- оценить трудоемкость процесса развертывания и начальной настройки СОВ.

В процессе натурных испытаний на основе сформированной тестовой выборки атак производится моделирование действий злоумышленников, направленных на получение несанкционированного доступа и регистрируются результаты (успешное, неуспешное) и длительность действий СОВ по выявлению, распознаванию и реагированию на атаки. По завершении тестирования определяется соответствие фактических (определенных по результатам тестирования) функциональных возможностей СОВ предъявленным требованиям.

По зафиксированным результатам и длительностям действий СОВ и в соответствии с описанной методикой производится вычисление значений показателей точности и производительности СОВ, а также показателей полноты и оперативности обнаружения атак. Поскольку разные атаки имеют различный уровень опасности для защищаемой информации, то применение весовых коэффициентов, вычисляемых на основе какого-либо метода ранжирования угроз, при расчете показателей точности, полноты и оперативности обнаружения позволяет количественно учитывать индивидуальные особенности СОВ по выявлению и блокированию атак.

Таким образом, разработанная методика оценивания функциональных возможностей СОВ позволяет, вне зависимости от архитектуры СОВ и реализованных в ней методов обнаружения атак, решить задачи сравнительного анализа обладающих схожими функциональными возможностями СОВ различных производителей и обоснования выбора для эксплуатации в конкретной АС СОВ, соответствующей требованиям политики безопасности информации АС, обладающей удовлетворительными значениями показателей точности, производительности, полноты и оперативности обнаружения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лукацкий А. В.* Обнаружение атак. – СПб.: БХВ, 2003 – 596 с.

2. Young G. Seven key selection criteria for network IPS. Gartner report G00123902, 2004.

3. *Лукацкий А.В., Цаплев Ю.Ю.* Системы обнаружения атак. Стратегия выбора. Internet Security Systems, Inc., 1999.

А.С. Марков, С.В. Миронов, В.Л. Цирлов,
Россия, г. Москва, ЗАО «НПП “БИТ”»

ВЫЯВЛЕНИЕ УЯЗВИМОСТЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРОЦЕССЕ СЕРТИФИКАЦИИ

Вопросами выявления уязвимостей и сертификационных испытаний программного обеспечения занимаются испытательные лаборатории Минобороны, ФСБ и ФСТЭК России, которые в своей работе опираются на Руководящий документ (РД) Гостехкомиссии России [1]. Однако опыт выявления авторами более 20 закладок и тысяч некорректностей кода показывает, что большинство из них обнаружены не столько в соответствии с указанным документом, сколько вопреки ему. Существующая нормативная, методическая и инструментальная база [2] выявления недекларированных возможностей в программном обеспечении не позволяет эффективно обеспечивать безопасность программ.

Объективные причины появления уязвимостей в программных продуктах заключаются в чрезвычайно высокой структурной сложности программного кода, динамичности развития версий и легкости модификации кода [3]. К этому можно добавить проблему достоверной идентификации преднамеренно созданных программных закладок, несовершенство нормативно-методической базы и отставание инструментальной базы сертификационных испытаний. Скажем, в отличие от

средств антивирусного контроля, отсутствуют средства гарантированного выявления программных закладок в структурно сложном программном обеспечении. Полувековые теоретические изыскания [4] так и не обеспечили разработки математического аппарата для оценки степени безопасности программного обеспечения, основанного на сертификационных испытаниях с целью подтвердить отсутствие программных закладок. Поэтому снижается качество разработки отечественных программных средств в государственных учреждениях, как и уровень доверия к зарубежным продуктам. Руководящий документ по недекларированным возможностям [1] был создан в 90-е годы для решения задачи контроля над поставляемыми в Россию зарубежными продуктами и тогда, вне сомнений, имел большое значение. Методы, определяемые в РД, берут начало в теории надежности функционирования программ, поэтому вопросы защиты собственно кода отражены в документе недостаточно явно. В соответствии с РД основными видами проверок, которые должны проводиться испытательными лабораториями, являются структурный статический и динамический анализ исходных текстов (структуры программы, формирования и прохождения всех ее путей).

К достоинствам документа следует отнести требования предоставлять исходный код и документацию, осуществлять контроль над избыточностью (что позволяет исключить некоторые закладные элементы) и наличие определения полномаршрутного тестирования. Последнее при соответствующем мониторинге и аудите работы позволяет выявить большинство уязвимостей несложных программ, но применительно к сложным программам это весьма проблематично. Назовем наиболее серьезные недостатки РД. Значительная вычислительная сложность статического и динамического анализа. Фактически, с ростом сложности программного продукта динамический анализ становится неразрешимой задачей и превращается в формальную процедуру, отнимающую много времени и ресурсов у экспертов [3].

