На правах рукописи

ГУРОВ ДМИТРИЙ ВАЛЕРЬЕВИЧ

МЕТОДЫ И СРЕДСТВА ПРОТИВОДЕЙСТВИЯ АТАКАМ НА КОМПЬЮТЕРНЫЕ СИСТЕМЫ, ОСНОВАННЫМ НА ИСПОЛЬЗОВАНИИ УЯЗВИМОСТЕЙ ПРОГРАММНОГО КОДА

Специальности

05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей
05.13.19 – методы и системы защиты информации, информационная безопасность

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Москва - 2012

Работа выполнена в Национальном исследовательском ядерном университете «МИФИ» (НИЯУ МИФИ)

Научный руководитель: доктор технических наук, профессор
Иванов Михаил Александрович

Официальные оппоненты:

д.т.н., профессор Минаев Владимир Александрович, проректор по инновационно-образовательной деятельности НОУ ВПО Российский новый университет,

к.т.н., доцент Федоров Николай Владимирович, доцент кафедры "Информационная безопасность" Московского Государственного Индустриального Университета,

Ведущая организация: ОАО "Концерн "Системпром".

Защита состоится «30» мая 2012 г. В 16 час. 30 мин. на заседании диссертационного совета Д 212.130.03 при Национальном исследовательском ядерном университете «МИФИ» по адресу: г. Москва, Каширское ш., д.31.

Телефоны для справок: +7 (499) 324-95-26, +7 (499) 324-73-34.

С диссертацией можно ознакомиться в библиотеке Национального исследовательского ядерного университета «МИФИ».

Отзывы в двух экземплярах, заверенные печатью организации, просьба направлять по адресу: 115409 г. Москва, Каширское ш., 31, диссертационные советы НИЯУ МИФИ, тел. +7 (499) 324-95-26.

Автореферат разослан « » _____ 2012 г.

Ученый секретарь диссертационного совета Леонова Н.М

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы

В настоящее время информатизация проникла практически во все сферы человеческой деятельности. Однако, уделяя большое внимание новым функциональным возможностям средств хранения и обработки информации, разработчики часто упускают из виду вопросы защищённости своих продуктов. Уязвимость программного кода, умышленно или чаще случайно оставленная разработчиком, может стать причиной глобальных электронных «эпидемий», наносящих ощутимый финансовый ущерб, а при определённых обстоятельствах способна даже привести к разрушению материальных объектов и гибели людей. Понятие уязвимости (vulnerability) можно определить как свойство компьютерной системы, наличие которого может позволить злоумышленнику нанести ущерб интересам владельца системы, иначе говоря, привести к реализации угрозы информационной безопасности (ИБ). Уязвимость программного кода – это особенность написания исходного кода программы, в результате наличия которой злоумышленник, используя специальным образом подготовленные входные данные или другие параметры программного окружения, может заставить уязвимую программу работать по алгоритму, отличному от заложенного программистом. Имеющиеся в настоящее время средства противодействия вредоносным программам не всегда могут справиться с решением своей задачи, главным образом потому, что нацелены на обнаружение и устранение последствий функционирования уже известных вредоносных программ и блокирование предварительно обнаруженных уязвимостей. Для повышения защищённости приложений требуется одновременно идти несколькими путями: искать универсальные признаки групп уязвимостей с целью их эффективного обнаружения, разрабатывать превентивные меры противодействия вредоносным программам, а также развивать системный подход к обучению новых разработчиков, способных уже на этапе написания программы обнаруживать и устранять содержащиеся в ней уязвимости.

Работы, посвящённые исследованию уязвимостей, содержат, в основном, перечисление существующих проблем, но не дают рекомендаций по повышению защищённости программного обеспечения (ПО). Разделение уязвимостей по классам могло бы дать программистам мощный инструмент выявления уязвимостей и предотвращения их использования. Однако в настоящее время комплексных работ по всем видам уязвимостей проведено не было.

Отчасти задача повышения защищённости приложений решается за счёт использования сканеров уязвимостей и безопасных компиляторов, которые в автоматическом режиме помогают обнаружить многие из из-

вестных уязвимостей и уменьшить риск их возникновения за счёт замены опасных функций на их безопасные аналоги. Однако использование таких инструментов возможно только при наличии исходного кода защищаемого приложения, а кроме того при этом увеличивается объём приложения и снижается его производительность. Эффективных инструментов защиты приложений, доступных только в виде исполнимых файлов в настоящее время не предложено.

Безопасность приложения должна закладываться на этапе написания исходного кода. Для этого необходима подготовка новых разработчиков, обладающих не только навыками программирования, но и хорошо знакомых с вопросами разработки защищённого ПО.

О необходимости подготовки таких специалистов говорится в Федеральной целевой программе развития образования на 2011 - 2015 гг., где ставится задача подготовки кадров по приоритетным направлениям модернизации и технологического развития экономики России, в числе которых выделяются стратегические компьютерные технологии и программное обеспечение. Также в Доктрине информационной безопасности среди основных задач выделяется создание единой системы подготовки кадров в области информационной безопасности и информационных технологий.

Таким образом, актуальность работы определяется возрастающими угрозами информационной безопасности, связанными с уязвимостями ПО, а также отсутствием эффективных методов противодействия уязвимостям программных систем, для которых не доступен исходный код.

При этом исследование механизмов появления наиболее опасных уязвимостей, принципов их использования при проведении атак на компьютерные системы и разработка программно-аппаратных методов противодействия им, а также разработка методики подготовки специалистов в области защиты от разрушающих программных воздействий (РПВ) является актуальной научной задачей.

Значительный вклад в исследование вопросов безопасного программирования, методов выявления уязвимостей внесли такие ученые как Н. Данжани (Nitesh Dhanjani), Дж. Кларк (Justin Clarke), Дж. С. Фостер (James foster), М. Мейсер (Matt Messier), Дж. Вега (John Viega), М. Ховард (Michael Howard), Д. Лебланк (David LeBlanc), Дж. Эриксон (Jon Erickson), Мэтт Бишоп (Matt Bishop), а также специалисты фирм Лаборатория Касперского, Symantec и других.

