получение персональных данных в 44% приложений, которые обрабатывают пользовательскую информацию. Такие данные представляют высокую ценность для нарушителей и могут использоваться при планировании и проведении социотехнических атак на пользователей.

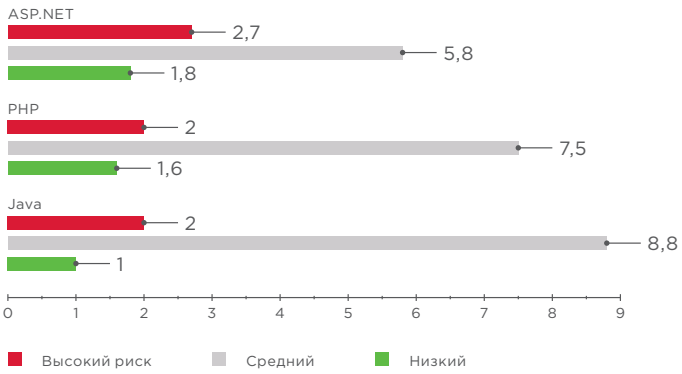
5.3. Анализ различных средств разработки

60% систем, разработанных на PHP, содержали критически опасные уязвимости. Для веб-приложений, созданных с использованием технологий Java и ASP.NET, это значение оказалось несколько ниже — 50% и 33% соответственно. Таким образом, третий год подряд сохраняется тенденция на снижение доли систем с критически опасными уязвимостями, разработанных с использованием технологий Java и ASP.NET.



Доля веб-приложений по максимальному уровню риска уязвимостей

Среднее количество уязвимостей высокой степени опасности выросло для технологии ASP.NET с 2,1 до 2,7. На каждое приложений на PHP и Java в среднем пришлось по 2 уязвимости высокой степени опасности (против 2,8 и 2,1 в 2016 году соответственно). Несколько больше уязвимостей средней степени опасности, как и в прошлом году, в среднем выявлялось в приложениях, разработанных с использованием технологии Java. Наименьшее количество подобных недостатков было снова обнаружено в приложениях на ASP.NET.



Среднее число уязвимостей на одну систему

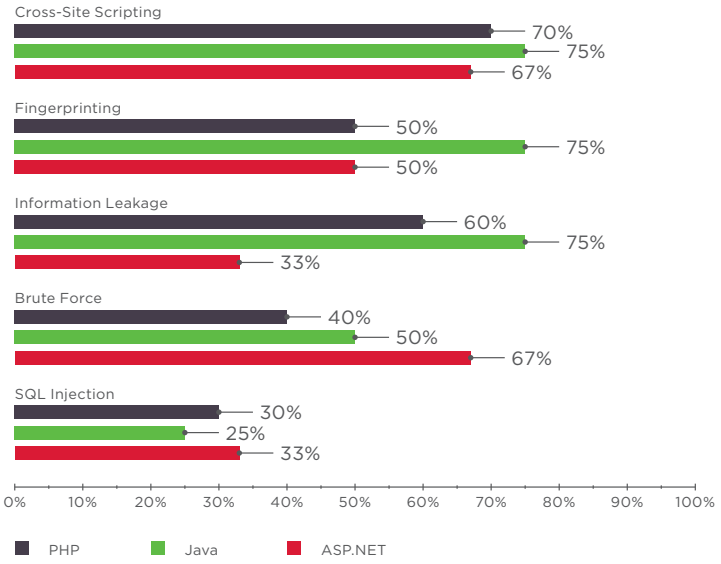
Наиболее распространенные уязвимости (по средствам разработки)

PHP	Доля сайтов	Java	Доля сайтов	ASP.NET	Доля сайтов
Cross-Site Scripting	70%	Cross-Site Scripting	75%	Cross-Site Scripting	67%
Information Leakage	60%	Fingerprinting	75%	Brute Force	67%
Fingerprinting	50%	Information Leakage	75%	Fingerprinting	50%
Brute Force	40%	Brute Force	50%	Cross-Site Request Forgery	50%
SQL Injection	30%	Insufficient Authorization	50%	SQL Injection	33%
Cross-Site Request Forgery	30%	Content Spoofing	50%	URL Redirector Abuse	33%
Application Misconfiguration	20%	XML External Entities	25%	Insufficient Authorization	17%
OS Commanding	10%	SQL Injection	25%	XML External Entities	17%
URL Redirector Abuse	10%	Cross-Site Request Forgery	25%	OS Commanding	17%
Path Traversal	10%	Deserialization of Untrusted Data	25%	Insecure indexing	17%

В таблице выше представлен рейтинг самых распространенных уязвимостей веб-приложений в зависимости от средств разработки.