Отсутствие или недостаточность проверок, непосредственно связанных с безопасностью изделия. Например, не упоминается сигнатурный анализ в требованиях к испытаниям программ ниже второго уровня контроля (т.е. программ, обрабатывающих секретную и конфиденциальную информацию). Не выдвигаются требования к содержимому базы потенциально опасных конструкций, что делает неэффективными методы сигнатурного анализа. Нет механизмов выявления ошибок кодирования, связанных с переполнением буфера, гонками, вызовом функций из чужого адресного пространства. Отсутствует очистка памяти и т.д. Неопределенность действий экспертов и достоверности получаемых результатов. Отсутствуют декларированные способы, нацеленные на построение перечня маршрутов при выполнении функциональных объектов (ветвей). Нет механизмов определения полноты покрытия и его достаточности при динамическом анализе. Отсутствуют рекомендации по выявлению недекларированных возможностей при анализе, например, матрицы связей по информации.

Самыми небезопасными остаются продукты, прошедшие сертификацию по третьему или четвертому уровню контроля [1]. Четвертый уровень сводится к проверке возможности компиляции и сборки программного продукта, а также отсутствия избыточных файлов в его дистрибутиве. Таким образом, создается иллюзия проверки программы в соответствии с требованиями безопасности. Для третьего уровня подход выявления недекларированных возможностей представляет собой чрезвычайно длительное изучение структуры программы, не включающее в себя анализ функций, процедур и методов на признаки программных закладок и некорректностей кодирования кода.

Чтобы получить аттестат аккредитации ФСТЭК по выявлению недекларированных возможностей, испытательная лаборатория должна иметь сертифицированные инструментальные средства («АИСТ», EMU). Здесь возникают два вопроса.

Во-первых, целесообразно ли делать обязательными к применению инструментальные средства, основанные на несовершенной нормативной базе и не позволяющие эффективно выявлять уязвимости кода? Например, проведенные нами исследования (в эксперименте участвовали несколько десятков экспертов) тестовой программы, содержавшей 28 простейших закладок, показали, что ни одна из них не была выявлена с помощью указанных средств.

Во-вторых, следует ли в обязательном порядке сертифицировать такие средства? Из-за длительности процесса сертификации средства тестирования уязвимостей никогда не смогут гарантировать результат согласно принципу «снаряд–броня». Совершенно очевидно, что эти средства не позволяют оперативно включать признаки современных уязвимостей кода в меняющейся среде защиты и в новых средах программирования.

Скажем, для анализа кода на языке Си рекомендуется программа «АИСТ-С» (непререкаемый «авторитет» в области анализа кода на наличие недекларированных возможностей). Однако, несмотря на ее очевидные достоинства, установлено, что она некорректно обрабатывает структуры объектно-ориентированных программ, например не анализирует конструктор и деструктор. Не анализируются и исходные тексты, подключаемые директивами #include — лишь проверяется исходный текст программы. Мало того, нормальная работа анализатора возможна, только если объем исходных текстов не превышает нескольких мегабайт: блок-схемы программ строятся некорректно уже для программ объемом более нескольких килобайт. Те же ограничения относятся к построению матрицы связей по информации, да и отчеты не в полной мере соответствуют требованиям РД по недекларированным возможностям. Но главное отставание от реалий дня состоит в том, что это средство вообще не имеет базы сигнатур и умеет анализировать лишь программы на языке Си/С++. Принципиальный недостаток принятой процедуры сертификации — отсутствие учета политики безопасности использования программного продукта в рамках объекта информатизации. Кроме того, возникает ряд вопросов.

- Если в изолированной системе нет программных каналов утечки информации, то так ли необходимо наличие сертификата на отсутствие недекларированных возможностей и не достаточно ли проверок на доступность (готовность, качество) изделия? Например, почему проверкам на отсутствие недекларированных возможностей не подлежит ряд сетевых экранов?

- В нормативной документации нет четкого ответа, как организовать сертификацию программных продуктов с постоянно изменяющимся кодом. Если в продукте обнаружены уязвимости, то должно ли быть приостановлено действие сертификата?

- Как определить уровень глубины выявления недекларированных возможностей: на уровне приложения, базового программного обеспечения, операционной среды, программно-аппаратном? Если имеются исходные коды только для программной оболочки (например, сетевого экрана, функционирующего в операционной среде), то как поступать со средой?