Цель исследования. Целью представленной работы является повышение защищённости компьютерных систем за счёт противодействия атакам, основанным на использовании уязвимостей, вносимых на этапе программирования.

Для достижения данной цели были решены следующие задачи:

– проведён анализ и разработана классификация существующих уязвимостей ПО;

– исследованы наиболее распространённые уязвимости ПО типа Buffer overflow (Stack smashing и Heap overflow), Race condition, Integer overflow (Widthness overflow, Arithmetic overflow, Signedness errors), Format string error, создающие предпосылки для проведения атак, в том числе основанных на внедрении вредоносного кода;

– исследованы существующие и разработаны оригинальные методы выявления и устранения уязвимостей программного кода, противодействия атакам, использующих известные уязвимости;

– разработана методика безопасного программирования, иначе говоря, создания программ, свободных от типичных уязвимостей, создающих предпосылки для проведения атак на компьютерные системы;

– разработаны методы противодействия уязвимостям прикладного ПО в случае, если исходный код приложения недоступен.

Объектом исследования является системное и прикладное ПО компьютерных систем.

Предмет исследования – уязвимости программного кода, создающие предпосылки для проведения атак на компьютерные системы, методы противодействия атакам на компьютерные системы.

Методы исследования. При решении поставленных задач использовались методы теории вероятностей и математической статистики, методы имитационного моделирования, методы модульного и объектно-ориентированного программирования, теория надёжности ПО, методы принятия решений.

Работа соответствует пунктам 1, 6, 14, 15 специальности 05.13.11 и пунктам 3, 8 специальности 05.13.19.

Научная новизна результатов работы заключается в следующем.

1) Разработана классификация уязвимостей ПО, создающая предпосылки для создания наиболее эффективных методов противодействия выявленным типам уязвимостей.

2) Разработаны модель нарушителя и модель атаки на компьютерную систему, основанной на использовании уязвимостей программного кода;

3) Предложен метод оценки повышения защищённости программы при использовании методики безопасного программирования, основанный на применения модели надежности ПО Миллса. Показано, что преимуществом применения этой модели по сравнению с классическим ее использованием является возможность учета выявленных на этапе классификации чётких признаков вносимых уязвимостей.

4) Предложен аппаратный метод защиты от эксплойтов, основанных на вставке вредоносного кода в программу; разработана модель рандомизационного генератора псевдослучайных чисел (ГПСЧ);

5) Впервые предложены методы повышения защищённости вычислительных систем, работающих в режиме виртуальной машины, эмулирующей аппаратные ресурсы вычислительной системы.

Практическая значимость результатов работы заключается в следующем.

1) Разработана методика безопасного программирования, позволяющая создавать ПО, свободное от появления типичных уязвимостей.

2) Разработана методика обучения безопасному программированию, поддержанная учебно-методическим пособием, электронными средствами обучения и системой тестирования знаний.

3) Впервые проведен анализ защищенности отечественного высокопроизводительного ВК «Эльбрус-3М1».

На защиту выносятся:

- классификация уязвимостей ПО;
- методика безопасного программирования, позволяющая создавать программы, свободные от известных уязвимостей программного кода;
- аппаратный метод повышения защищённости вычислительной системы от атак, основанных на внедрении вредоносного кода и разрушении стека;
- метод оценки количества уязвимостей, остающихся в программе после этапа её тестирования на предмет наличия уязвимостей;
- результаты исследования защищенности отечественного высокопроизводительного ВК «Эльбрус-3М1»;
- методы повышения защищённости вычислительных систем, работающих в режиме двоичного транслятора.

Достоверность основных положений диссертации обеспечивается корректностью применения математического аппарата, доказанностью выводов, совпадением теоретических результатов с экспериментальными, успешной практической реализацией результатов в образовательной деятельности, апробацией на научно-технических конференциях и семинарах, а также внедрением результатов в практическую деятельность ряда организаций.

Апробация работы. Основные результаты работы докладывались и обсуждались на Научных сессиях МИФИ 2003, 2004, 2008 гг. и НИЯУ МИФИ 2010 г., выставках-конференциях «Телекоммуникации и новые информационные технологии в образовании» (МИФИ, 2004 и 2006 гг.), международных телекоммуникационных конференциях студентов и молодых ученых "МОЛОДЕЖЬ И НАУКА" (МИФИ, 2005, 2010, 2012 гг.), Международном симпозиуме «Образование через науку», посвященную 175-

летию МГТУ им. Н.Э. Баумана (2005 г.), Первой Международной научно-практической конференции «Современные информационные технологии и ИТ-образование» (МГУ, 2005 г.), Международном научно-техническом семинаре «Современные технологии и задачи управления, автоматики и обработки информации» (Алушта, 2006 г.), Всероссийской научно-практической конференции с международным участием «Информационные технологии в обеспечении нового качества высшего образования» (Москва, 2010 г.), 54-й научной конференции МФТИ – Всероссийской молодёжной научной конференции с международным участием «Проблемы фундаментальных и прикладных, естественных и технических наук в современном информационном обществе» (2011 г.), XIX Всероссийской научной конференции "Проблемы информационной безопасности в системе высшей школы" (НИЯУ МИФИ, 2012 г.), научном семинаре ЗАО МЦСТ (2010 г.). Система тестирования «Исток» экспонировалась на Всероссийской выставке научно-технического творчества молодежи НТТМ-2006.

Реализация результатов работы.

Разработанная классификация уязвимостей была использована при анализе защищённости ВК «Эльбрус-3М1» (ЗАО МЦСТ). Разработанная методика обучения безопасному программированию внедрена на кафедре «Компьютерные системы и технологии» НИЯУ МИФИ.

Результаты диссертационной работы в части, касающейся общих принципов построения компьютерных обучающих программ и систем тестирования и реализованных на их основе программных средств, внедрены в учебный процесс НИЯУ МИФИ (факультет Кибернетики и информационной безопасности, факультет Очно-заочного (вечернего) обучения, Институт инновационного менеджмента).