Уязвимость «Межсайтовое выполнение сценариев» по-прежнему является наиболее распространенной для всех языков программирования, ей подвержены от 67% до 75% ресурсов в соответствующих категориях. Широко распространены независимо от технологии разработки такие уязвимости, как «Утечка информации», «Раскрытие информации о версии ПО», «Недостаточная защита от подбора учетных данных».

Среди уязвимостей, отмеченных как критически опасные, в десятку наиболее распространенных вошли «Внедрение SQL-кода» (вне зависимости от языка разработки), «Внедрение внешних сущностей XML» (Java, ASP.NET), «Выполнение произвольного кода» (PHP, ASP.NET), а также «Выход за пределы назначенного каталога» (PHP), «Десериализация недоверенных данных» (Java) и «Недостаточная авторизация» (ASP.NET).

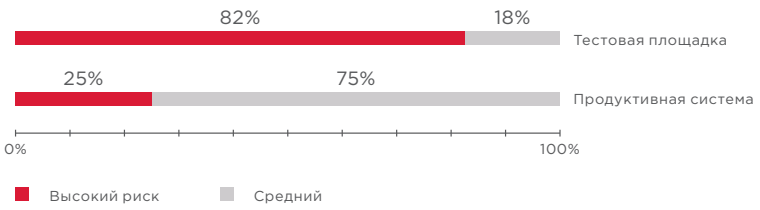


Доли веб-приложений, подверженных распространенным уязвимостям

Независимо от средства разработки почти в каждом из рассмотренных приложений были выявлены уязвимости из числа наиболее распространенных.

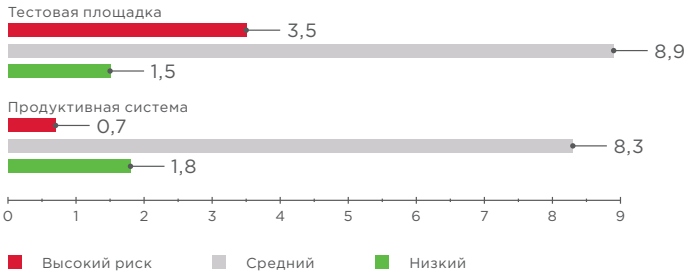
5.4. Сравнение тестовых и продуктивных систем

В рамках исследования была проведена оценка защищенности тестовых и продуктивных веб-приложений. Системы, еще не запущенные в эксплуатацию, более подвержены уязвимостям высокой степени риска: 82% таких приложений содержали критически опасные недостатки информационной безопасности. Для продуктивных систем этот показатель снижается уже три года подряд и составил 25%, что свидетельствует о том, что программисты стали больше уделять внимания безопасной разработке и устранению уязвимостей еще до ввода приложения в эксплуатацию, хотя такие показатели еще нельзя назвать приемлемыми. Будем надеяться на развитие позитивной тенденции.



Доли систем по максимальному уровню риска

Продуктивные системы являются более защищенными и по количеству обнаруженных уязвимостей. Среднее число уязвимостей высокой степени риска на одну систему в 5 раз выше по сравнению с продуктивными (3,5 против 0,7). Отметим, что для большинства тестовых систем анализ защищенности проводился методом белого ящика, который более эффективен при выявлении недостатков безопасности.



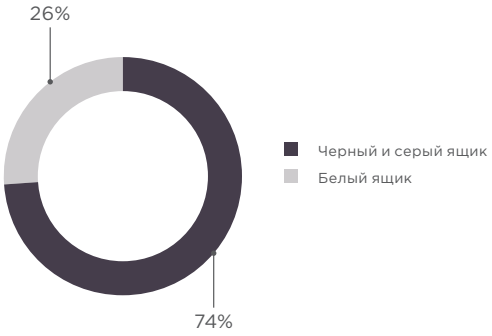
Среднее количество уязвимостей на одну систему

Как в продуктивных, так и в тестовых системах присутствует множество опасных уязвимостей. И важно, чтобы процесс обеспечения безопасной разработки внедрялся с самого начала, а не представлял собой только исправление выявленных недостатков непосредственно перед вводом системы в эксплуатацию.

