Эшелон
комплексная безопасность
http://эшелон.рф
mail@cnpo.ru

БЕЗОПАСНОСТЬ ПРОГРАММНОГО КОДА: 2010 ГОД

Сводный отчёт по безопасности программного обеспечения в России и мире за 2010 год

Содержание

Предисловие 3
1 Распространение языков программирования 4
1.1 Рейтинг языков программирования в мире 4
1.2 Рейтинг языков программирования в России 7
2 Уязвимости программного обеспечения 9
2.1 Общий обзор по уязвимостям 9
2.2 Уязвимости веб-приложений 17
2.3 Рейтинг уязвимостей в приложениях ведущих разработчиков . . . 19
2.4 Распределение уязвимостей по регионам происхождения 25
2.5 Вывод 25
3 Безопасность и языки программирования 26
3.1 Рейтинг уязвимости языков веб-программирования 26
Заключение 29
Авторы 30
О компании ЗАО «НПО «Эшелон» 31
Список использованных источников 32

Предисловие

Компания ЗАО «НПО «Эшелон» представляет сводный отчёт по состоянию безопасности программного обеспечения в России и мире за 2010 год.

Данной публикацией мы планируем открыть серию отчётов по ситуации в отрасли информационной безопасности в целом и отдельных её направлений (например, тестирование приложений) в частности.

Нам интересно поделиться с Вами своим опытом в области исследования безопасности приложений, услышать дополнения, встречные предложения, советы и другие мнения. Это тот процесс, от которого выигрывают все его участники.

Адрес электронной почты для связи: code_audit@cnpo.ru

Сводный отчет был подготовлен сотрудниками департамента программных разработок ЗАО «НПО «Эшелон».

С уважением, редактор Алексей Марков

1 Распространение языков программирования

Безопасность программного обеспечения — это свойство защищенности от угроз реализации уязвимостей всех уровней (слоев) функционирования продуктов. Уязвимости могут быть реализованы как во внешних приложениях, сторонних компонентах, в среде исполнения (интерпретации).

С этой точки зрения наиболее важно рассмотреть распространенность языков и систем программирования и проверить наличие связи между выбором платформы разработки и вероятностью возникновения проблем с безопасностью приложения [3, 5].

Распространение того или иного языка сказывается и на общей картине уязвимостей, выявленных в программном обеспечении, поскольку большинство уязвимостей специфичны для конкретного языка программирования [1].

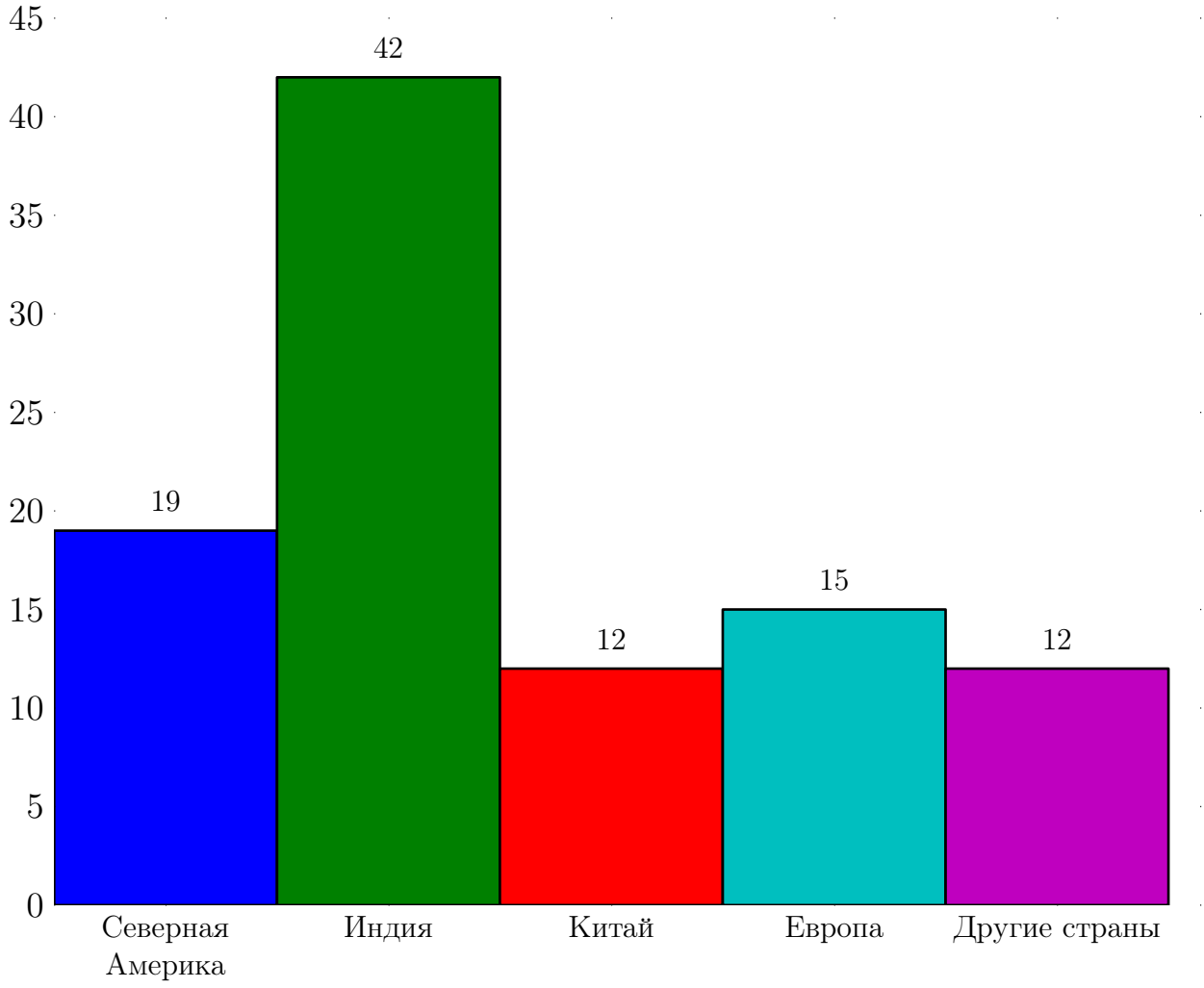
1.1 Рейтинг языков программирования в мире

Известны несколько общепризнанных мировых индексов популярности языков программирования. Каждый из них основывается на своей выборке данных, сформированных определенным образом.

Индекс, доступный на интернет-ресурсе langpop.com, сформирован энтузиастами по целому ряду косвенных признаков, в частности:

- статистике запросов к поисковым системам (поиск решения проблем с определенным языком программирования или поиск работы программистом на определенном языке),
- анализе тематики новых книг, появляющихся в интернет-магазинах (таких как amazon.com,oreally.com),
- статистике появления новых и развития старых проектов с открытым исходным кодом, находящихся на хостингах (таких как freshmeat.net, code.google.com).

Итоговый индекс, получаемый компиляцией всех описанных, приведен ниже (см. рис.1.1).