Основные научные результаты работы были получены в процессе выполнения НИР в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2012 гг.

За работы в области создания средств учебного назначения автор в составе коллектива разработчиков награжден Дипломом Всероссийской выставки научно-технического творчества молодежи НТТМ-2006 и тремя Грамотами выставок-конференций «Телекоммуникации и новые технологии в образовании», проводимых в рамках ежегодных Научных сессий НИЯУ МИФИ.

Практическое использование результатов диссертации подтверждено тремя актами о внедрении.

Публикации.

Результаты диссертации опубликованы в 22-х печатных работах, в том числе трёх статьях в журналах, входящих в Перечень ведущих рецензируемых научных журналов и изданий, рекомендованных ВАК. 7 работ

опубликованы без соавторов. Результаты, изложенные в остальных работах, получены при определяющем личном участии автора. На разработанные обучающие программные средства получено два Свидетельства об официальной регистрации программ для ЭВМ.

Структура и объем работы. Работа состоит из введения, четырех глав, заключения, списка использованных источников из 135 наименований и приложения. Общий объем работы составляет 189 страниц и включает 36 рисунков и 22 таблицы.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы, определены цели и задачи исследований, представлены основные положения, выносимые на защиту.

Глава 1 диссертации посвящена анализу современного состояния проблемы уязвимостей компьютерных систем. Рассматриваются существующие в настоящее время методы и средства противодействия таким уязвимостям. Обосновывается актуальность разработки средств противодействия РПВ, использующим уязвимости ПО.

Проведённый анализ сообщений, в которых говорится об успешных хакерских атаках или выпущенных программных «заплатках» позволяет заключить, что увеличивается как количество отдельных атак, так и ущерб, наносимый каждой из них. Так, относительно недавно заявил о себе сетевой червь Stuxnet, функционирование которого привело к физическому разрушению оборудования, задействованного в иранской ядерной программе. Если до Stuxnet можно было считать, что разработкой вирусов и других РПВ занимались, в основном, любители, то этот червь стал без преувеличения настоящим кибероружием, положившим начало новому этапу в развитии РПВ.

Рассмотренные случаи обнаружения и эксплуатации уязвимостей в распространённых по всему миру программных продуктах показывают, что они являются серьезной угрозой безопасности компьютерных систем и в определённом смысле являются препятствием к развитию общества в направлении ускорения обмена информацией и всеобщей компьютеризации.

Хотя в последнее время и стали появляться специальные подразделения, занимающиеся поиском уязвимостей, а также дополнительные этапы при разработке программного продукта, направленные на выявление и устранение уязвимостей, тем не менее, для каждой найденной уязвимости каждая фирма-разработчик ищет свой способ противодействия, в результате меры оказываются недостаточно эффективными или приводят даже к порождению новых уязвимостей. Такой подход не позволяет вырабаты-

вать общие методы устранения не отдельных уязвимостей, а их классов, в результате чего уязвимости остаются даже в продуктах широко известных фирм – разработчиков ПО.

Время, проходящее обычно между моментом обнаружения уязвимостей и моментом её закрытия, может составлять до полугода, поэтому задача написания кода, изначально лишённого уязвимостей, очень актуальна.

В связи с этим необходима система, которая позволила бы выявлять уязвимости по определённым признакам и гарантированно их закрывать. В основу системы поиска уязвимостей должна быть положена классификация уязвимостей с чётко определёнными признаками, характеризующими уязвимый код. В настоящее время такой классификации не предложено, поэтому каждый производитель старается искать уязвимости по собственным методикам, уделяя большее или меньшее внимание отдельным уязвимостям согласно собственным предпочтениям.

В настоящее время существует достаточно много программно-аппаратных подходов к поиску и устранению уязвимостей, однако они не позволяют обнаружить и исправить все уязвимости, требуют знания исходного кода и высокой квалификации программиста, работающего с результатами анализа, а кроме того, многие из предлагаемых методов носят декларативный характер и не доведены до практического воплощения.

Таким образом, необходимо:

– разработать набор правил, обеспечивающих обнаружение уязвимых фрагментов исходного кода программ;

– разработать набор правил, позволяющих реализовывать распространённые операции безопасным образом;

– разработать методы и средства, обеспечивающие безопасное выполнение программ, даже если в них содержатся уязвимости и исходный код для изменения не доступен.

Именно поэтому актуальна задача разработки классификации по механизмам возникновения уязвимостей исходного кода, что позволит разработчику определить методы и средства противодействия уязвимостям в существующих программах и целенаправленно создавать программы, свободные от уязвимостей.

Для снижения числа успешных атак на программные системы или уменьшения их последствий используются различные программно-аппаратные средства. К программным средствам относятся статические сканеры исходного кода, динамические анализаторы уязвимых приложений, надстройки к компилятору (или безопасные диалекты языков программирования), а также надстройки к ядру. Статические анализаторы исходного кода проводят синтаксический анализ кода программы, выявляя небезопасные функции, о чём информируют программиста. К статистическим сканерам относятся такие приложение, как RATS, Flawfinder,

Splint, ITS4 и др. Недостатками сканеров являются высокие требования к квалификации программиста при исправлении найденных уязвимых мест, а также длительное время работы, т.к. процесс итерационен. Динамические анализаторы работают путём многократного запуска тестируемого приложения с автоматически формируемыми входными параметрами, которые выбираются по принципу близости к пограничным значениями (большая длина, максимально возможное целое и т.п.) Такая технология получила название Fuzzing и чаще применяется для web-приложений. К динамическим анализаторам относятся приложения Sharefuzz, OWASP JBroFuzz, Bunny the Fuzzer и др.