5.5. Сравнение методов тестирования

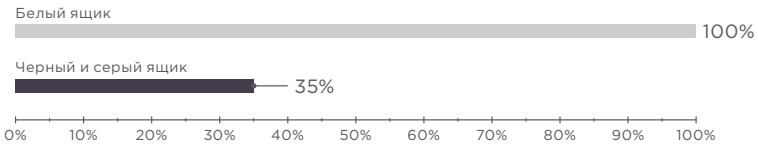
Большинство проектов в 2017 году проводились с использованием методов черного и серого ящика (74%), для каждого четвертого проекта был предоставлен исходный код — для анализа методом белого ящика.

Важно отметить, что в ходе работ по одному из проектов было выявлено несколько десятков критически опасных уязвимостей, в том числе «Внедрение SQL-кода», а в рамках другого — несколько сотен недостатков конфигурации средней степени опасности. Результаты данных работ не были учтены при подсчетах, так как существенно искажают усредненные статистические данные для всех исследованных приложений.



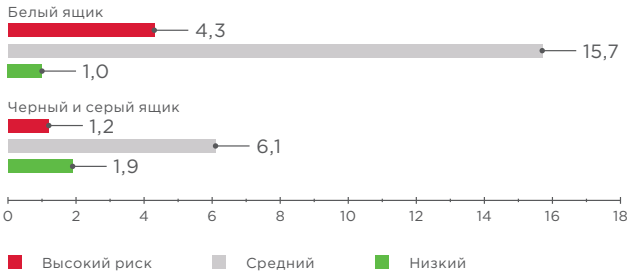
Доля приложений по методу тестирования

Сравнительный анализ методов тестирования черным (серым) и белым ящиком третий год подряд подтверждает более высокую эффективность последнего. Если уязвимости средней степени риска были выявлены во всех системах независимо от метода исследования, то для систем, где исходный код был недоступен, критически опасные уязвимости были выявлены в 35% систем против 100% для метода белого ящика. Таким образом, отсутствие исходного кода у злоумышленника не гарантирует защищенность системы, но вот его наличие значительно повышает вероятность обнаружения и последующей эксплуатации критически опасных уязвимостей.



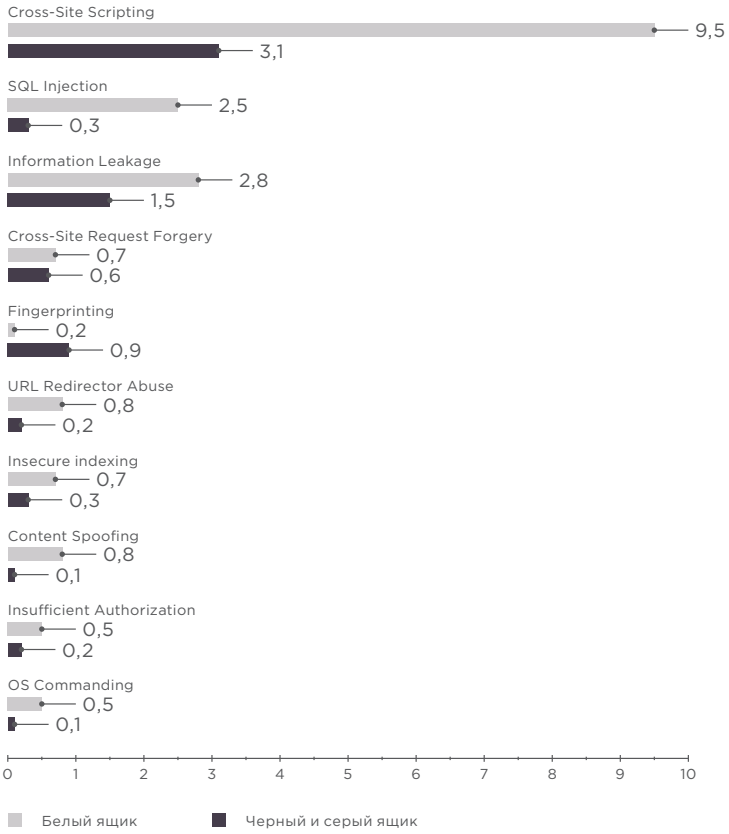
Доли систем, в которых обнаружены уязвимости высокой степени риска

Среднее количество обнаруженных уязвимостей на одну систему при отсутствии исходного кода составило 1,2 в 2017 году, в то время как методом белого ящика эксперты выявляли 4,3 уязвимости на одно веб-приложение. Также при наличии исходного кода удавалось выявить 15,7 уязвимости средней степени опасности, а при их отсутствии — 6,1.