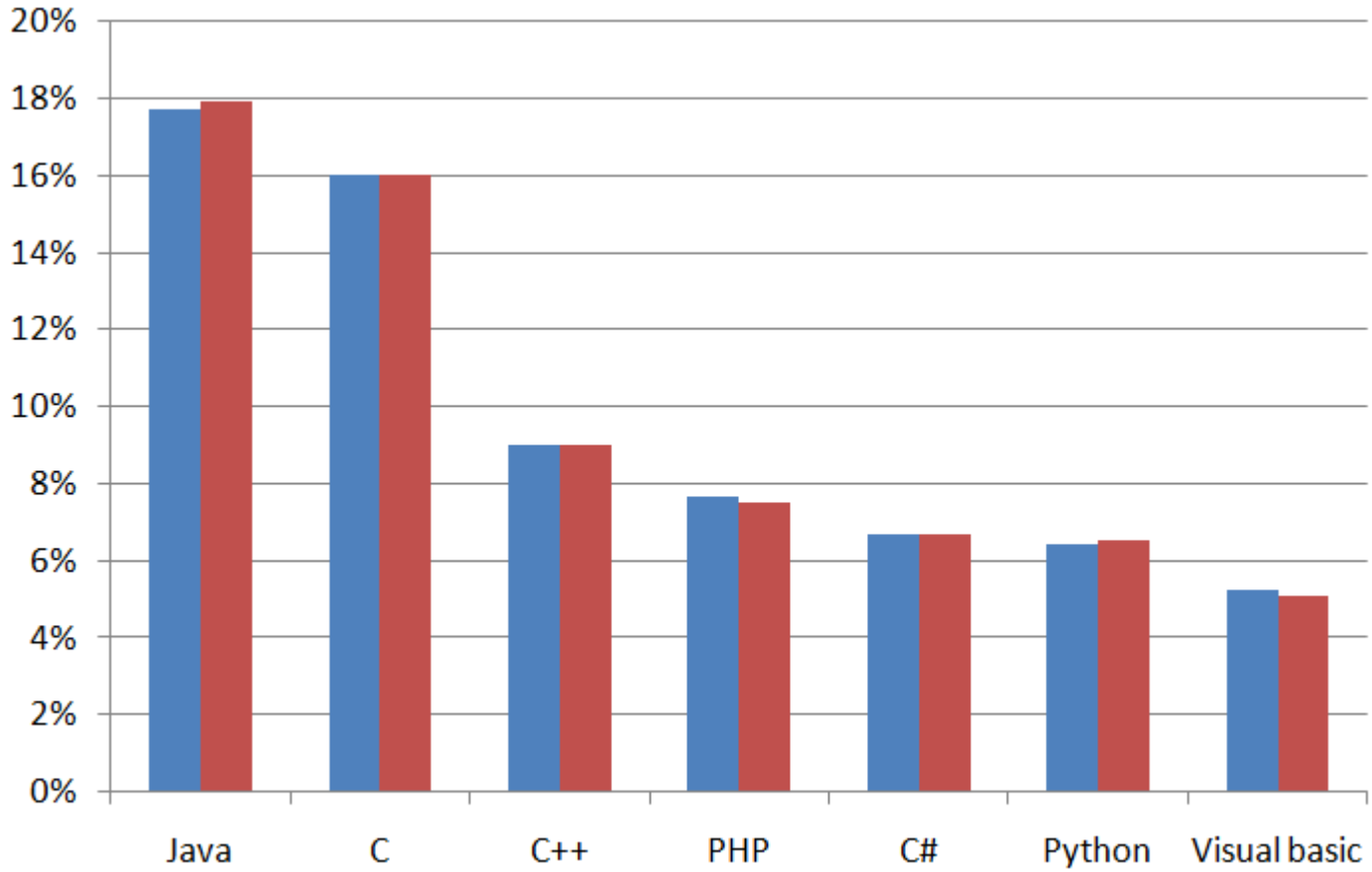
Источник: LangPop

Рис. 1.1. Общемировое распространение языков программирования (%)

Другим источником подобной информации является веб-сайт компании TIOBE Software, специализирующейся на контроле качества исходных текстов. Принципиальным отличием индекса TIOBE является учёт в рейтинге только пол-

ных по Тьюрингу языков программирования[1] (не учитываются такие языки, как SQL, HTML, XML) [7].

Статистические данные о популярности языков программирования представлены на рисунке 1.2).



Источник: TIOBE Software

Рис. 1.2. Общемировое распространение языков программирования (%)

По данным графика 1.2 видно, что за последний год выросла популярность языка Java (вероятно, это связано с продвижением мобильной операционной системы Android, основным языком разработки для которой является Java), хотя

[1] В теории вычислимости исполнитель (множество вычисляющих элементов) называется тьюринг-полным, если на нём можно реализовать любую вычислимую функцию. Другими словами, для каждой вычислимой функции существует вычисляющий её элемент (например, машина Тьюринга) или программа для исполнителя, а все функции, вычисляемые множеством вычислителей, являются вычислимыми функциями (возможно, при некотором кодировании входных и выходных данных).

сама платформа JVM вышла лидеры достаточно давно, ещё с середины 2000-х. Популярность достаточно низкоуровневых языков C и C++ также стабильна и, скорее всего, связана с их эффективностью в системном программировании, устоявшимся сообществом разработчиков и уже сформировавшейся базой курсов в учебных заведениях.

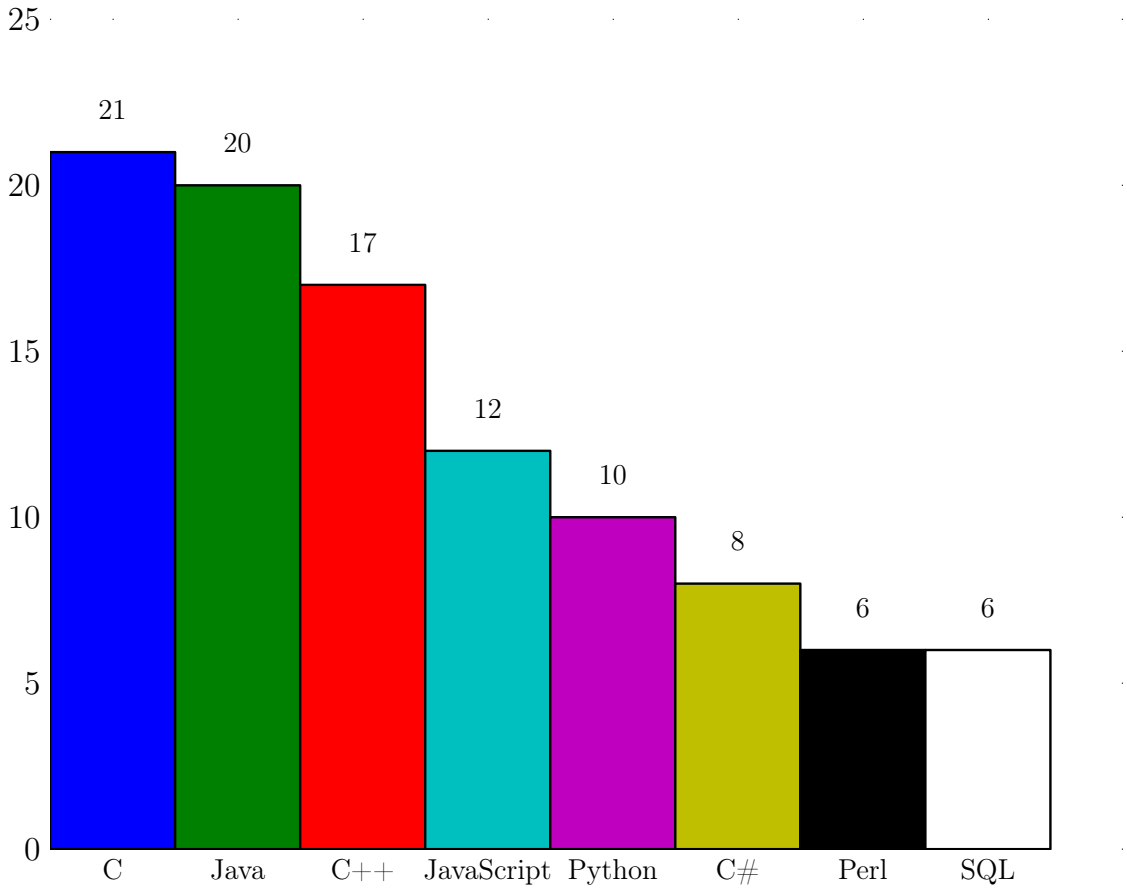
1.2 Рейтинг языков программирования в России

Несомненный интерес представляет положение дел в России. В качестве источника информации будет выступать перечень сертифицируемых продуктов в РФ (по данным испытательных лабораторий) [16].

Используемые данные:

- 73 проведенных сертификационных испытаний и аудита кода за 2010 год;
- 121 проведенное сертификационное испытание и аудит кода за 2007-2009 годы.

Данная выборка является достаточно репрезентативной для того, чтобы в целом рассмотреть популярность языков для разработки программного обеспечения, и отображена на графике (см. рис.1.3).



Источник: ЗАО «НПО «Эшелон»

Рис. 1.3. Распространение языков программирования среди сертифицируемых в России изделий (в %)

Как видно из рисунка, основные программные изделия, сертифицированные в России, разработаны на языках C/C++, Java, C#. Обращает на себя внимание меньшее количество веб-приложений. Это можно объяснить тем, что подавляющее большинство систем, аудит которых проводился испытательной лабораторией, рассчитано на применение внутри крупных организаций федерального уровня.

2 Уязвимости программного обеспечения

2.1 Общий обзор по уязвимостям

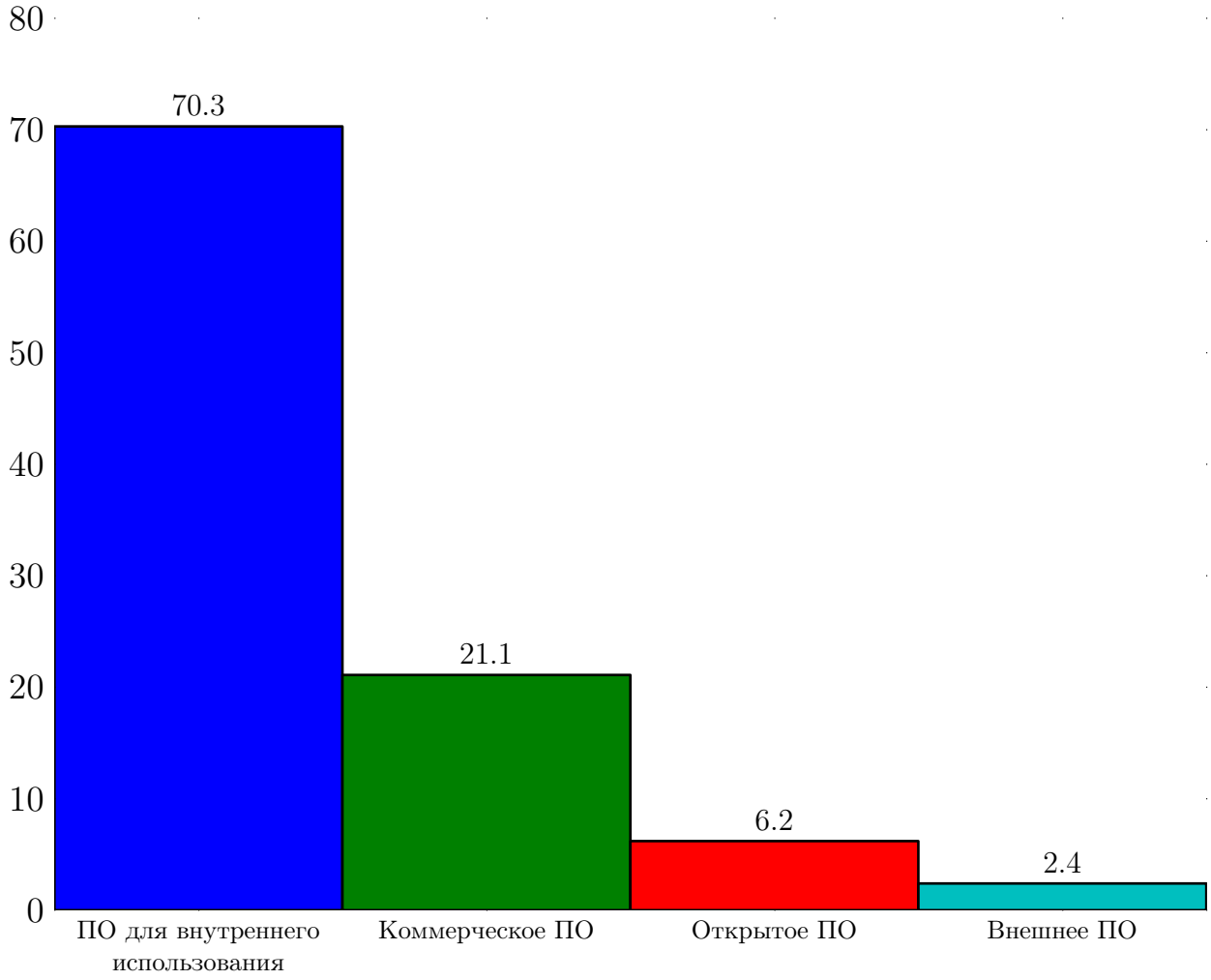
Подробный анализ уязвимостей, обнаруженных в исследуемом программном обеспечении, представлен в отчете компании Veracode [13, 14], которая предоставляет «облачный» сервис по анализу программного обеспечения на наличие уязвимостей [12]. Приведенная ниже статистика используется компанией при создании годового отчета. Согласно данным компании, 58% из программ проверенных Veracode, не имеют приемлемого уровня безопасности.

Программы сторонних производителей для программ корпоративного уровня (это большая часть всех программ),

60% создаваемых программ применяются для внутреннего использования, 30% идут на продажу, а в 10% случаев используют внешнее программное обеспечение: open-sourced[1] (далее — открытое программное обеспечение) или outsourced[2] (далее — внешнее программное обеспечение) (см. рис.2.1).

[1] Open-sourced — программное обеспечение с открытыми исходными текстами.

[2] Outsourced — программное обеспечение, разработанное для компании по контракту с внешними исполнителями.

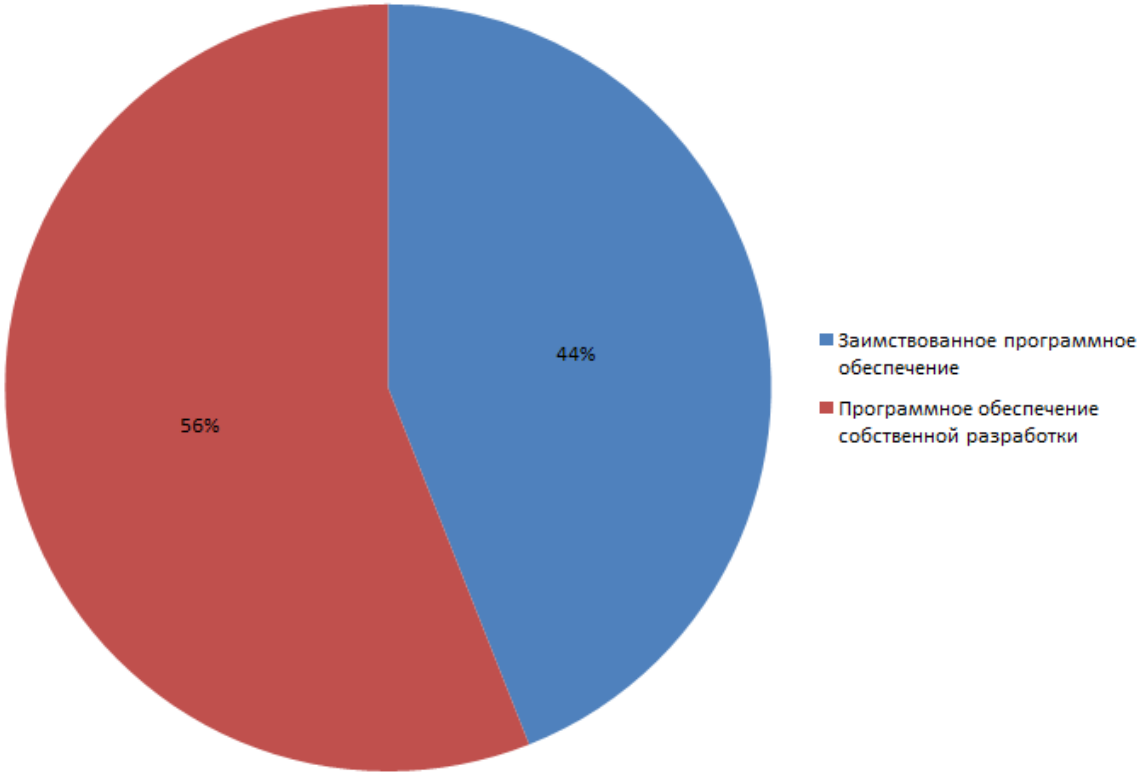


Источник: Veracode

Рис. 2.1. Распределение уязвимостей по типам программных разработок в мире (в %)

Стоит заметить, что даже в собственных программных проектах у компаний используется примерно 30%-70% кода сторонних разработчиков. Кроме того, присутствует так называемый «эффект вложенности», то есть компоненты сторонних разработчиков содержат компоненты других сторонних разработчиков (см. рис. 2.1) [2, 4, 10].

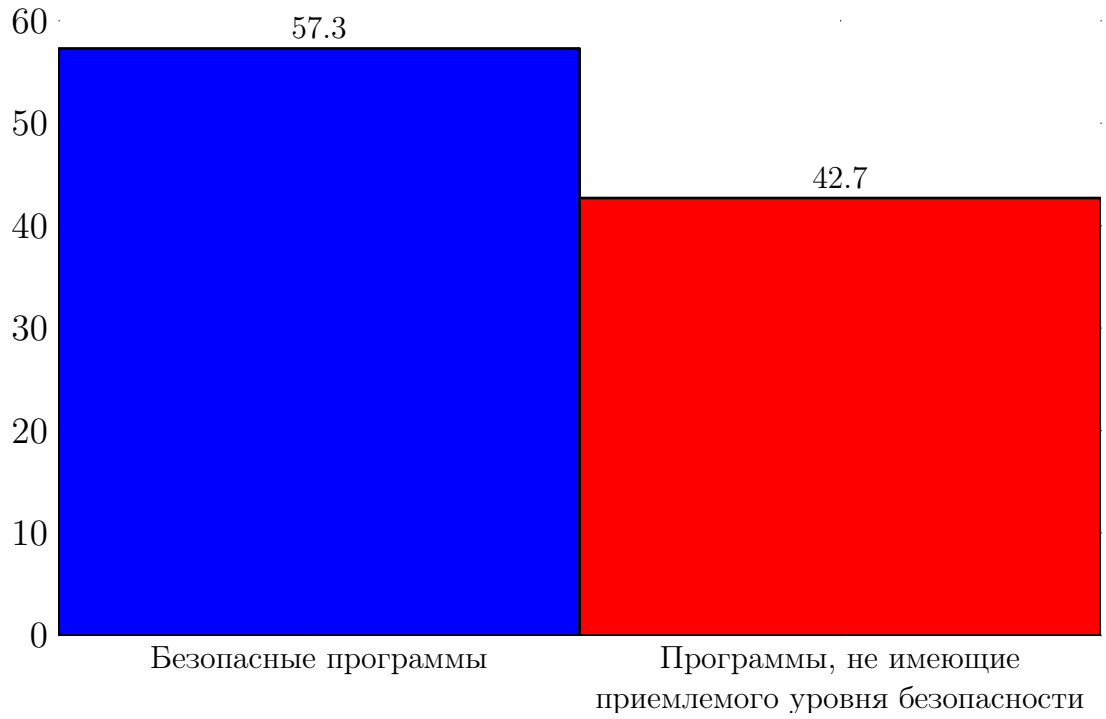
Что характерно, очень близкую к рассмотренной выше картине распределения показало исследование материалов сертификационных испытаний программных изделий в России (см. рис. 2.2).



Источник: ЗАО «НПО «Эшелон»

Рис. 2.2. Соотношение источников кода в сертифицируемом в России программном обеспечении (в %)

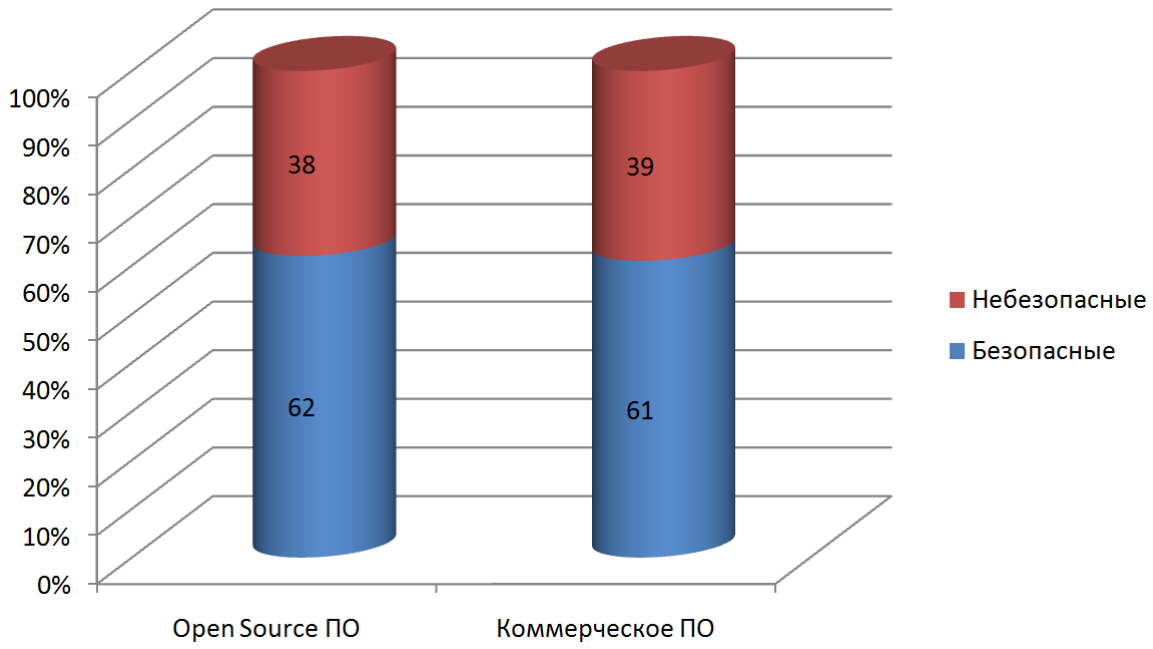
Также необходимо заметить, что значительная часть из проведенных аудитов программного обеспечения дают отрицательный результат (т.е. продукт не прошел проверку). Рассмотрим, к примеру, результаты компании Veracode.



Источник: Veracode

Рис. 2.3. Общее распределение программного обеспечения в области безопасности (в %)

Проекты с открытым кодом имеют в целом сравнимый с коммерческим программным обеспечением уровень безопасности, но у них в среднем меньше время восстановления и потенциальных возможностей для обхода безопасности (например backdoors), чем у коммерческого программного обеспечения или у внешнего программного обеспечения.



Источник: Veracode

Рис. 2.4. Сравнение уровня безопасности в открытом и коммерческом программном обеспечении (в %)

Таблица 2.1

Распределение языков по типам программных разработок

	C/C++	**Java**	**.Net**	**Другие платформы**
Внутреннее программное обеспечение	11%	56%	33%	2%
Коммерческое программное обеспечение	29%	45%	24%	3%
Открытое программное обеспечение	51%	45%	4%	0%
Внешнее программное обеспечение	0%	81%	14%	5%

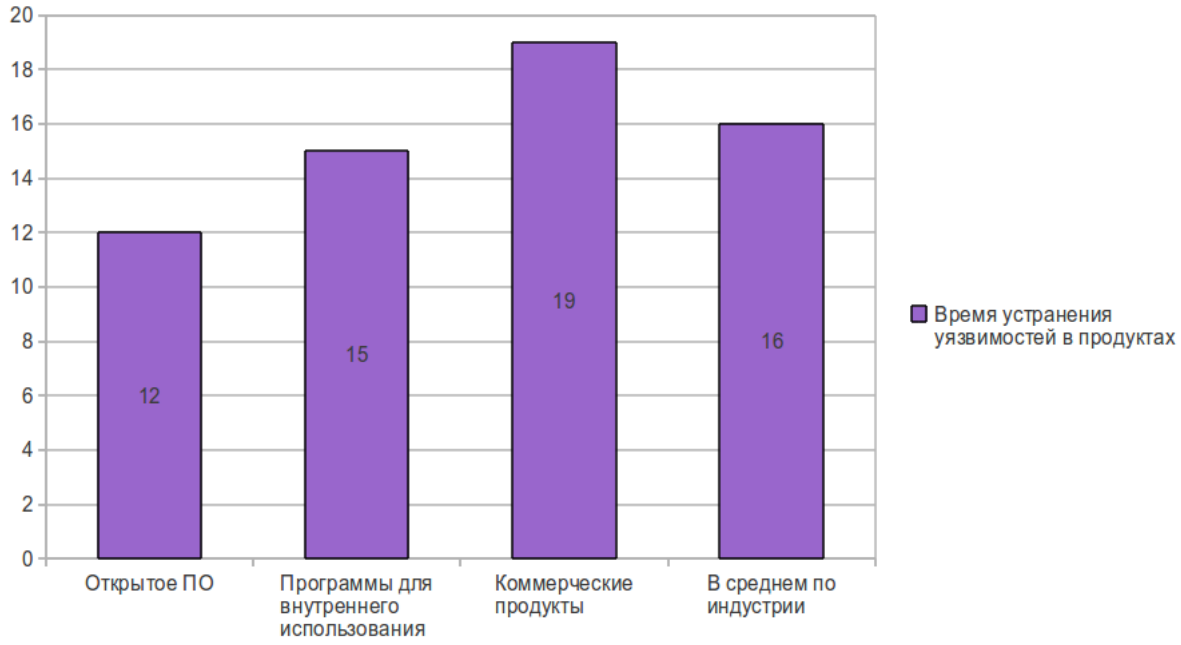
Таблица 2.2

Распределение языков по среде функционирования

	Среда: веб-приложения	**Среда: автономные приложения**
Внутреннее программное обеспечение	61%	39%
Коммерческое программное обеспечение	36%	65%
Открытое программное обеспечение	29%	71%
Внешнее программное обеспечение	71%	29%

Исследование безопасности программных продуктов показало, что в 38% открытых продуктов были обнаружены незакрытые уязвимости, что несколько меньше уязвимостей, найденных в коммерческих проектах [1].

После нахождения уязвимостей, для их исправления в проектах с открытым программным обеспечением потребовалось в среднем 36 дней, 48 дней заняла та же операция для внутренних приложений, и 82 дня — для исправления уязвимостей в коммерческих продуктах.



Источник: Veracode

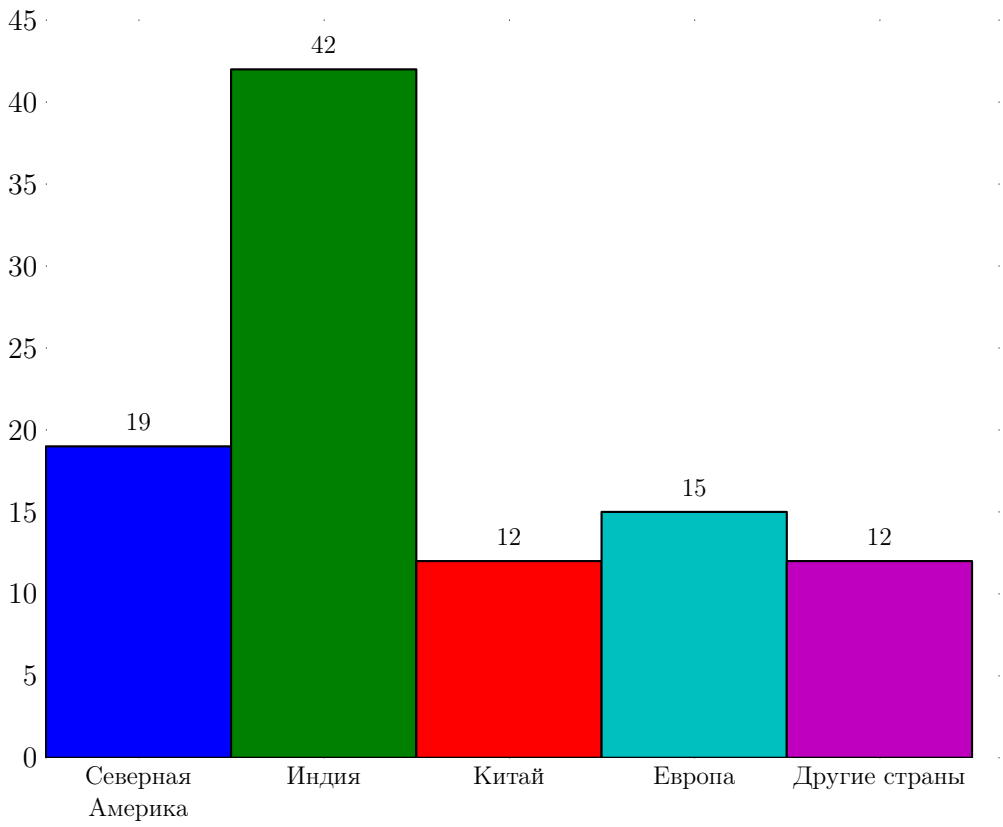
Рис. 2.5. Время устранения уязвимостей в зависимости от типа программного обеспечения (в днях)

Более уязвимыми оказались продукты, написанные на языках C и C++, в первую очередь из-за специфических недостатков прямого использования памяти, таких как переполнение буфера, переполнение переменных (после использования вызовов типа free). Рейтинг уязвимостей по причинам возникновения представлен ниже (см. табл.2.3).

Таблица 2.3

Распределение уязвимостей по типам программных разработок

Внутреннее программное обеспечение		**Коммерческое программное обеспечение**		**Открытое программное обеспечение**		**Внешнее программное обеспечение**	
Межсайтовый скриптинг (XSS)	49%	Межсайтовый скриптинг (XSS)	56%	CRLF-атаки	15%	Межсайтовый скриптинг (XSS)	31%
CRLF-атаки	14%	Утечка информации	16%	Числовые ошибки	14%	Атака на файловую систему	16%
Утечка информации	10%	CRLF-атаки	6%	Ошибки управления буфером	14%	Криптографические атаки	
Криптографические атаки	6%	Криптографические атаки	5%	Межсайтовый скриптинг (XSS)	13%	Ошибки времени и локализации	12%
SQL-атаки	5%	Атака на файловую систему	4%	Криптографические атаки	12%	Утечка информации	9%
Атака на файловую систему	3%	SQL-атаки	4%	Управление ошибками	9%	Управление правами доступа	8%
Переполнение буфера	3%	Переполнение буфера	2%	Переполнение буфера	9%	Неправильное использование API	6%
Потенциальные лазейки	2%	Потенциальные лазейки	2%	Ошибки времени и локализации	4%	CRLF-атаки	3%
Ненадежный поиск путей	2%	Арифметические ошибки	2%	Атака на файловую систему	4%	SQL-атаки	2%
Ошибки времени и локализации	2%	Управление ошибками	2%	Потенциальные лазейки	1%	Недостаточные проверки входных данных	<1%
Управление ошибками	1%	Ошибки времени и локализации	1%	SQL-атаки	1%	Управление ошибками	<1%
Инкапсуляция	1%	Управление правами доступа	<1%	Утечка информации	1%	Фиксация сессии	<1%
Управление правами доступа	<1%	Ошибки управления буфером	<1%	Неправильное использование API	<1%	Атаки командами ОС	<1%
Неправильное использование API	<1%	Неправильное использование API	<1%	Управление правами доступа	<1%	Другие	<1%
Недостаточные проверки входных данных	<1%	Атаки командами ОС	<1%	Фиксация сессии	<1%	Состояние гонки	<1%



Источник: Veracode

Рис. 2.6. Распределение уязвимого кода по странам и регионам его происхождения (в %)

2.2 Уязвимости веб-приложений

По данным компании IBM [6], в первом полугодии прошедшего года был отмечен стремительный рост количества ссылок на вредоносные веб-ресурсы, также резко увеличилось число случаев онлайнового финансового мошенничества, называемого «фишингом» (fishing).

Отмечен рост и числа уязвимостей программ для просмотра и редактирования документов, в первую очередь документов формата Portable Document Format (PDF).

Среди других тенденций и рисков ИТ-безопасности, отмеченных в отчете X-Force можно указать следующие [6]:

- Число новых уязвимостей уменьшилось, но пока еще остается на рекордно высоких уровнях. В целом была выявлена 6601 уязвимость, что на 11% меньше, чем, например в 2008 году. В отчете отмечается снижение числа новых уязвимостей в таких распространенных категориях, как SQL-инъекция и ActiveX (мобильный код Internet Explorer). Эта тенденция может указывать на то, что некоторые из наиболее легко обнаруживаемых уязвимостей в этих категориях были устранены, что привело к повышению общего уровеня безопасности.
- В ряде ключевых категорий значительно уменьшилось число опасных и критических уязвимостей, не устраненных с помощью программных «заплаток» (patch). Количество уязвимостей в веб-браузерах и программах просмотра/редактирования документов, в целом, снизилось, что свидетельствует о большей активности поставщиков в борьбе с угрозами информационной безопасности.
- Существенно выросло число случаев выявления уязвимостей в программах просмотра и редактирования документов и в мультимедийных приложениях. При этом было обнаружено более чем на 50% больше уязвимостей указанных категорий, чем в предыдущем году.
- Категория "уязвимости веб-приложений"продолжает быть одной из самых больших. Число уязвимостей веб-приложений, выявляемых организациями, не уменьшается, и они не становятся менее опасными. 49% всех уязвимостей связаны с веб-приложениями, причем, по числу обнаруженных уязвимостей, технология взлома межсайтовый скриптинг (Cross-Site Scripting, также известная как XSS-атака, когда атакующий пытается внедрить на стороне клиента скрипт, который будет в дальнейшем выполнять нужные для злоумышленника действия) теперь опережает SQL-инъекции. По состоянию на конец года не были устранены 67% уязвимостей веб-приложений.
- Значительно выросло число веб-атак с маскировкой. Часто запускаемые с помощью автоматического инструментария для «эксплуатации» веб-уязвимостей, многие атаки используют технологию маскировки их нападений на интернет-браузеры, пытаясь скрыть эти эксплоиты (программный код, «эксплуатирующий» уязвимости в программном обеспечении для проведе-

ния атак) в документах и на веб-страницах, чтобы избежать обнаружения системами информационной безопасности. По сравнению с тем же 2008 годом было выявлено в 4 раза больше атак с маскировкой.

- Большинство фишинговых атак организовывалось из Бразилии, США и России, которые вытеснили Испанию, Италию и Южную Корею с первых позиций.
- В целом отмечено, что 61% фишинговых электронных писем отправляются мошенниками якобы от финансовых институтов, тогда как 20% выглядят как официальные письма государственных организаций.

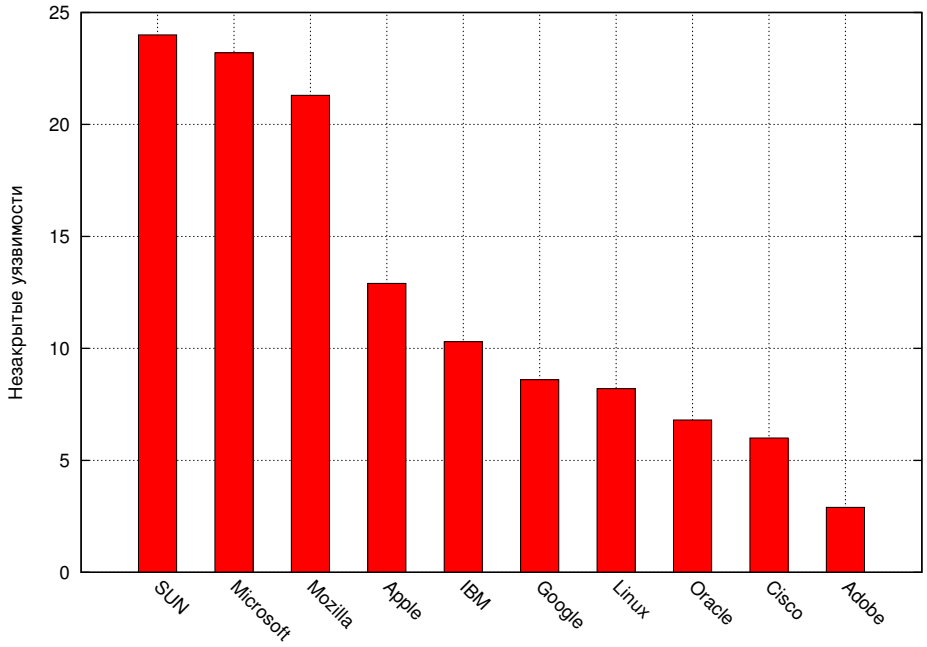
2.3 Рейтинг уязвимостей в приложениях ведущих разработчиков

Особый интерес представляет рейтинг крупнейших компаний-разработчиков, в программном обеспечении которых было обнаружено наибольшее количество уязвимостей. Один из таких рейтингов ежегодно составляет компания IBM (см. рис.2.7). В отчете Security X-Force [6] содержится анализ основных источников угроз и уровня опасности уязвимостей в программном обеспечении за первую половину 2010 года.