Для повышения защищённости приложения используются надстройки к компилятору. При компиляции программы опасные функции могут заменяться на безопасные «обёртки», которые проверяют корректность входных параметров, могут использоваться «сторожевые байты» для выявления факта записи за пределы выделенной памяти, может быть использовано перемешивание данных уязвимой функции для более безопасного их размещения. Наиболее распространёнными надстройками к компилятору являются приложения Stackguard, ProPolice, StackShield, PointGuard, FormatGuard, Libformat и др. Недостатками в работе надстроек к компиляторам являются требование доступности исходного кода для безопасной компиляции, снижение быстродействия программы, а также требования по расстановке специальных символов в программе для возможности проверки компилятором замысла разработчика.

Наконец, надстройки к ядру (например, Raceguard, Systrace, Janus) позволяют осуществлять мониторинг действий недоверенных процессов и при необходимости пресекать их. Существенными их недостатками являются трудности администрирования системы и низкая совместимость различных выполняющихся процессов.

В ряде работ предлагаются аппаратные методы повышения защищённости работающей системы. Среди прочих узкоспециализированных методов ограничения и контроля в них предлагается использование рандомизации в отдельных элементах выполнения процесса. Под рандомизацией в данном случае понимается внесение неопределённости по сравнению с классической архитектурой, в которой данный параметр либо заранее определён, либо рассчитывается на основании известных атрибутов и свойств системы. Основным недостатком таких подходов является противодействие далеко не всем видам атак, основанным на использовании уязвимостей ПО (в первую очередь, переполнения буфера), а кроме того многие из них не позволяют использовать динамически подключаемые библиотеки.

Глава 2 работы посвящена исследованию уязвимостей программного кода. В ходе работы была составлена ER-модель эксплуатации уязвимости, показанная на рис. 1.

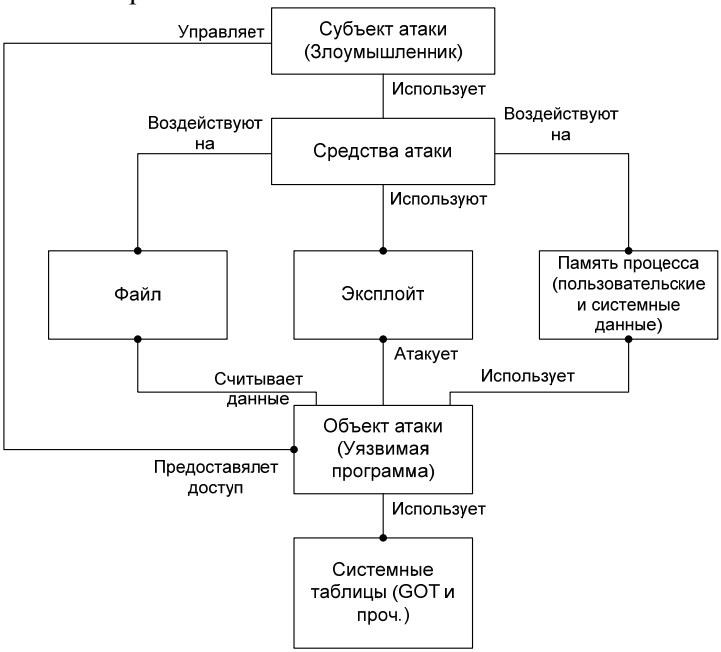

Рис. 1. ER-модель процесса эксплуатации уязвимостей

На основании информации о причинах возникновения уязвимостей, встречающихся в программах, была составлена их классификация по следующим параметрам (рис. 2 - 6): модифицируемой сущности; последствиям атаки; эффективным методам обнаружения; механизму возникновения и размещению вредоносной сущности.

Систематизация уязвимостей позволяет:

1) с высокой достоверностью обнаруживать их в исходных кодах;

2) разработать методы безопасного программирования, исключающие возможность появления уязвимостей в программах;

3) определить механизмы использования уязвимостей, что позволит бороться с ними в исполняемой программе при отсутствии исходного кода.

В ходе работы были исследованы механизмы возникновения уязвимостей, рассмотрены примеры уязвимого и безопасного кода, исследованы принципы использования уязвимостей при проведении атак на компьютерные системы.

Проведена оценка повышения защищённости программы при использовании методики безопасного программирования, которая дала следующие результаты.

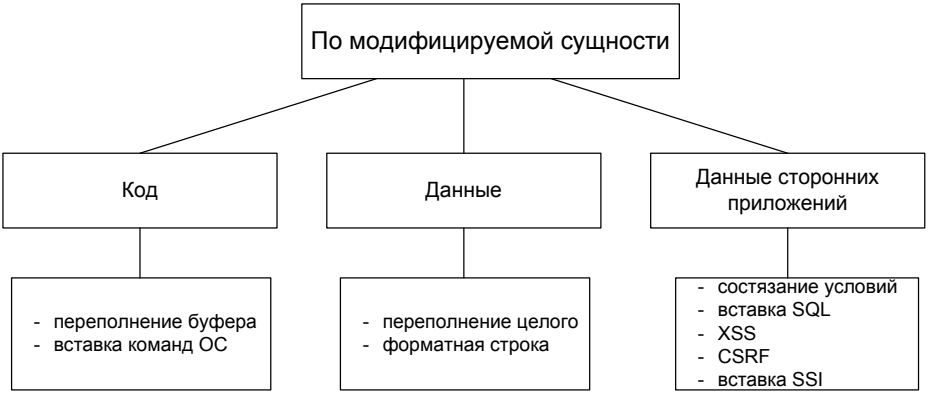

Рис. 2. Классификация уязвимостей по модифицируемой сущности

Рис. 3. Классификация уязвимостей по последствиям атаки с использованием уязвимости

Если представить множество всех уязвимых функций как $F = \{f_1, f_2, ..., f_n\}$, а всех уязвимостей, к которым может привести использование этих функций как $V = \{v_1, v_2, ..., v_m\}$, то определив множество вероятностей, с которыми функция f_i может привести к появлению уязвимости v_j как $P = \{p_i^j\}$ ($i = 1, ..., n, j = 1, ..., m$), можно определить уровень опасности той или иной программы как $D = \sum_{i=1}^{n} (K_i \sum_{j=1}^{m} p_i^j r_j)$, где K_i – число функций i-го вида в программе, а r_j – опасность j-й уязвимости в баллах, определяемая методом экспертных оценок.