Число обнаруженных уязвимостей на одну систему

При наличии исходного кода эксперты в среднем на одну систему обнаруживали в восемь раз больше уязвимостей типа «Внедрение SQL-кода» и в пять раз чаще выявляли возможность эксплуатации «Выполнение произвольного кода». Кроме того, методом белого ящика в среднем на одну систему было выявлено больше уязвимостей типа «Межсайтовое выполнение сценариев», «Утечка информации», «Открытое перенаправление», «Недостаточная авторизация» и других.



Число выявленных уязвимостей на одну систему

ЗАКЛЮЧЕНИЕ

Подводя итоги, стоит отметить, что по-прежнему общий уровень защищенности веб-приложений остается низким. В каждом приложении присутствуют недостатки разной степени опасности. Злоумышленники могут эксплуатировать критически опасные уязвимости более чем в половине приложений, получая доступ к чувствительным данным, возможность выполнения команд на сервере и полный контроль над системой. Успешные атаки могут проводиться в отношении компаний из самых разных сфер экономики, от интернет-магазинов до государственных предприятий. Доля приложений, в которых конечные пользователи не попадают под угрозу, составляет всего 4%. При этом в результате атаки на веб-приложение в каждом четвертом случае злоумышленник может получить персональные данные, которые сами по себе представляют высокую ценность.

Отдельно отметим, что могут проводиться атаки на веб-ресурсы с целью заражения пользователей вредоносным ПО. Так действовали распространители вируса-шифровальщика Bad Rabbit: злоумышленники взламывали веб-приложения СМИ и маскировали распространяемый файл под обновление для Adobe Flash Player. Другой пример это группировка Cobalt[3], которая успешно атаковала через веб-приложения инфраструктуру контрагентов для последующего проведения фишинговых рассылок на целевые банки. Таким образом, под ударом может оказаться любое приложение, даже то, которое не является целью атакующего: оно может стать лишь промежуточным звеном в атаке. Что же тогда говорить о тех сайтах, которые критически важны для компаний, где атаки на клиентов могут привести не только к репутационным, но и к финансовым потерям!..

Исходя из полученных данных, мы отмечаем необходимость регулярного проведения работ по анализу защищенности веб-приложений. При этом рекомендуется использовать метод белого ящика (с анализом исходного кода): его эффективность выше, чем у других методов тестирования. Можно избежать значительных

3 ptsecurity.com/upload/corporate/ru-ru/analytics/Cobalt-2017-rus.pdf

затрат на переработку приложения в случае выявления серьезных уязвимостей в предрелизной версии — если проводить анализ защищенности кода еще в процессе его написания (Secure Software Development Life Cycle[4]).

Также для защиты от атак на веб-приложения рекомендуется применять превентивные меры защиты, такие как межсетевой экран уровня приложений (web application firewall, WAF). При этом WAF должен не только обнаруживать и предотвращать известные атаки на уровне приложения и бизнес-логики, но и выявлять эксплуатацию уязвимостей нулевого дня, предотвращать атаки на пользователей, анализировать и сопоставлять множество событий для выявления цепочек атак, что возможно только при использовании инновационных технологий нормализации, эвристического и поведенческого анализа и самообучения.

Вопросам безопасности по-прежнему уделяется недостаточное внимание, ежегодно мы наблюдаем одни и те же ошибки, совершаемые разработчиками и администраторами систем, а вопрос внедрения процессов обеспечения безопасности на протяжении всего жизненного цикла веб-приложений по-прежнему остается нерешенным.

4 owasp.org/index.php/OWASP_Secure_Software_Development_Lifecycle_Project

О компании

Positive Technologies — один из лидеров европейского рынка систем анализа защищенности и соответствия стандартам, а также защиты веб-приложений. 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.

Деятельность компании лицензирована Минобороны России, ФСБ России и ФСТЭК России, продукция сертифицирована Минобороны России и ФСТЭК России.

POSITIVE TECHNOLOGIES

pt@ptsecurity.com ptsecurity.com facebook.com/PositiveTechnologies facebook.com/PHDays