В результате проведенного исследования был зафиксирован рост числа уязвимостей на 36% по сравнению с тем же периодом 2009 года.

Показатель доли незакрытых уязвимостей в каждый промежуток времени на протяжении всего анализируемого периода сохраняется на уровне 55% от всего числа обнаруженных уязвимостей. Если учитывать характер обнаруживаемых уязвимостей и считать критическими, те уязвимости, которые могут быть использованы для удаленного распространения вредоносного кода, то эта цифра возрастает до 71%. Таким образом, можно сделать вывод, что чем более опасна уязвимость, тем больше времени компании разработчики тратят на ее устранение. Лидерами среди разработчиков программного обеспечения, которые оставляют неустраненными самую большую долю уязвимостей стали бывшая Sun Microsystems (ныне — часть Oracle Corporation)(24%) и Microsoft (23.2%) [11].

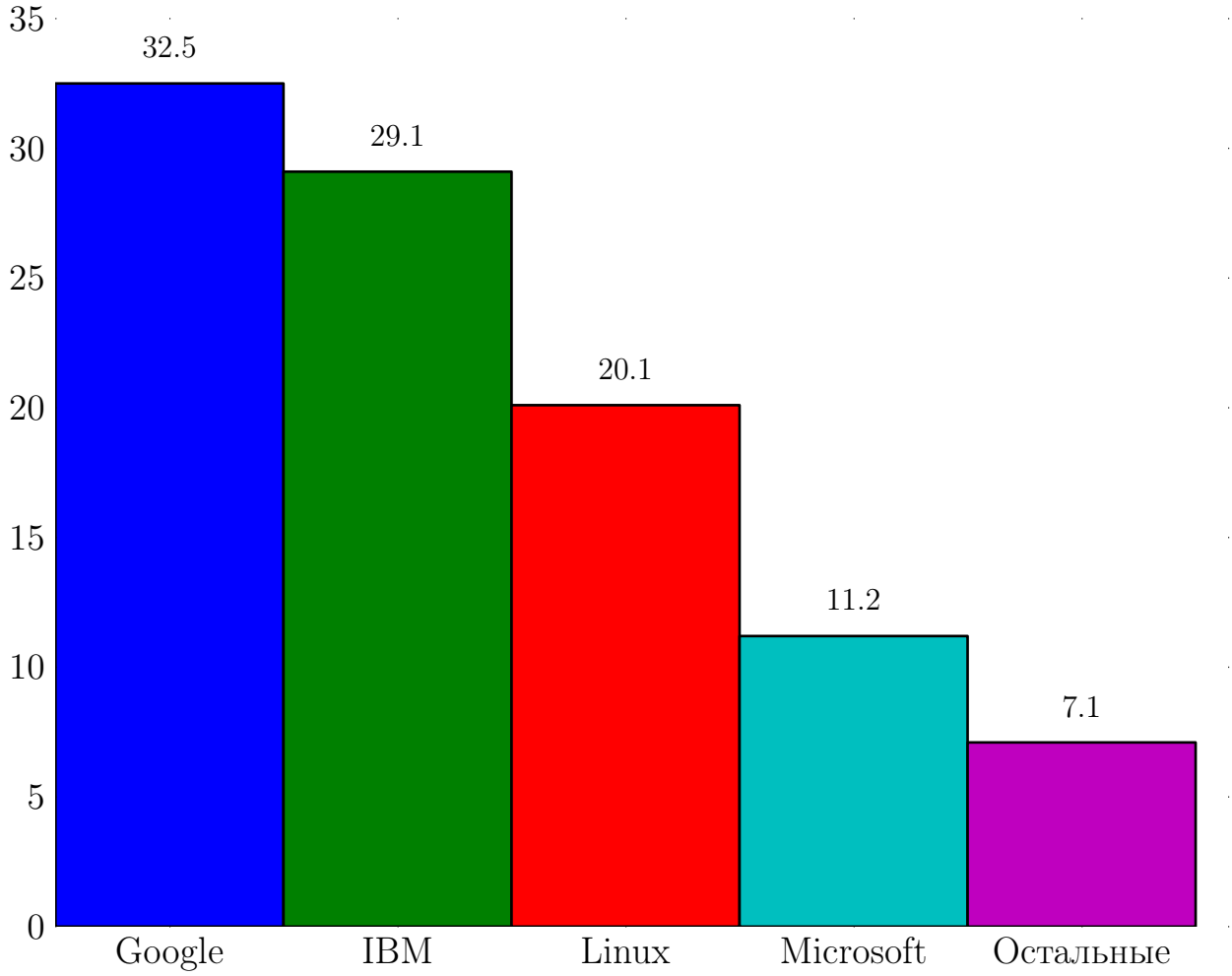
В прошлом году Microsoft [9] возглавила тот же список, оставив незакрытыми 15.8% уязвимостей, тогда как Sun - 2.6%. Третье место по незакрытым уязвимостям среди разработчиков занимает организация Mozilla Foundation(21.3%), за ней следует Apple (12.9%) и IBM (10.3%). Новичком списка стала компания Google, оставив неустраненными около 8.6% уязвимостей, тем самым опередив сообщество разработчиков платформы Linux (8.2%).



Источник: IBM X-Force

Рис. 2.7. Рейтинг компаний по числу открытых уязвимостей в программном обеспечении

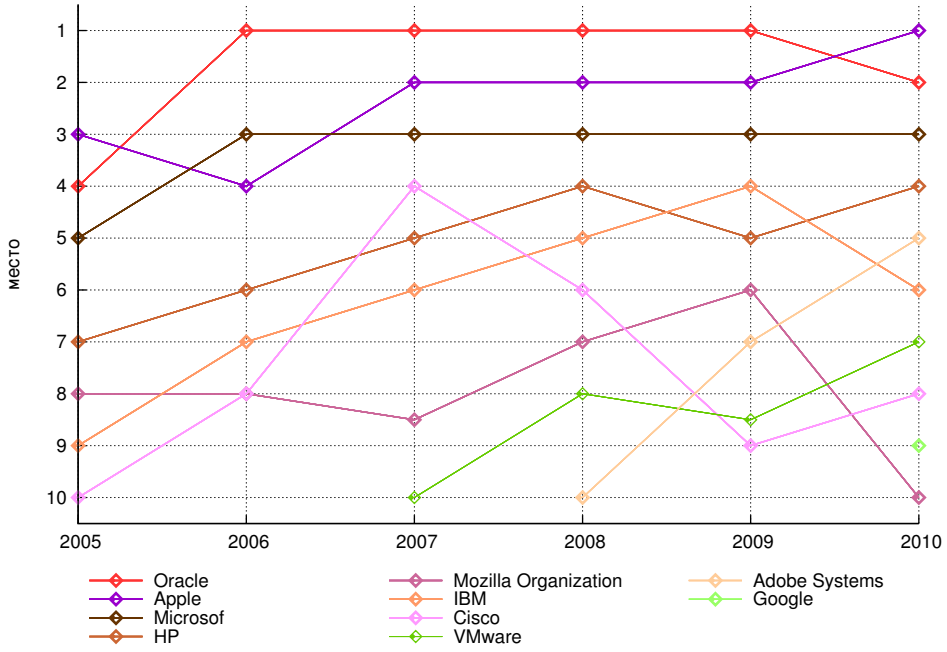
Критические уязвимости дольше всего устраняют следующие компании (сообщества разработчиков): Google (33%), IBM (29%), Linux (20%), Microsoft (11%). Наибольшее число уязвимых приложений было создано для платформы Linux (31%), чуть менее для Mac OS X(Apple) (29%). Однако по времени устранения, критических уязвимостей бесспорным лидером стали приложения для ОС компании Microsoft (73%), на втором месте — ОС Linux (16%) (см. рис.2.8).



Источник: IBM X-Force

Рис. 2.8. Средняя продолжительность устранения критических уязвимостей (в днях)

Можно сравнить эти данные с результатами, полученными компанией Secunia [8], специализирующейся на компьютерной и сетевой безопасности (см. рис.2.9).



Источник: Secunia

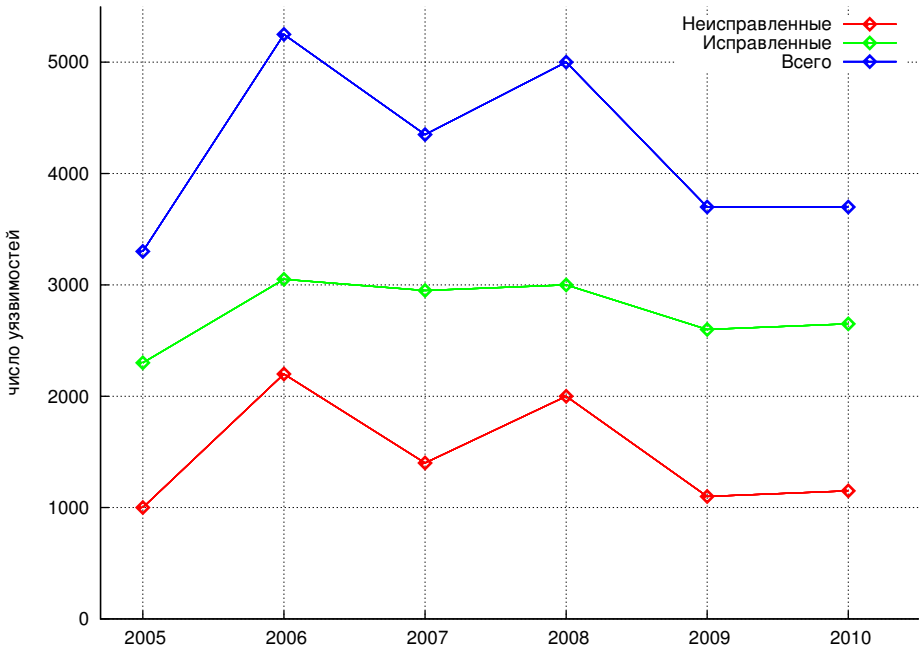
Рис. 2.9. Рейтинг 10 ведущих вендоров

Общая картина распределения уязвимостей представлена на графике (см. рис.2.10).

Источник: Secunia

Рис. 2.10. Динамика уязвимостей на основе публичных данных компании Secunia

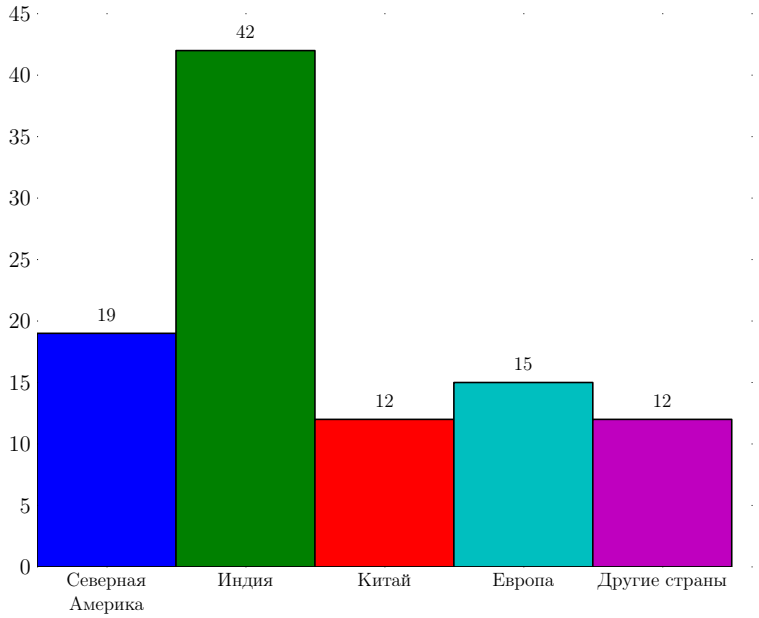
Доступен также график очень близкий по результатам, составленный специалистами компании «Эшелон» по данным баз уязвимостей CVE (см. рис.2.11).



Источник: ЗАО «НПО «Эшелон»

Рис. 2.11. Общая динамика уязвимостей по данным публичных баз данных

2.4 Распределение уязвимостей по регионам происхождения



Источник: Secunia

Рис. 2.12. Распределение уязвимостей по происхождению кода (%)

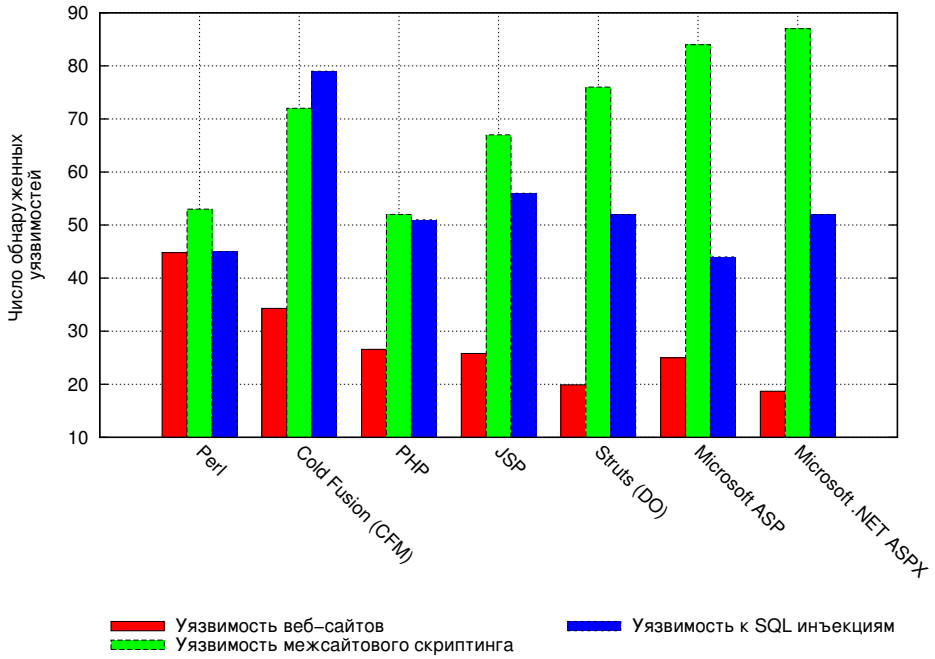
2.5 Вывод

Несмотря на широкое распространение бюллетеней безопасности и достаточно высокую популярность темы защищенности в пресс-релизах вендоров и интеграторов программного обеспечения, существенного снижения реального числа уязвимостей за последние годы не произошло, при этом остаётся достаточно большим время, требуемое разработчикам на исправлении обнаруженных уязвимостей в их продукте. Также не замечено больших различий в защищенности закрытого (коммерческого) и открытого программного обеспечения.

3 Безопасность и языки программирования

3.1 Рейтинг уязвимости языков веб-программирования

Весной 2010 года компания WhiteHat Security опубликовала отчет по исследованию взаимосвязи веб-уязвимостей и используемых при разработке этих ресурсов платформ и языков программирования [15]. Данные были получены на основе анализа собранной информации по уязвимостям примерно 1659 сайтов, находящихся в ведении этой компании в период с 2006 г. по 2010 г. Ниже приведена гистограмма рейтинга уязвимых языков программирования (см. рис.3.1).



Источник: WhiteHat Security

Рис. 3.1. Распределение языков программирования по типам уязвимостей (в %)

Исследование показало следующее:

- Распределение уязвимостей среди языков программирования неоднородно. Это связано с их спецификой и сферой применения.
- В продуктах, написанных на языке Perl, в настоящее время найдено наибольшее число уязвимостей 11,8% (в прошлом году было — 44,8%).

- Набор библиотек (фреймворк) Struts (DO) обошел Microsoft's .NET (ASPX), показав меньшее число открытых уязвимостей (5,5 против 6,2 на сайт в среднем).
- Платформа Cold Fusion (CFM) заняла второе место по среднему числу уязвимостей на веб-сайт с показателем 34,4, но при этом показала низкую вероятность присутствия серьезной неустраненной уязвимости, в случае, когда веб-платформа работала под управлением WhiteHat Sentinel — (54%). Недалеко от него в рейтинге ушел Microsoft ASP Classic, который с показателем в 57% выиграл один процент у своего преемника Microsoft .NET.
- Веб-сайты, построенные с помощью Perl (PL), Cold Fusion (CFM), Java Server Pages (JSP), PHP, скорее всего, имеют, по крайней мере, одну серьезную[1] уязвимость, вероятность этого составляет примерно 80%. Сайты, построенные на других языках/платформах, имели уязвимости в среднем лишь в 10% случаев.
- Если рассматривать сайты, содержащие URL-адреса с расширениями файлов, специфичными для Microsoft .NET(ASPX), то 36% уязвимостей были обнаружены на страницах с расширениями специфичными для Microsoft ASP Classic. С другой стороны, лишь у 11% уязвимостей на веб-сайтах ASP было расширение ASPX.
- 37% веб-сайтов на Cold Fusion (CFM) были уязвимы для SQL-инъекций, это самый высокий показатель, в то время как у Struts (DO) и JSP был самый низкий уровень подобной уязвимости - от 14% до 15%.
- В среднем уязвимости для SQL-инъекций устранялись быстрее всего на сайтах под управлением Microsoft ASP Classic, за 44 дня, далее следовал Perl (PL) со значением в 45 дней.
- Большинство (79%) критических уязвимостей SQL-инъекции были зафиксированы на сайтах под управлением платформой Struts(DO). Далее следуют сайты на платформе Microsoft .NET (ASPX) (71%), Perl (PL) - 71%, уровни остальных сайтов находятся в промежутке между 58% и 70%.

[1] Серьезной считается уязвимость, позволяющая обеспечить при её эксплуатации удаленное управление системой.

- Более 80% сайтов на Perl имеют уязвимости межсайтового скриптинга (это — самый высокий уровень), в то время как на чуть более половине ASPX-сайтов есть данный тип уязвимости (самый низкий уровень).
- Веб-сайты на языках PHP и Perl показали наихудшее значение по среднему числу уязвимостей, но у них был самый короткий период устранения уязвимостей к межсайтовому скриптингу - 52 и 53 дня соответственно. В то же время, сайты на базе Microsoft .NET (ASPX) были одними из лучших по числу уязвимостей, но при этом они оказались в числе худших по среднему времени устранения уязвимости - 87 дней.
- Скорость устранения "критических"уязвимостей межсайтового скриптинга во всех проверенных языках программирования и фреймворках оказалась в диапазоне 50% - 60%, исключение составил лишь PHP с 66%.
- Языки Perl и JSP удивили временем устранения уязвимостей неправомерной авторизации — 20 и 29 дней соответственно. Исторически сложилось так, что уязвимости неправомерной авторизации требуют более 50 дней на устранение.

Заключение

Как видно из рассмотренных исследований, такие системно-ориентированные языки как C и C++ ещё сохраняют высокую популярность, в том числе и в России, несмотря на то, что из-за прямого доступа к памяти и особенностей библиотек они подвержены ошибкам, связанным с переполнением буфера и форматной строкой. Также у нас в стране всё ещё весьма велик объем кода, написанного в среде Delphi — Object Pascal находится в тройке популярных по проведению аудита кода в России).

Что касается среды веб-приложений, то наиболее подверженным уязвимостям показали программы, написанные на языке Perl, одними же из наиболее защищенных веб-приложений являются программные продукты, написанные на Java (с использованием фреймворка Struts).

Опыт по проведению аудита и сертификации показывает, что наиболее важным, с точки зрения безопасности являются моменты, связанные с менеджментом процесса разработки и квалификация программистов, которые должны быть знакомы с достоинствами и недостатками той программной технологии, которую они используют.

Несмотря на развитие средств, методов и подходов к аудиту программного обеспечения, приходится констатировать, что ситуация в области защищенности приложений практически не изменилась в лучшую сторону.

Авторы

В подготовке отчета приняли участие следующие сотрудники компании «Эшелон»: Андрей Фадин, Михаил Никулин, Дарья Кистанова, Константин Лебедев, Алексей Маркин, Антон Огородников, Алексей Фамбулов.

Под редакцией Алексея Маркова и Валентина Цирлова.

О компании ЗАО «НПО «Эшелон»

Одними из ключевых направлений деятельности компании «Эшелон» является сертификация средств защиты информации, программных продуктов и автоматизированных систем различного назначения, а также проведение внешнего аудита разрабатываемого программного обеспечения с целью выявления программных закладок, уязвимостей, некорректностей программирования и недекларированных возможностей в целом. Компания имеет опыт тестирования более 300 программных изделий по требованиям безопасности, сотрудники компании проводили испытания и экспертизы материалов испытаний программных продуктов таких известных компаний, как Microsoft, Oracle, SAP, Symantec, ESET, Huawei, Astaro, Motorola, Tadiran Technologies, Saperion, Trend Micro, Amdocs, ГК Информзащита, Доктор Веб, Лаборатория Касперского, Анкад, Positive Technologies, РусБИТех, ВНИИНС и других.

В компании действует учебный центр, проводящий обучение по учебным программам согласованным со ФСТЭК России и Минобороны России, в том числе компания организует авторизованный курс по сертификации программного обеспечения.

Компания «Эшелон» является разработчиком средств автоматизации аудита безопасности программного обеспечения (анализатор исходных текстов АК–ВС), анализа защищенности сети (Сканер–ВС), инспекционного контроля (ПИК–Эшелон) и других.

Список использованных источников

1. 2010 CWE/SANS Top 25 Most Dangerous Software Errors: Tech. Rep. 1.07 / Bob Martin, Mason Brown, Alan Paller, Dennis Kirby.
2. Chang Bor-Yuh, Harren Matthew, Necula George. Analysis of Low-Level Code Using Cooperating Decompilers. — 2006. — Pp. 318–335.
3. Hicks Boniface, Ahmadizadeh Kiyan, McDaniel Patrick. From Languages to Systems: Understanding Practical Application Development in Security-typed Languages // Computer Security Applications Conference, Annual. — 2006. — Vol. 0. — Pp. 153–164.
4. Hovemeyer David, Spacco Jaime, Pugh William. Evaluating and tuning a static analysis to find null pointer bugs // PASTE '05: The 6th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering. — New York, NY, USA: ACM Press, 2005. — Pp. 13–19.
5. Information Technology — Programming Languages — Guidance to Avoiding Vulnerabilities in Programming Languages through Language Selection and Use: Tech. Rep. ISO/IEC PDTR 24772: May 2009.
6. Lamb K. L. The IBM X-Force 2010 Mid-Year Trend and Risk Report. — 2010.
7. Nystrom Erik M., Kim Hong-Seok, Hwu Wen-Mei W. Bottom-Up and Top-Down Context-Sensitive Summary-Based Pointer Analysis. — 2004. — Pp. 165–180.
8. Rasmussen N. H. Tech. Rep.: / Ed. by N. H. Rasmussen; Secunia: 2010.
9. Reavey Mike, Stone Adrian, Jones Jeff. Software Vulnerability Management At Microsoft. — July 2010.
10. Sankaranarayanan Sriram, Ivančić Franjo, Gupta Aarti. Program Analysis Using Symbolic Ranges. — 2007. — Pp. 366–383.
11. The Microsoft Vulnerability Research Program. — July 2010.
12. Veracode. Automating Your Code Review: Moving to a SaaS Model for Application Security. — 2008.
13. Veracode. The Intractable Problem of Insecure Software: Volume 1. — March 2010.
14. Veracode. The Intractable Problem of Insecure Software: Volume 2. — September 2010.

15. WhiteHat Security Inc. WhiteHat Website Security Statistic Report.
16. Материалы. сертификационных испытаний: НПО «Эшелон», 2007-2010.

ЗАО «НПО «Эшелон» работает на рынке информационной безопасности с 2006 года. За это время компанию выбрали в качестве стратегического партнера ведущие мировые производители ERP-систем, операционных систем, антивирусного программного обеспечения, средств защиты от несанкционированного доступа.
Компания «Эшелон» аккредитована в качестве испытательной лаборатории Минобороны России, ФСТЭК России, ФСБ России, системы сертификации «АйТи-Сертифика». Наша компания является аттестационным центром Минобороны России и органом по сертификации ФСТЭК России, органом по аттестации ФСТЭК России. Система менеджмента качества компании сертифицирована на соответствие требованиям ISO 9001, ГОСТ РВ 15.002 и СРПП ВТ. Учебный центр «Эшелон» проводит переподготовку специалистов по программам, согласованным с ФСТЭК России и Минобороны России.

Наши продукты для комплексной защиты

СКАНЕР-ВС

Программный комплекс «Сканер-ВС» предназначен для комплексного анализа защищенности информационных систем.

Возможности

- сканирование на наличие уязвимостей
- локальный и удаленный анализ стойкости паролей (поддерживаются Linux, Windows, а также более 20 сетевых протоколов)
- перехват и анализ сетевого трафика
- инвентаризация сетевых сервисов
- инвентаризация локальных ресурсов
- поиск остаточной информации на локальном диске компьютера

Сертификат соответствия

ФСТЭК России и Минобороны России

РУБИКОН

Межсетевой экран и система обнаружения вторжений «Рубикон» предназначена для обеспечения полной защиты периметра сетей, в которых обрабатывается информация с грифом до «совершенно секретно» включительно.

Сертификат соответствия

Минобороны России

ГЕНЕРАТОР

Программное средство «Генератор» предназначено для генерации устойчивых паролей и управления ими.

Возможности

- генерация паролей установленной длины с помощью заданного алгоритма (в том числе алгоритма, совместимого с ГОСТ 28.147-89)
- удаленное назначение паролей пользователям
- поддержка ОС Windows NT4/2K/XP/Vista/7
- импорт списков пользователей с локального или удаленного компьютера, из домена Windows, а также импорт из файлов в формате XML
- соответствие криптографическим и инженерно-криптографическим требованиям к программным датчикам случайных чисел, используемым в средствах защиты информации объектов вычислительной техники Вооруженных Сил Российской Федерации

Сертификат соответствия

Минобороны России