Рис. 4. Классификация уязвимостей по механизму возникновения

Рис. 5. Классификация уязвимостей по размещению вредоносной сущности

Рис. 6. Классификация уязвимостей
по эффективному методу обнаружения

Предлагаемый подход позволяет практически полностью устранить ряд уязвимых функций из исходного кода, в результате вероятность, что они приведут к появлению уязвимости, становится равной нулю. Тогда повышение защищённости приложения можно оценить как отношение уровня опасности программы до (D) и после (D') применения методики безопасного программирования. Проведённые исследования показали, что оно составит около 20%.

Наконец, в работе предлагается метод оценки количества уязвимостей, оставшихся в программе после завершения этапа первичного тестирования. Метод основан на модели Миллса, изначально использовавшейся для определения количества ошибок в программном продукте. Суть предлагаемого подхода заключается в том, что необходимо анализировать каждый класс уязвимостей независимо от остальных. После внесения в программу уязвимостей определённого класса и их последующего поиска, на основании сравнения количества внесённых и найденных уязвимостей можно сделать вероятностную оценку количества оставшихся уязвимостей, присутствовавших в программе до начала тестирования.

Преимуществом применения модели Миллса для оценки остающихся в программе уязвимостей по сравнению с классическим использованием этой модели является возможность учета чётких признаков вносимых уязвимостей, выявленных в данной работе на этапе их классификации. По сравнению с разнообразными и слабоформализованными ошибками, которые требуется искусственно вносить для обычной модели Миллса, это в

конечном счете приведет к более обоснованным выводам.

Количество уязвимостей *i*-го типа, оставшихся в программе после применения предложенной методики, можно вычислить следующим образом

$$k_i = \frac{n_i}{v_i} s_i - n_i ,$$

где k_i – количество оставшихся «собственных» уязвимостей *i*-го типа (то есть существовавших в программе до внесения дополнительных уязвимостей);

s_i – количество вносимых уязвимостей *i*-го типа;

v_i – количество обнаруженных уязвимостей *i*-го типа из числа внесённых;

n_i – количество обнаруженных уязвимостей *i*-го типа из числа «собственных».

Мера достоверности рассчитанного количества оставшихся уязвимостей может быть определена в общем случае (если удалось обнаружить не все из внесённых уязвимостей) по формуле

$$C_i = \begin{cases} 1, & \text{при } n_i > k_i \\ \binom{S}{j_i - 1} \Big/ \binom{S_i + k_i + 1}{k_i + j_i}, & \text{при } n_i \le k_i, \end{cases}$$

где j_i – количество обнаруженных в ходе тестирования уязвимостей *i*-го типа из числа «собственных» ($j_i \le s_i$).

Уровень доверия рассчитывается отдельно по каждому классу уязвимостей, поскольку их анализ производится независимо.

Глава 3 посвящена описанию предлагаемого метода аппаратной защиты от эксплойтов, осуществляющих вставку вредоносного кода в программу. Метод основан на динамическом стохастическом преобразовании кодов операции инструкций процессора при занесении программы в оперативную память и обратном преобразовании этих кодов при вызове программы на исполнение.

Программа хранится на внешнем носителе в обычном виде. При загрузке в оперативную память происходит рандомизация кодов операции, и в оперативную память программа заносится в преобразованном виде. При выполнении каждой очередной команды происходит её обратное преобразование к исходному виду, интерпретация процессором и исполнение. Преобразование происходит на основании таблиц замен, для перемешивания которой используется схема, представленная на рис. 7. Изна-

чально в каждую ячейку таблицы замен записан её собственный адрес, затем ГПСЧ формирует два адреса, которые последовательно записываются в Регистры, после чего происходит обмен значениями между ячейками, адресуемыми Регистрами.

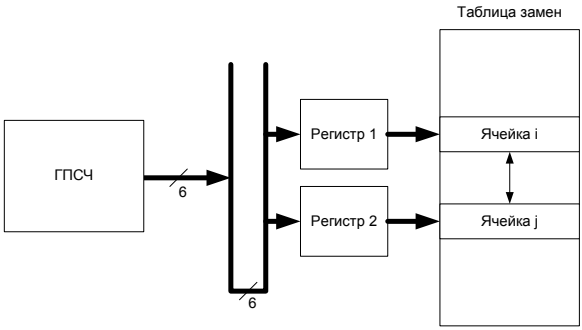

Рис. 7. Схема формирования таблицы замен

Проведена оценка временных и аппаратных затрат, связанных с реализацией предложенного метода. Показано, что для реализации динамической рандомизации потребуется дополнительный объём памяти около 64K, при этом замедление скорости работы составит примерно 6%.

Проведена оценка трудоемкости реализации вариантов компрометации системы, построенной с применением предлагаемых аппаратных модификаций процессора. Таких вариантов три: считывание сегмента кода из оперативной памяти, где он хранится в преобразованном виде, после чего его можно сравнить с исходным кодом, затем подбор таблицы замен методом грубой силы и, наконец, восстановление таблицы замен с помощью статистического анализа кода преобразованной программы. Первый способ нереализуем ввиду того, что из другого процесса память сегмента кода недоступна (даже на уровне привилегий 0). Вероятность подбора кодов 9 команд (именно столько используется в shell-коде, который вызывает командную строку) при реализации второго способа составляет $\approx 10^{-16}$. Наконец, третьему способу противопоставляется особенность формата команд x86, который предполагает различную длину каждой команды: не восстановив первую, невозможно найти начало следующей, и, следовательно, не представляется возможным собрать статистику по частоте появления команд.

Проведена оценка степени перемешивания системы команд в зависимости от шага перемешивания в соответствии с предложенным алгоритмом. При этом было выявлено, что степень перемешивания в 98,5% является практически достижимым максимумом. При достижении степени переме-

шивания в 97,5% дальнейшее её увеличение требует существенного повышения числа шагов перемешивания.