- Что делать, если нет исходных кодов? А ведь некоторые проверки можно (и даже желательно) проводить и без оного. Например, анализ механизмов парольной защиты целесообразно осуществлять в отладочном режиме на уровне исполняемых модулей: тогда наиболее эффективно выявляются низкоуровневые деструктивные функции.

- Что делать, если сертификацию нельзя провести по правовым или техническим причинам? Например, неизвестен правообладатель продукта или фрагментов кода, разработчик находится за рубежом и распространяет продукт бесплатно, нет

спецификации на продукт, утеряны исходные тексты. Не пора ли ввести систему спецпроверки программ по аналогии с аппаратными средствами [3]?

Отдельного рассмотрения заслуживает сложившаяся в настоящее время в России достаточно противоречивая ситуация с нормативными документами, регламентирующими порядок сертификации автоматизированных систем и средств вычислительной техники по требованиям защиты информации. Помимо традиционных Руководящих документов Гостехкомиссии России (ФСТЭК) принят и введён в действие ГОСТ Р ИСО/МЭК 15408-2002 [8], представляющий собой аутентичный перевод международного стандарта, известного как «Общие критерии».

При этом практическому применению данного стандарта для проведения сертификационных испытаний должны предшествовать разработка и утверждения пакета профилей защиты для автоматизированных систем и средств вычислительной техники различного уровня. По своей сути такие профили должны послужить заменой РД Гостехкомиссии, регламентирующих порядок сертификации автоматизированных систем и средств вычислительной техники на отсутствие НСД.

При этом остаётся открытым вопрос о взаимосвязи между «Общими критериями» и руководящим документом [1]. Считается, что проведение сертификационных испытаний на отсутствие недекларированных возможностей не может быть регламентировано в рамках «Общих критериев», а значит, сертификационные испытания в рамках существующей парадигмы не могут быть основаны исключительно на Задании по безопасности соответствующего объекта оценки.

В то же время, вопреки сложившемуся мнению, анализ «Общих критериев» показывает, что в рамках классов третьей части стандарта могут быть сформулированы практически все требования РД НДВ. Примерное соответствие между пунктами РД НДВ и классами требований доверия «Общих критериев» показано в таблице.

№ п/п	Требования РД	Классы «Требований доверия» общих критериев
	Требования к документации	
1.	Контроль состава и содержания документации	APE – оценка профиля защиты. ASE – оценка задания по безопасности.
1.1.	Спецификация (ГОСТ 19.202-78)	
1.2.	Описание программы (ГОСТ 19.402-78)	ADV_FSP – функциональная спецификация
1.3.	Описание применения (ГОСТ 19.502-78)	ADO – поставка и эксплуатация AGD_ADM – руководство администратора AGD_USR – руководство пользователя
1.4.	Пояснительная записка (ГОСТ 19.404-79)	ADV_HLD – проект верхнего уровня ADV_LLD – проект нижнего уровня ADV_RCR – соответствие представлений
1.5.	Тексты программ, входящих в состав ПО (ГОСТ 19.401-78)	ADV_IMP – представление реализации
	Требования к содержанию испытаний	
2.	Контроль исходного состояния ПО	ACM_AUT – управление конфигурацией ACM_CAP – возможности управления конфигурацией ACM_SCP – область управления конфигурацией ALC_TAT – инструментальные средства и методы

№ п/п	Требования РД	Классы «Требований доверия» общих критериев
3.	Статический анализ исходных текстов программ	
3.1.	Контроль полноты и отсутствия избыточности исходных текстов	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
3.2.	Контроль соответствия исходных текстов ПО его объектному (загрузочному) коду	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
3.3.	Контроль связей функциональных объектов по управлению	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
3.4.	Контроль связей функциональных объектов по информации	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
3.5.	Контроль информационных объектов	
3.6.	Контроль наличия заданных конструкций в исходных текстах	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
3.7.	Формирование перечня маршрутов выполнения функциональных объектов (ветвей)	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
3.8.	Анализ критических маршрутов выполнения функциональных объектов	AVA_CCA – анализ скрытых каналов AVA_SOF – стойкость функций безопасности ОО AVA_VLA – анализ уязвимостей
3.9.	Анализ алгоритма работы функциональных объектов на основе блок-схем, диаграмм и т.п., построенных по исходным текстам контролируемого ПО	ADV_RCR – соответствие представлений AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
4.	Динамический анализ исходных текстов программ	ATE_IND – независимое тестирование AVA_MSU – неправильное применение
4.1.	Контроль выполнения функциональных объектов (ветвей)	ATE_COV – покрытие ATE_DPT – глубина AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
4.2.	Сопоставление фактических маршрутов выполнения функциональных объектов и маршрутов, построенных в процессе проведения статического анализа	AVA_CCA – анализ скрытых каналов AVA_VLA – анализ уязвимостей
5.	Отчетность	