Предложен алгоритм обхода программы в скомпилированном виде для эмуляции ее разбора процессором с учётом команд, нарушающих естественный порядок выборки команд. Данный алгоритм был использован при определении границ применимости предложенного метода рандомизации.

Предложен аппаратный метод противодействия эксплойтам, построенным по схеме “ret to libc”. Метод основан на преобразовании байтов системной информации, сохраняемой в стек при передаче управления в вызываемую функцию.

Рандомизация системы команд эффективна в борьбе с эксплойтами, использующими непосредственную вставку кода. Однако существует и другой способ эксплуатации уязвимости переполнения буфера, называемый «ret to libc», для реализации которого злоумышленнику достаточно перезаписать адрес возврата и сохранённый кадр стека, при этом новые команды не внедряются.

Для защиты значений системных регистров, сохраняемых в стек при переключении контекста, предполагается использовать их стохастическое преобразование при выполнении неявных команд `push` и `pop`. Эти команды не присутствуют в исходном коде программы, но выполняются процессором при переключении контекста.

При помещении в стек системные данные рандомизируются и становятся недоступны для чтения и предсказуемого изменения.

Выделены критерии эффективности разработанной системы. Они представлены в таблице 1.

Для выявления наиболее приоритетных критериев была использована процедура, принятая в методе анализа иерархий. При этом составляется матрица A парных сравнений выбранных критериев: $A = \{a_{ij}\}$, где $i, j = 1, ..., N$, и N – число критериев (в рассматриваемом случае семь). Сравнение осуществляется по 9-ти бальной шкале, где оценки проставляются от 1 до 9 таким образом, что 1 означает равную предпочтительность двух критериев (два критерия одинаково значимы), …, а 9 - один из критериев очевидно и неоспоримо значимее другого.

Таблица 1. Критерии эффективности предложенного метода рандомизации

Условное обозначение критерия	**Наименование критерия**	**Содержательное описание критерия**
K_1	Степень покрытия множества уязвимостей	Критерий, определяемый на основе количества и опасности блокируемых уязвимостей
K_2	Снижение производительности системы	Определяет, насколько велики накладные расходы (используемая память, время работы процессора)
K_3	Объём защищаемого кода	Система защищает только код пользовательских приложений, либо также защищает код системных библиотек и программ ядра
K_4	Сложность компрометации	Определяет стойкость системы защиты
K_5	Затраты на внедрение	Определяются расходами на проектирование системы и стоимостью её реализации непосредственно на ЭВМ
K_6	Объём дополнительного оборудования	Поскольку все блоки и таблицы замен выполняются аппаратным способом, требуется дополнительные логические элементы на кристалле, обеспечивающую рандомизацию
K_7	Требования к обслуживающему персоналу	Определяет минимальный уровень квалификации, которым должен обладать персонал, использующий предлагаемую систему

Для определения вектора приоритетов выбранных критериев необходимо найти собственный вектор матрицы A. Расчёт компонентов собственного вектора происходит итеративно путём последовательного

возведения матрицы парных сравнений в степени с вычислением по следующим формулам:

$$w = \{w_i\},\ i = 1, ..., N$$

$$w_i = \frac{\sum_{j=1}^{N} a_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}} \qquad (1)$$

Процесс расчета заканчивается, когда выполнено условие:

$$\max_{i=1}^{N}\left(\left|w_i^T - w_i^{T-1}\right|\right) < \delta$$

где w_i^T - i-й компонент вектора (1) для матрицы парных сравнений на шаге T, а δ - заданная точность вычислений.

В результате определения вектора приоритетов критериев эффективности были получены следующие веса для выбранных критериев (табл. 2):

Таблица 2. Веса критериев эффективности предложенного метода

Критерий	K_1	K_2	K_3	K_4	K_5	K_6	K_7
Вес	0,28	0,06	0,14	0,39	0,04	0,05	0,04

Таким образом, были выделены следующие наиболее важные критерии:

1) сложность компрометации системы защиты;
2) степень покрытия множества уязвимостей;
3) объём защищаемого кода (пользовательские программы, ядро и др.).

Для наиболее важного критерия – сложности компрометации системы защиты – дополнительно были определены пути его максимизации.

Показаны преимущества предложенного метода рандомизации по сравнению с другими системами.

Рассмотрены вопросы реализации рандомизационного ГПСЧ на основе трех блоков стохастического преобразования (*R*-блоков). Предложена модель генератора, соответствующего двум двоичным примитивным многочленам, определяющим вид функции обратных связей.

Был выбран и обоснован метод подготовки таблиц замен, основанный на использовании ГПСЧ с *R*-блоками в цепи обратной связи (RFSR). Традицион-

но RFSR строятся на основе одного образующего многочлена. В работе проведено исследование вопросов реализации ГПСЧ, соответствующего двум образующим многочленам. Рассмотрены варианты реализации ГПСЧ по схемам Фибоначчи и Галуа. Выбранный вариант реализации генератора имеет следующие достоинства:

- наличие нелинейных функций обратной связи и выхода;
- с выходов отдельных регистров генератора снимаются не сдвинутые копии одной и той же последовательности, а различные последовательности псевдослучайных чисел;
- сигналы с выходов *R*-блоков не поступают в цепь обратной связи, иначе говоря, выходы стохастических сумматоров не доступны для анализа, а сигналы обратной связи ГПСЧ не проходят на выход.

Рассмотрен пример построения генератора для случая, когда в качестве образующих многочленов были взяты $\varphi_1(x) = x^7 + x^4 + 1$ и $\varphi_2(x) = x^5 + x^2 + 1$. При таких значениях многочленов получаются следующие уравнения работы ГПСЧ, построенного по схеме Галуа

$$\begin{cases} Q_0(t+1) = Q_6(t) \oplus Q_4(t), \\ Q_j(t+1) = Q_{j-1}(t),\ j = 1,\ 3,\ 5,\ 6, \\ Q_4(t+1) = R(Q_6(t),\ Q_3(t)), \\ Q_2(t+1) = R(Q_4(t),\ Q_1(t)), \\ Out(t) = R(Q_0(t),\ Q_5(t)), \end{cases}$$

где $Q_i(t)$ – содержимое *i*-го регистра генератора в момент времени *t*, $R(A, B)$ – результат стохастического преобразования, *A* – преобразуемое значение, *B* – параметр преобразования. Статистическое тестирование ГПСЧ по методике NIST подтвердило обоснованность сделанного выбора.

Глава 4 содержит описание результатов исследования защищённости отечественного высокопроизводительного комплекса «Эльбрус-3М1», выполненного с применением разработанной в диссертации классификации уязвимостей и методики безопасного программирования. На основании проведённого исследования были составлены рекомендации программистам для повышения защищённости разрабатываемого ПО.

Также было проведено исследование режима двоичного транслятора, в котором может работать «Эльбрус-3М1». Даны рекомендации по повышению защищённости системы, работающей в этом режиме.

Описана предлагаемая методика обучения безопасному программированию. Компьютерные обучающие программы, предназначенные для обучения безопасному программированию, должны быть нацелены на решения двух взаимосвязанных задач.

– Во-первых, они должны прививать студенту навыки безопасного программирования, главным из которых является использование только безопасных функций.

– Во-вторых, они должны обеспечивать возможность анализа потенциально небезопасной программы и определения механизмов, с помощью которых злоумышленник может использовать существующие уязвимости.

В связи с эти программы, обучающие безопасному программированию, имеют ряд особенностей по сравнению с прочими обучающими программами. Они обусловлены тем, что здесь практически невозможна полностью автоматическая генерация тестовых заданий как для тренировочного, так и для контрольного выполнения. Это связано с тем, что задания должны иметь вид осмысленной программы. Поэтому многовариантность заданий должна обеспечиваться сочетанием принципа автоматической генерации задания с использованием базы данных заданий. Кроме того, такие обучающие программы должны, во-первых, иметь развитую систему подсказок, во-вторых, такие подсказки должны использоваться не только на этапе выработки навыков в ходе тренировочной работы, но и при выполнении контрольных заданий, и в-третьих, использование подсказки должно приводить к начислению определённого числа штрафных баллов.

На рис. 8 приведён общий вид экранной формы разработанного электронного урока. В первом задании, которое представлено на рисунке, студенту требуется найти уязвимую функцию и заменить её на безопасный эквивалент. Задание состоит из нескольких программ, которые выбираются из базы. При извлечении программы из базы случайным образом выбирается один из двух вариантов, а именно, будет ли функция, потенциально содержащая уязвимость, представлена в опасном или безопасном варианте.

Целью второго задания является получение навыков выявления уязвимостей, которые необходимы при проведении теста на проникновение.

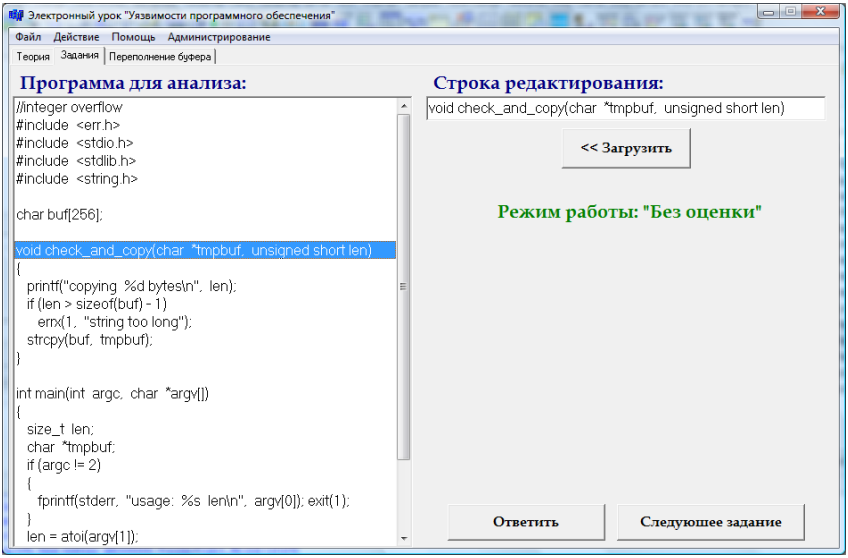

Рис. 8. Общий вид экранной формы электронного урока

Неотъемлемой частью обучения является контроль усвоения учебного материала, который необходимо проводить не только во время сдачи студентом экзамена, но и периодически в течение семестра. Для автоматизации этого процесса была разработана система тестирования «ИСТОК», имеющая широкие возможности по настройке критериев выставления оценки, ведения протоколов тестирования и формирования тестов на основе базы вопросов.

В заключении отражены основные результаты, полученные в данной диссертационной работе.

В приложении приведены примеры уязвимого и безопасного кода для уязвимостей, исследуемых в работе, а также приведены расчёты вероятности повторной генерации того же тестового задания для электронного урока.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В диссертации получены следующие основные результаты:

1) проведено исследование уязвимостей программного кода, выявлены основные методы их обнаружения и основные способы устранения;

2) предложена классификация исследованных уязвимостей по следующим параметрам: по механизмам возникновения, по модифицируемой сущности, по последствиям атаки с использованием уязвимости, по размещению вредоносной сущности, по эффективному методу обнаружения;
3) разработан метод аппаратной защиты от эксплойтов, осуществляющих вставку вредоносного кода в программу; метод основан на динамическом стохастическом преобразовании кодов операции инструкций процессора при занесении программы в оперативную память и обратном преобразовании этих кодов при вызове программы на исполнение;
4) впервые проведён анализ защищённости отечественного высокопроизводительного вычислительного комплекса «Эльбрус-3М1»; даны рекомендации по повышению защищённости ВК; разработаны тестовые модели РПВ, функционирующие в среде ВК «Эльюрус-3М1»;
5) разработана методика безопасного программирования, позволяющая создавать программы, свободные от наиболее распространённых уязвимостей; описаны механизмы появления уязвимостей, даны примеры уязвимого и безопасного кода, принципы их использования РПВ, основные методы противодействия уязвимостям различных классов;
6) предложен метод оценки эффективности мероприятий по противодействию уязвимостям, основанный на использовании модели надежности ПО Миллса; метод позволяет оценить вероятное количество уязвимостей, остающихся в программе после этапа её тестирования;
7) разработана методика обучения безопасному программированию, нацеленная на увеличение компетенций обучаемых в области разработки защищённого ПО; методика подкреплена учебно-методическим пособием, компьютерными средствами обучения и тестирования знаний.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

В журналах, рекомендованных ВАК, опубликованы следующие работы:

1. Гуров В.В., Гуров Д.В., Иванов М.А., Шустова Л.И. Технология безопасного программирования и особенности ее преподавания в вузе. – Дистанционное и виртуальное обучение, 09.2010. – С. 35-44.

2. Гуров Д.В. Применение аппаратного шифрования при работе со стеком для защиты от эксплойтов. – Безопасность информационных технологий, № 3, 2011. – С. 37-41.

3. Гуров Д.В., Иванов М.А. Динамическая рандомизация системы команд микропроцессора для защиты от эксплойтов. – Вестник Рязанского государственного радиотехнического университета. № 1 (выпуск 35) 2011 г. – С. 81-86.

Основные положения диссертации изложены также в следующих печатных трудах:

4. Разрушающие программные воздействия: Учебно-методическое пособие / Васильев Н.П., Вельмякина Е.В., Гуров Д.В. и др.; Под ред. М.А. Иванова. – М.: НИЯУ МИФИ, 2011. – 328 с.

5. Гуров Д.В. Разработка электронных уроков для обучения специальным дисциплинам. – Современные проблемы информатизации в непроизводственной сфере и экономике: Сб. трудов. Вып. 10 / Под ред. д.т.н., проф. О.Я. Кравца. – Воронеж: Изд-во «Научная книга», 2005. – С. 37-38.

6. Гуров Д.В. Использование метода проверки логики результата при анализе ответа пользователя. – Научная сессия МИФИ-2005. Сборник научных трудов. – М.: МИФИ, 2005. Т. 14: Конференция "Молодежь и наука". Компьютерные науки. Информационные технологии. Экономика и управление. – С. 94-95.

7. Гуров В.В., Гуров Д.В., Кузнецова П.В., Михайлов Д.М. Использование компьютерных обучающих программ при обучении техническим дисциплинам. – Образование через науку. Сборник тезисов докладов Международного симпозиума " Образование через науку", посвященную 175-летию МГТУ им. Н.Э. Баумана. – М.: Изд. МГТУ им. Баумана, 2005. – С. 44-45.

8. Гуров Д.В. Разработка инструментальных средств подготовки и проведения электронного тестирования. – Сборник докладов Первой Международной научно-практической конференции. Под ред. проф. В.А. Сухомлина. – М.: МАКС Пресс, 2005. – С. 129-130.

9. Гуров Д.В., Кузнецова П.В., Михайлов Д.М. Контроль знаний через сеть Internet и оптимизация конструирования тестов. – Современные технологии и задачи управления, автоматики и обработки информации: Труды XV Международного научно-технического семинара. Сентябрь 2006 г., Алушта. – М.: МИФИ, 2006. – С. 178.

10. Гуров В.В., Гуров Д.В. Использование электронных образовательных ресурсов в учебном процессе технического вуза. – Труды Все-

российской научно-практической конференции с международным участием «Информационные технологии в обеспечении нового качества высшего образования» (14–15 апреля 2010 г., Москва). – М.: Исследовательский центр проблем качества подготовки специалистов, 2010. – С. 53-58.

11. Гуров Д.В. Анализ статистики обнаружения уязвимостей в продуктах семейства Microsoft Office. – Научная сессия НИЯУ МИФИ-2010. Конференция "Молодежь и наука". Компьютерные науки. Информационные технологии. – М.: НИЯУ МИФИ, 2010. – С. 193-194.

12. Гуров Д.В. Однобайтовое переполнение при использовании механизмов шлюзов. – Научная сессия НИЯУ МИФИ-2010. Аннотации докладов. В 3 тт. Т. 3. Информационно-телекоммуникационные системы. Проблемы информационной безопасности в системе высшей школы. Экономика, управление и нормативно-правовые вопросы высоких технологий. Инновационные образовательные технологии в Национальном исследовательском ядерном университете. М.: НИЯУ МИФИ, 2010. – С. 119.

13. Гуров В.В., Гуров Д.В. Оценка эффективности аппаратных методов защиты информации. – Труды 54-й конференции МФТИ «Проблемы фундаментальных и прикладных естественных и технических наук в современном информационном обществе» Информационные бизнес-системы. – М.: МФТИ, 2011. – С. 25-26.

14. Гуров Д.В. Программные средства обучения безопасному программированию. – XV Международная телекоммуникационная конференция молодых учёных и студентов «МОЛОДЁЖЬ И НАУКА». Тезисы докладов. В 3-х частях. Ч. 3. М.: НИЯУ МИФИ, 2012. – С. 18-19.

15. Гуров Д.В., Гуров В.В., Криводаев А.В., Матюшенков Н.В. Пакет компьютерных обучающих программ по курсу "Организация ЭВМ" – Свидетельство об официальной регистрации программы для ЭВМ №2007611495. Зарегистрировано в реестре программ для ЭВМ 10.04.2007.

16. Гуров В.В., Гуров Д.В., Кузнецова П.В., Михайлов Д.М. Программа «Интерактивная система тестирования на основе компьютерных технологий ИСТОК». – Свидетельство об официальной регистрации программы для ЭВМ №2006613218. Зарегистрировано в реестре программ для ЭВМ 13.09.2006.