Прежде всего, заметим, что классы требований доверия полностью покрывают все вопросы, связанные с анализом документации. Более того, классы APE и ASE учитывают необходимость анализа документов, специфических для «Общих критериев», а именно Профиля защиты и Задания по безопасности. В явном виде отсутствуют требования к анализу спецификации (в том виде, в каком она регламентируется ЕСПД), однако соответствующие проверки можно провести, например, в рамках ADV_HLD.

Конкретный механизм контроля исходного состояния ПО (например, необходимость применения того или иного алгоритма контрольного суммирования) может быть регламентирован в качестве уточнения для одного из компонентов ACM_AUT

или ALC_TAT. Безусловно, в явном виде требования и конкретные методики проведения статического анализа исходных текстов программ в «Общих критериях» отсутствуют. Однако все они могут быть сформулированы в рамках анализа уязвимостей (AVA_VLA) и анализа скрытых каналов (AVA_CCA). В данном случае общий характер соответствующих компонентов классов доверия ни в коей мере не является недостатком, а наоборот предоставляет эксперту возможность реализации более гибкого подхода к проведению проверок, позволяя учитывать особенности конкретной системы. Динамический анализ, помимо уже упомянутых семейств AVA_VLA и AVA_CCA, обеспечивается компонентами ATE_COV (покрытие) и ATE_DPT (глубина). Косвенно здесь можно упомянуть семейство ATE_IND, более, впрочем, ориентированное на функциональное тестирование. Тем самым, потенциал «Общих критериев» вполне может быть использован для организации сертификационных испытаний программных средств на отсутствие недекларированных возможностей и программных закладок в системах сертификации ФСТЭК и МО РФ. Соответствующие методики, разработанные, например, в рамках НИР в интересах ФСТЭК России или МО РФ и интегрированные в единый процесс сертификации по ГОСТ Р ИСО/МЭК 15408-2002, могли бы обеспечить методическую полноту и самодостаточность таких испытаний.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Руководящий документ. Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей. М.: Гостехкомиссия России, 1998.

2. *Калайда И.А.* Недекларированные возможности в программном обеспечении средств защиты информации. Jet Info, 2000, № 8.

3. *Марков А.С., Щербина С.А.* Испытания и контроль программных ресурсов. InformationSecurity, 2003, № 6.

4. *Дастин Э., Рэшке Д., Пол Д.* Автоматизированное тестирование программного обеспечения: внедрение, управление, эксплуатация. М.: Лори, 2003.

5. ГОСТ Р ИСО/МЭК 15408-2002. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Части 1, 2, 3. М.: ИПК «Изд-во стандартов», 2002.

6. Методическое руководство по оценке качества функционирования информационных систем. М.: Изд-во 3 ЦНИИ МО РФ, 2003.

7. ГОСТ РВ 51719-2001. Испытания программной продукции. М.: ИПК «Изд-во стандартов», 2001.

8. Руководящий документ. Безопасность информационных технологий. Критерии оценки безопасности информационных технологий. Части 1, 2, 3. — М.: Гостехкомиссия России, 2002.

9. Скрытые каналы, // Тематический сборник. Jet Info, 2002, № 11.

10. *Хогланд Г., Мак-Гроу Г.* Взлом программного обеспечения: анализ и использование кода. М.: Вильямс, 2005.

Е.В. Горковенко
Казахстан, г .Алматы, Институт проблем информатики и управления

ПАРАМЕТРЫ БЕЗОПАСНОСТИ В МНОГОУРОВНЕВОЙ МОДЕЛИ РАЗГРАНИЧЕНИЯ ДОСТУПА

Основное назначение полномочной политики безопасности, базирующейся на многоуровневой мандатной модели разграничения доступа, заключается в регулировании доступа субъектов системы к объектам с различным уровнем критичности и предотвращении утечки информации с верхних уровней должностной иерархии на нижние, а также блокировании возможного проникновения с нижних уровней на верхние. Полномочное управление доступом подразумевает, что: