УДК 004.75

В.А. Десницкий

КОНФИГУРИРОВАНИЕ ВСТРОЕННЫХ И МОБИЛЬНЫХ УСТРОЙСТВ НА ОСНОВЕ РЕШЕНИЯ ОПТИМИЗАЦИОННОЙ ЗАДАЧИ

Десницкий В.А. **Конфигурирование встроенных и мобильных устройств на основе решения оптимизационной задачи.**

Аннотация. Исследование посвящено изучению вопросов проектирования и анализа комбинированных механизмов защиты сложных коммуникационных систем со встроенными и мобильными устройствами. В работе вводится понятие конфигурации устройства, которая представляет собой комбинацию компонентов защиты, развертываемых для поддержки безопасности устройства, а также предоставляемых им программных сервисов. На основе решения оптимизационной задачи с учетом функциональных и нефункциональных свойств отдельных компонентов защиты производится поиск наиболее эффективной конфигурации. Анализ эффективности предложенного подхода к конфигурированию осуществляется на основе экспериментов путем его сравнения с альтернативными стратегиями конфигурирования. В частности, производится сравнение со стратегией «произвольного конфигурирования», которая представляет сценарий «ручного» конфигурирования, проводимого оператором системы без использования автоматизированных средств перебора и оценки возможных конфигураций.

Ключевые слова: конфигурация, встроенное устройство, мобильное устройство, компоненты защиты.

Desnitsky V.A. **Configuring embedded and mobile devices on the basis of solving an optimization problem.**

Abstract. The paper encompasses design and analysis of combined protection mechanisms applied to complex communication systems containing embedded and mobile devices. A notion of configuration is proposed in order to represent a combination of particular security building blocks deployed to support security of the device as well as software services it provides. Starting from functional and non-functional properties of specific building blocks the optimization problem allows arranging the search of the most effective configuration. Effectiveness evaluation of the configuring approach is conducted by means of its comparing with alternative configuring strategies, including “manual” configuring scenarios realized by an operator of the system without using any automated tools for enumeration and evaluation of configurations.

Keywords: configuration, embedded device, mobile device, security building block.

1. Введение. Работа направлена на исследование вопросов безопасности комбинированных механизмов защиты сложных коммуникационных систем. Объектом исследования являются комбинированные механизмы конфигурирования распределенных информационных систем со встроенными и мобильными устройствами. Отличительными особенностями таких систем являются, во-первых, мобильность устройств, входящих в систему и, во-вторых, ограниченность ресурсов

устройств и вытекающая из этого их слабая производительность [3, *10*, *13*].

К встроенным устройствам относятся такие электронные, мультимедиа, бытовые и иного вида устройства, в которые встраиваются специализированные информационно-вычислительные блоки, которые ответственны за решение задач управления функциями устройства, организации обработки и обеспечения работы устройства, поддержки пользовательского интерфейса и других задач [1, 12, 16]. Примеры встроенных устройств: переносные средства управления бизнес-процессом, портативные измерительные приборы, сканнеры штрих-кодов, устройства ввода и хранения информации распознавания речи, устройства поддержки работы геоинформационных систем. Под мобильными устройствами в первую очередь понимаются переносные коммуникационные устройства, такие как мобильные телефоны и мультимедийные смартфоны.

Мобильность устройства предоставляет широкие возможности атакующему проводить различные атаки, в том числе через прямое подключение к интерфейсам и элементам устройства. В свою очередь, ограниченность ресурсов влечет сложность применения традиционных криптографических и других средств защиты, которые используются для обеспечения защиты персональных ЭВМ и серверных станций. Поэтому требуются новые подходы к проектированию механизмов защиты распределенных информационных систем со встроенными и мобильными устройствами, которые смогли бы обеспечить разумный компромисс между суммарной производительностью системы и ее защищенностью. Одним из путей достижения такого компромисса является использование механизма конфигурирования защиты путем выбора наиболее эффективных наборов компонентов, реализующих требуемые свойства безопасности.

Важность разработки и исследования таких механизмов защиты обуславливается появлением, а также тенденцией к стремительному увеличению количества устройств, осуществляющих коммуникации в сети Интернет и управляющиеся удаленно посредством беспроводных протоколов соединения – т.н. «Интернет вещей» (“Internet of Things”) [*14*]. Осуществляя коммуникации в неконтролируемом и потенциально опасном окружении, такие системы подвержены как специализированным, так и универсальным сетевым атакам, включая атаки, относящиеся к классу DDoS-атак [*17*]. Поэтому особенно важным становится вопрос построения эффективных механизмов защиты, нацеленных на [2, 8, 18] противодействие атакам со стороны потенциальных

нарушителей при существенных ограничениях на объемы системных ресурсов, предоставляемых устройствами для осуществления функций защиты.

Под конфигурированием понимается процесс нахождения и последующего применения оптимального набора специализированных программных компонентов защиты, применяемых для обеспечения безопасности информационной системы, которые, во-первых, реализуют все необходимые функциональные требования защиты и, во-вторых, позволяют достичь наиболее эффективного расхода ресурсов устройства в процессе осуществления защиты.

Предлагаемая стратегия конфигурирования основывается на применении методов системного анализа, математического моделирования, теории защиты информации, объектно-ориентированного проектирования и анализа, формальной логики, экспертных оценок, методов решения оптимизационных задач, а также средств графического моделирования систем на основе языка UML.

Статья состоит из разделов: *введение*; *конфигурирование*; *формальная постановка задачи*; *пример применения*; *архитектура механизма конфигурирования*; *оценка эффективности конфигурирования*; *заключение*; *литература* и включает следующие иллюстрации: *5 рисунков*, *2 таблицы, 3 формулы*.

2. Конфигурирование. Процесс конфигурирования нацелен на поиск и применение некоторого множества компонентов защиты, которые, во-первых, реализуют все необходимые функциональные свойства защиты информационной системы, и, во-вторых, удовлетворяют ограничениям, накладываемым устройством на объемы доступных ресурсов. Разрабатываемое программное средство конфигурирования является инструментом «времени разработки» и ориентировано на решение следующих задач, выполняемых разработчиком устройства или системным интегратором: проверка допустимости заданной конфигурации; проверка оптимальности заданной конфигурации; нахождение оптимальной конфигурации.

Под допустимой понимается такая конфигурация компонентов защиты, которая удовлетворяет всем нефункциональным ограничениям устройства и реализует все требуемые устройством функциональные свойства защиты. В целом, процесс конфигурирования состоит из следующих стадий:

- формирование репозитория компонентов защиты, а также доменно-специфичного описания, которое включает средства для сопоставления требованиям к защите информаци-

онной системы конкретных элементов формального представления;

- формальная спецификация набора нефункциональных свойств («дерева свойств»), на основе которых производится выделение допустимых конфигураций, а также набора функциональных свойств, определяющих требования к безопасности устройства;
- формирование показателей на основе нефункциональных свойств для формирования количественных оценок ресурсов устройства, которые требуются для выполнения компонентов защиты;
- построение алгоритмов получения значений показателей, с помощью аналитического подхода, метода экспертных оценок, экспериментальных методов по оценке расходования системных ресурсов различными компонентами защиты;
- формальная математическая постановка оптимизационной экстремальной задачи для нахождения наиболее эффективных конфигураций компонентов защиты, включая определение целевой функции оптимизационной задачи и необходимых ограничений для случаев однокритериальной и многокритериальной оптимизации;
- построение критериев оптимальности как средства распознавания наиболее эффективных конфигураций;
- определение путей решения оптимизационной задачи включая методики, позволяющие снизить вычислительную сложность процесса поиска решения на основе метода динамического программирования.

Процесс конфигурирования имеет следующие входные и выходные параметры. Входными данными являются:

- данные о функциональных свойствах защиты, которые должны быть обеспечены устройству посредством применения компонентов защиты при помощи процесса конфигурирования;
- ограничения на нефункциональные свойства устройства;
- описание имеющихся компонентов защиты, хранящихся в рамках репозитория;
- дерево свойств, представляющее совокупность рассматриваемых функциональных и нефункциональных свойств компонентов защиты и значений нефункциональных свойств, а

также свойств совместимости программно-аппаратной платформы.

Выходными данными процесса конфигурирования являются:

- множество допустимых конфигураций защиты;
- оптимальная конфигурация или набор оптимальных конфигураций.

Отметим, что оптимальных конфигураций может быть несколько, как в случае, если для двух или более конфигураций их множества свойств и значений свойств совпадают, так и в случае, если конфигурации рассматриваются как эквивалентные в соответствии с некоторым критерием оптимальности.

В отличие от существующих моделей комбинированных механизмов защиты, таких как Pioneer [19], SWATT [20], Genuinity [11], предоставляющих защиту информационных систем на основе функциональных свойств защиты, таких как необходимость реализации функций удаленной аттестации, метода «барьерного разделения», методов обфускации, метода динамического замещения и других, предлагаемый процесс конфигурирования учитывает также и нефункциональные свойства защиты, включая свойства потребления объемов того или иного системного ресурса устройств.

3. Формальная постановка задачи. Решаемая оптимизационная задача является в общем случае многокритериальной экстремальной задачей с заданным набором ограничений. Ее математическая постановка формулируется с использованием следующего теоретико-множественного представления:

$$Objective(N) \rightarrow min;$$
$$Constr(F, N);$$

где $Objective$ обозначает целевую функцию на множестве конфигураций, зависящую от множества нефункциональных показателей защиты N; F представляет множество функциональных показателей, которые используются в качестве элементов описания требований безопасности; $Constr$ обозначает серию ограничений на функциональные и нефункциональные свойства, а также на свойства программно-аппаратной совместимости. Цель задачи – найти экстремальное значение, представляющее конфигурацию, на которой достигается оптимум задачи.

Решение оптимизационной задачи позволяет найти наиболее эффективную конфигурацию на множестве всевозможных конфигураций

защиты. Эффективность может пониматься как наиболее экономичное расходование некоторого заданного ресурса устройства или набора ресурсов.

Оптимизационная задача включает: целевую функцию, задающую критерий оптимизации, и набор ограничений. Рассматриваются два вида ограничений оптимизационной задачи:

- Ограничения программно-аппаратной совместимости компонента защиты и платформы устройства, причем такие ограничения представляют бинарные свойства, определяющие тип и версию необходимой операционной системы устройства или виртуальной машины, версию коммуникационных протоколов, поддерживаемых устройством, и другие характеристики;
- Ограничения «сверху» на значения численных показателей ресурсов, получаемые исходя из спецификации устройства (так для ресурса оперативной памяти устройства выделяется показатель «объем расходуемой памяти»).

Используются следующие виды критериев оптимальности:

- Минимизация показателя, определяющего объемы некоторого ресурса, который выделяется устройством на выполнение функций защиты;
- Оптимальность на основе «цепочки свойств», при которой предыдущий критерий рассматривается последовательно для каждого элемента последовательности нефункциональных свойств;
- Оптимальность на основе интегрального показателя расходов устройства;
- Оптимальность на основе DEA-метода.

Оптимальность на основе «цепочки свойств» предполагает процедуру выбора конфигураций последовательно путем анализа некоторой заданной последовательности нефункциональных свойств в порядке убывания их предпочтения при конфигурировании информационной системы. Критерий оптимальности на основе интегрального показателя расхода ресурсов устройства предполагает минимизацию выражения, приведенного на следующей формуле:

$$min\{ dev_i \mid i \};$$

где i задает номер рассматриваемого нефункционального свойства; dev_i определяется как отклонение значения свойства n_i от его ограничения сверху $constr_i$ в процентном соотношении, определяемого объемами соответствующего ресурса устройства:

$$dev_i = \frac{constr_i \quad n_i}{constr_i} \quad 100\%.$$

DEA-метод характеризуется, в первую очередь, своей наглядностью, однако, вопрос его точности требует дальнейшего рассмотрения [*15*].

В работе рассматривается несколько путей решения оптимизационной задачи:

- Решение *методом исчерпывающего поиска* [*7*], характеризующийся относительной простой реализации, однако в процессе последовательного перебора при большом количестве рассматриваемых компонентов защиты метод потребует значительных временных затрат, что может быть особенно актуально в случае систем реального времени;
- Применение *оптимизаций процесса поиска решения*, позволяющих ускорить работу метода исчерпывающего поиска путем сужения множества перебираемых конфигураций за счет проверки оптимизационных ограничений для групп конфигураций;
- Решение с использованием *методов динамического программирования*. Сведение оптимизационной задачи к классической дискретной «задаче о рюкзаке» или одной из ее разновидностей [*7*].

4. Пример применения. В качестве примера применения предложенного подхода к конфигурированию в работе рассматривается мобильное коммуникационное устройство «смартфон», имеющее постоянную связь с сетью Интернет и управляющееся на основе мобильной операционной системы или виртуальной машины. На смартфон устанавливается прикладное программное обеспечение, реализующее клиентскую сторону работы специализированного мультимедийного сервиса по предоставлению пользователю устройства функций скачивания, воспроизведения и хранения звуковых данных в формате *mp3*. Для поддержки безопасности программы в нее встраивается комбинированный механизм защиты, который должен быть сконфигурирован. Для обеспечения работы программы выделяются ресурсы устройства, которые расходуются как на выполнение целевых функций программы, так и на функции защиты. К целевым функциям, выполняемым на устройстве, можно отнести следующие функции:

- поддержка сменных носителей информации miniSD для хранения пользовательской информации;

- поддержка беспроводных каналов связи с удаленным сервером в рамках сетей типа *3G* или *4G*.

Задача конфигурирования: выбор инструментов и средств защиты из множества имеющихся так, чтобы защита была бы оптимальной с точки зрения эффективности расхода ресурсов устройства на обеспечение функций защиты. Определяются следующие угрозы безопасности устройства:

- несанкционированное получение данных в процессе их передачи на устройство по беспроводным каналам связи или с него (передаваемых на устройство звуковых файлов);
- несанкционированное копирование и последующее воспроизведение звуковых файлов на других устройствах;
- несанкционированный доступ к устройству и к функциям ПО;
- незаконная модификация ПО.

В соответствии с угрозами, приведенными выше, рассматриваются следующие атаки:

- перехват и криптоанализ трафика при обмене с удаленным сервером путем прослушивания беспроводных каналов связи (в частности, извлечение звуковых файлов, паролей и другой служебной информации);
- анализ хранилища данных на miniSD-карте и извлечение из него конфиденциальных данных;
- атаки «обратной разработки» (reverse engineering attacks) [4], включающие исследование программного кода и поиск уязвимостей.

В рамках процесса конфигурирования учитываются те компоненты защиты, которые функционируют на протяжении не всего сценария работы устройства, а однократно. Процедура аутентификации пользователя может быть выполнена единожды на начальном этапе при создании сессии соединения и поэтому не учитывается при конфигурировании.

В качестве базовых компонентов защиты рассматриваются компоненты, обеспечивающие:

- Защиту беспроводного канала передачи данных (HTTPS-протокол, протокол TCP/IP и алгоритмы симметричного шифрования AES, 3DES);
- Защиту данных на сменном носителе [21] (Симметричное шифрование носителя);

- Защиту ПО на основе удаленной аттестации (контроль контрольных сумм);
- Защиту ПО путем реализации исполнения ПО с применением шифрования (метод Аусмита [5]);
- Защиту ПО на основе разделение кода (Метод барьерного разделения кода [6]).

Таблица 1. **Функциональные и нефункциональные свойства защиты и свойства программно-аппаратной совместимости**

Функциональные свойства защиты	Нефункциональные свойства	Свойства совместимости программно-аппаратной платформы
применение безопасных коммуникационных протоколов для поддержки конфиденциальности передаваемых данных	объем расходуемой оперативной памяти	Поддержка определенной ОС/виртуальной машины
применение симметричного шифрования для поддержки конфиденциальности данных, сохраняемых на носителе	величина пропускной способности сетевого интерфейса	Поддержка определенного сетевого коммуникационного протокола
реализация функции защиты: удаленная аттестация	затраты на объемы данных, сохраняемых локально	
реализация функции защиты «исполнение ПО с применением шифрования»		
реализация функции «разделение кода»		

В таблице 1 приведены значения рассматриваемых функциональных и нефункциональных свойств, а также свойств совместимости программно-аппаратной платформы и компонента защиты.

Процедуры получения значений нефункциональных свойств проводятся экспериментально путем моделирования поведения компонентов защиты и измерения значений расхода потребления ресурсов для каждого из них на физической реализации устройства.

5. Архитектура механизма конфигурирования. Разрабатываемый программно-технический комплекс реализует процесс конфигурирования и ориентирован на достижение следующих целей:

- построение программного средства для решения задач конфигурирования существующих сложных информационных систем и коммуникационных сетей;
- демонстрация процесса конфигурирования и работы отдельных его функций;
- получение экспериментальных данных для сравнения эффективности предлагаемого подхода с другими существующими подходами и реализованными программно-техническими решениями;
- применение полученных экспериментальных данных в дальнейших исследованиях и программных прототипах.

Средство включает следующие основные функции:

- функцию конфигурирования, которая согласно заданным ограничениям и списку заданных компонентов защиты выдает оптимальную конфигурацию;
- функции проверки эффективности выбранной конфигурации и сравнения эффективности двух заданных конфигураций.

Построение архитектуры программно-технического комплекса производится с использованием универсального языка моделирования *UML* [*9*]. Статическая модель механизма конфигурирования определяет его структурные элементы и связи между ними и представляется в виде «диаграммы классов» *UML* (см. рис. 1). Используемые элементы представления: Классы (сущности), их Атрибуты и Операции, Ассоциации и их кратность, Обобщения (наследование) выделены в соответствии нотацией UML 2.0 и согласуются с принципами объектно-ориентированного проектирования и анализа. На диаграмме показаны элементы модели, отвечающие за представление защищаемого устройства и его свойств; компонентов защиты; классификации свойств, присущих устройству и отдельным компонентам защиты; представление критериев оптимальности и функций конфигурирования и проверки допустимости конфигураций.

Для универсального хранения описания используемых механизмов компонентов защиты предполагается их формальная спецификация с использованием языка XML (eXtensible Markup Language). Преимуществом такого представления будет возможность реализации автоматизированных процедур получения данных о сторонних компо-

нентах защиты из открытых источников включая публичные базы данных.

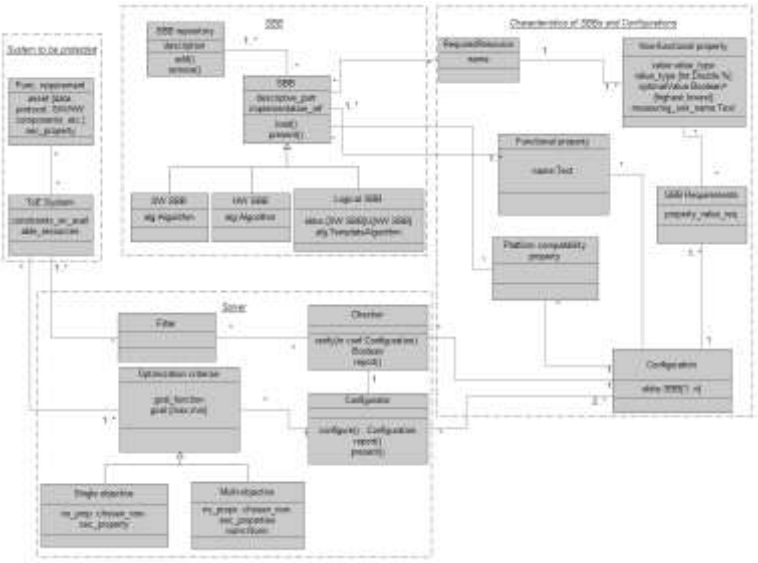

Рис. 1. Статическая модель механизма конфигурирования.

В таблице 2 приведен пример определения нефункционального свойства расходования ресурса оперативной памяти в килобайтах на выполнение функций для компонента защиты.

Таблица 2. **Пример описания нефункциональных свойств компонента защиты**

Компонент защиты	«Security Component 1»
Ресурс	«Memory”
Нефункциональное свойство	«consumed memory»
Имя свойства	«consumed memory amount»
Границы значений	промежуток 25−30
Тип значений	Integer
Оптимальное значение	Lowest
Единицы измерения	«KB»

Динамическая модель архитектуры описывает поведение механизма конфигурирования и сценарии его функционирования и пред-

ставляется при помощи *диаграмм последовательности* и *диаграммы активностей* в нотации UML 2.0.

На рис. 2 и рис. 3 приведены диаграммы последовательности, которые нацелены на представление взаимодействия объектов в рамках механизма конфигурирования.

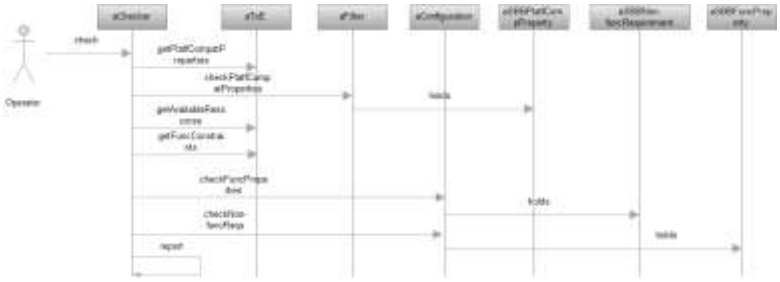

Рис. 2. Динамическая модель механизма конфигурирования, функция проверка допустимости конфигурации.

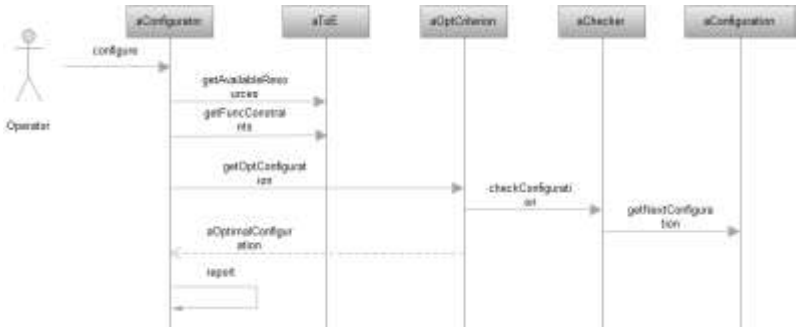

Рис. 3. Динамическая модель механизма конфигурирования, функция конфигурирования.

Используемые элементы представления: участники взаимодействия − объекты конфигуратора (экземпляры соответствующих классов) с их линией жизни; сообщения между объектами (посредством вызова метода объекта получателя), используются только синхронные сообщения; начальное (найденное) сообщение от оператора.

На рис. 4 показана *диаграмма деятельности*, которая нацелена на представление сценариев работы механизма. Приведенные на диаграмме *деятельности* агрегируют следующие операции:

- *Анализ защищаемого устройства* (Analysis of ToE) включает получение численных показателей расходуемых ресурсов устройства, а также получение информации о функциональных свойствах защиты;
- *Загрузка компонентов защиты* (Load SBBs) представляет получение из репозитория данных о доступных компонентах защиты;
- *Анализ доступных компонентов защиты (*Analysis of available SBBs) включает действия по получению данных о компонентах защиты, отсутствующих в спецификации (процедура получение внешних данных);
- *Проверка конфигурации (*Check configuration) представляет проверку допустимости заданной конфигурации путем сопоставления соответствующих свойств для конфигурации и защищаемого устройства;
- *Оптимизация (*Optimization) включает формирование критерия оптимальности, построение целевой функции и ограничений, а также перебор возможных комбинаций компонентов защиты и проверку их допустимости.

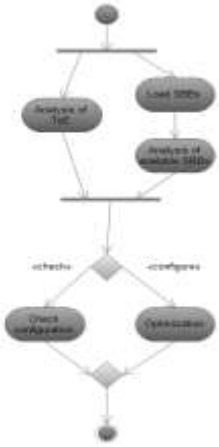

Рис. 4. Динамическая модель механизма конфигурирования, сценарий конфигурирования.

На рис. 5 приведен фрагмент пользовательского интерфейса механизма конфигурирования, реализованного в рамках платформы *Java 2*. Показано главное окно механизма, где оператору предоставляется информация о функциональных и нефункциональных свойствах компонентов защиты, критерии оптимизации, спецификация защищаемого устройства, а также элементы управления функциями механизма.

Рис. 5. Фрагмент пользовательского интерфейса прототипа механизма конфигурирования.

6. Оценка эффективности конфигурирования. Эффективность стратегии конфигурирования на основе решения оптимизационной задачи определяется путем ее сравнения с альтернативными стратегиями конфигурирования, в частности стратегией «произвольного конфигурирования». Стратегия произвольного конфигурирования представляет сценарии действий оператора системы, проводящего конфигурирование вручную, без использования автоматизированных средств анализа имеющихся в наличие компонентов защиты и поиска эффективных конфигураций. Введем следующие обозначения: P_{opt} – страте-

гия оптимальности на основе решения оптимизационной задачи; P_{rand} – стратегия «произвольного конфигурирования». Сравнение стратегий P_{opt} и P_{rand} конфигурирования производится путем эксперимента, включающего выбор условий конфигурирования, выполнение конфигурирования согласно P_{opt} и P_{rand}, и оценки и сравнения результатов.

В общем случае эффективность стратегии конфигурирования определяется путем ее оценки по некоторому заранее определенному критерию эффективности, а также на основании ее сравнения с альтернативными стратегиями. В контексте задачи конфигурирования для определения эффективности предложенной стратегии выясняется, обе ли стратегии при одинаковых условиях выдают допустимые конфигурации. Совпадают ли эти множества?

Определяются два типа критериев эффективности: критерий оптимальности на основе решения оптимизационной задачи; критерий на основе каких-либо внешних параметров устройства, механизма конфигурирования или отдельных компонентов защиты. В первом случае, стратегия P_{opt} будет наиболее эффективной по построению.

6.1. Стратегия «произвольного конфигурирования». Стратегия «произвольного конфигурирования» предполагает выполнение следующих действий.

Требуется выстроить в некотором порядке функциональные свойства устройства, которые нужно обеспечить. Порядок может быть произвольным.

Далее циклично, на каждой итерации оператор рассматривает очередное функциональное требование и выбирает некоторый компонент защиты, его реализующий, а также проверяет суммарные значения нефункциональных свойств по данному компоненту и всем компонентам защиты, взятым ранее, на предмет выполнения нефункциональных ограничений устройства. Компонент выбирается либо произвольно, либо как локально оптимальный компонент защиты.

Если необходимого компонента, удовлетворяющего этим условиям нет, то оператор выбирает компонент с «минимальным перерасходом» данного свойства и пытается пересмотреть компоненты, выбранные на предыдущих итерациях. Если не хватает ресурса сразу по двум или более нефункциональным свойствам, то для каждого такого свойства выбирается соответствующий компонент (с минимальным перерасходом данного свойства) и осуществляется пересмотр ранее выбранных блоков. Если в некоторый момент искомая конфигурация

была найдена, то в рамках данной итерации дальнейший поиск прекращается.

Пересмотр ранее выбранных компонентов защиты осуществляется следующим образом. Оператор пытается заменить один ранее выбранный компонент каким-нибудь другим с меньшими затратами по данному свойству и заменяет его. Причем, если изымаемый компонент реализовывал так же какое-либо функциональное свойство (которое теперь оказывается нереализованным), то это свойство добавляется в список еще нерассмотренных функциональных свойств, которые нужно реализовать для защиты устройства, и будет рассматриваться в рамках одной из последующих итераций. Если в какой-то момент происходит перерасход по какому-либо другому свойству, то такой компонент не должен быть выбранным. Если не смогли высвободить достаточного количества ресурса, то считаем, что допустимых конфигураций нет.

6.2. Сравнение стратегий. Стратегия P_{rand} определяет множество сходных сценариев действий оператора, отличающихся порядком рассмотрения функциональных свойств устройства.

Для каждого порядка последовательности функциональных свойств определяется, позволяет ли P_{rand} получить допустимую конфигурацию. Проводится выяснение того, дает ли каждый конкретный сценарий применения данной стратегии хотя бы какое-нибудь решение. Суммируется статистика по всем сценариям применения данной стратегии и вычисляется процент сценариев, которые дают допустимую конфигурацию. Определяется вероятность того, что стратегия P_{rand} будет выдавать допустимую конфигурацию. Если допустимая конфигурация существует, стратегия P_{opt} позволяет ее получить с вероятностью 100%.

Для каждого сценария P_{rand} определяется, дает ли P_{rand} оптимальную конфигурацию на основе рассматриваемого критерия эффективности или в сравнении с P_{opt}, в частности. Аналогично определяется вероятность получения оптимальной конфигурации на основе стратегии P_{rand}. При этом стратегии P_{opt} являются оптимальными по построению.

Для стратегий на основе веществennoзначного критерия эффективности для каждого сценария стратегии P_{rand}, дающего допустимые конфигурации, вычисляется значение критерия эффективности, кото-

рый сравнивается в процентном отношении со значением критерия для P_{opt}. Другими словами, определяется насколько каждый сценарий P_{rand} хуже P_{opt} при допущении, что стратегия P_{opt} оптимальна по построению. Вычисляется математическое ожидание E значений критерия эффективности для P_{rand} по всем сценариям, которые дают допустимые конфигурации (вариант 1), а также по всем сценариям (вариант 2) и E сравнивается с оптимальным значением. Вариант 1 дает возможность оценить эффективность стратегии в случае ее успешного применения (т.е. когда она дает решение), тогда как вариант 2 позволяет получить усредненную характеристику стратегии по всем ее сценариям, включая те, которые не дают допустимых конфигураций. Вычисляется также значение дисперсии, которая позволяет определить усредненное отклонение стратегии P_{rand} от ее математического ожидания. В целом при разработке эффективной стратегии конфигурирования математическое ожидание по всем ее сценариям должно быть близким к значению показателя эффективности при минимальной дисперсии.

7. Заключение. В работе предложена модель конфигурирования встроенных и мобильных устройств в рамках распределенной информационной системы на основе оптимизационной задачи. Модель нацелена на поддержку процесса защиты системы и предоставляет средства для поиска оптимальной конфигурации компонентов защиты для устройств системы. Особенностью модели является построение комбинированной защиты на основе отдельных компонентов защиты с учетом как функциональных свойств защиты компонентов, так и нефункциональных. Нефункциональные свойства компонентов защиты характеризуют объемы ресурсов, которые должны быть предоставлены устройством для корректной работы защитных функций.

Разработанная архитектура среды конфигурирования, а также программный прототип позволяют оператору системы или системному интегратору на этапе разработки и развертывания системы найти наиболее эффективную конфигурацию защиты. Программный прототип позволяет продемонстрировать принципы конфигурирования на основе оптимизационной задачи, и его ограничением является использование целочисленных показателей для нефункциональных свойств.

В качестве дальнейшей работы планируется: построение алгоритмов получения значений нефункциональных показателей и их практическая реализация в рамках рассмотренного варианта применения; проведение экспериментов по оценке эффективности конфигурирова-

ния на основе оптимизационной задачи и его сравнение с альтернативными стратегиями конфигурирования; решение задачи взаимосвязанного конфигурирования нескольких разнотипных встроенных и мобильных устройств с учетом связей между отдельными компонентами защиты на разных устройствах.

Литература

1. *Десницкий В.А., Котенко И.В., Чечулин А.А.* Построение и тестирование безопасных встроенных систем // Труды XII Санкт-Петербургской Международной конференции «Региональная информатика 2010» (РИ-2010), СПб.: СПОИСУ, 2011, С.115–121.
2. *Десницкий В.А., Чечулин А.А.* Модели процесса построения безопасных встроенных систем // Системы высокой доступности, № 2, 2011. С. 97–101.
3. *Котенко И.В., Десницкий В.А., Чечулин А.А.* Исследование технологии проектирования безопасных встроенных систем в проекте Европейского сообщества SecFutur // Защита информации. Инсайд, № 3, 2011. С.68–75.
4. *Atallah M., Bryant E., Stytz M.* A survey of Anti-Tamper Technologies // The Journal of Defence Software Engineering. Arxan Technologies, Inc. Nov. 2004.
5. *Aucsmith D.* Tamper-resistant software: An implementation // Information Hiding: First International Workshop: Proceedings, volume 1174 of Lecture Notes in Computer Science, pages 317–333. Springer-Verlag, 1996.
6. Ceccato M., Preda M., Nagra J., Collberg C, Tonella P. Trading-off security and performance in barrier slicing for remote software entrusting // Journal of Automated Software Engineering, Springer. 16(2): pp. 235–261, June 2009.
7. *Cormen T.H., Leiserson C.E., Rivest R.L., Stein C*. Introduction to Algorithms // Publisher: The MIT Press; third edition, 2009, ISBN-13: 978-0262033848.
8. Desnitsky V., Kotenko I., Chechulin A. An abstract model for embedded systems and intruders // Proceedings of the Work in Progress Session held in connection with the 19th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2011). Ayia Napa, Cyprus, February 2011. SEA-Publications. 2011.
9. *Fowler M*. UML Distilled: A Brief Guide to the Standard Object Modeling Language // Addison-Wesley Professional; 3 edition (September 25, 2003). Paperback: 208 pages ISBN-10: 0321193687.
10. *Grand J.* Practical Secure Hardware Design for Embedded Systems // Proceedings of the 2004 Embedded Systems Conference, San Francisco, California, April 1, 2004.
11. *Kennell R., Jamieson L.H*. Establishing the genuity of remote computer systems // Proc. of the 12th USENIX Security Symp. Washington, DC, 2003.
12. *Kocher P., Lee R., Mcgraw G., Ravi S.* Security as a new dimension in embedded system design // DAC '04 – Proceedings of the 41st Design Automation Conference. 2004.
13. *Koopman P.* Embedded System Security // IEEE Computer, July 2004.
14. *Lee G.M., Kim J.Y.* The Internet of Things – A problem statement // Information and Communication Technology Convergence (ICTC), 2010 International Conference on 17-19 November 2010, pp. 517–518, 978-1-4244-9806-2.
15. *Mastotakis N. E., Caraus I., Tkacenko A.* The DEA method in economical efficiency analysis (micro-level) // Proceeding MATH'07 of the 11th WSEAS International Conference on Applied Mathematics World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA, 2007 ISBN: 978-960-8457-60-7.

16. *Ovaska E., Balogh A., Campos S., Noguero A., Pataricza A., Tiensyrjä K.* Model and Quality Driven Embedded Systems Engineering // Technical Research Centre of Finland, 2009.
17. *Rae A. J., Wildman L.P.* A Taxonomy of Attacks on Secure Devices // Australian Information Warfare and IT Security, 20–21 November 2003, Australia, pp. 251–264, 2003.
18. *Ruiz J. F., Harjani R., Maña A., Desnitsky V., Kotenko I., Chechulin A*. A Methodology for the Analysis and Modeling of Security Threats and Attacks for Systems of Embedded Components // The 20th Euromicro International Conference on Parallel, Distributed and Network-Based Computing (PDP2012). Munich, Germany, February 15-17, 2012. Статья принята на конференцию.
19. *Seshadri A., Luk M., Shi E. et al*. Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems // Proc. of the 20th ACM Symp. on Operating Systems Principles (SOSP '05). NY: ACM Press, 2005.
20. *Seshadri A., Perrig A., Doorn L.V., Khosla P.* SWATT: Software-based attestation for embedded devices // Proc. of the IEEE Symp. on Security and Privacy. 2004.
21. *Stamp M*. Digital Rights Management: The Technology Behind the Hype // Journal of Electronic Commerce Research, Vol. 4, Nr. 3 (2003), p. 102−112.

Десницкий Василий Алексеевич – мл. научный сотрудник лаборатории проблем компьютерной безопасности Учреждения Российской академии наук Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН). Область научных интересов: методы защиты программного обеспечения, защиты встроенных систем, политики безопасности. Число научных публикаций ─ 32. desnitsky@comsec.spb.ru, http://comsec.spb.ru/desnitsky; СПИИРАН, 14 линия В.О., 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450. Научный руководитель — И.В. Котенко.

Desnitsky Vasily Alekseevich – junior researcher, Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research field: software protection methods, embedded system security, security policies. The number of publications — 32. desnitsky@comsec.spb.ru, http://comsec.spb.ru/desnitsky; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812)328-2642, fax +7(812)328-4450. Scientific leader — I.V. Kotenko.

Научный руководитель: **Котенко Игорь Витальевич** – д-р тех.наук, проф.; заведующий лабораторией Проблем компьютерной безопасности Учреждения Российской академии наук С.-Петербургский институт информатики и автоматизации РАН (СПИИРАН). Область научных интересов: безопасность компьютерных сетей, искусственный интеллект, телекоммуникационные системы. Число научных публикаций ─ более 450. ivkote@comsec.spb.ru, http://comsec.spb.ru/kotenko; СПИИРАН, 14 линия В.О., 39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450.

Scientific leader: **Kotenko Igor Vitalievich** – prof. of Computer Science; head of Laboratory of Computer Security Problems, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). Research field: computer network security, artificial intelligence, telecommunication systems. The number of publications — more than 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328-2642, fax +7(812)328-4450.

Поддержка исследований. Работа выполнена при финансовой поддержке РФФИ (проект № 10-01-00826), программы фундаментальных исследований ОНИТ РАН (проект № 3.2), проектов Евросоюза SecFutur и MASSIF, а также в рамках других проектов.

Рекомендовано лабораторией Проблем компьютерной безопасности СПИИРАН, заведующий лабораторией д-р техн. наук, проф. Котенко И.В.
Статья поступила в редакцию 11.11.2011.

РЕФЕРАТ

Десницкий В.А. **Конфигурирование встроенных и мобильных устройств на основе решения оптимизационной задачи.**

Работа посвящена разработке и исследованию конфигурационной модели, предназначенной для построения комбинированных механизмов защиты информационных систем со встроенными и мобильными устройствами на основе решения оптимизационной задачи. Предлагаемая модель описывает процесс конфигурирования, включающий нахождение оптимального набора специализированных программных компонентов защиты, используемых для поддержки безопасности информационной системы.

Работа ориентирована на построение и практическое применение комбинированных механизмов защиты для широкого круга распределенных информационных систем со встроенными и мобильными устройствами. Такие системы характеризуются динамически изменяемой топологией сети и отличающимися типами коммуникаций между отдельными узлами, а также заранее не фиксированным кругом функционирующих агентов и задействованных устройств. Как составные элементы информационной системы встроенные и мобильные устройства обладают, как правило, достаточно слабыми вычислительными возможностями и поэтому характеризуются низкой производительностью. Таким образом, задача поддержки защиты таких систем требует принципиально новых путей решения, которые помимо реализации защиты системы от заданного вида угроз учитывали бы также нефункциональные («ресурсные») требования к механизму защиты.

Подход к нахождению оптимальной конфигурации компонентов защиты, предлагаемый в работе, базируется на получении серии численных нефункциональных показателей защиты, и − путем постановки и решения оптимизационной экстремальной задачи при ограничениях на значения этих показателей и заданной целевой функции − позволяет построить наиболее эффективную конфигурацию компонентов защиты для обеспечения безопасности информационной системы. На базе функциональных и нефункциональных свойств и численных показателей формируются критерии оптимальности, используемые, во-первых, для нахождения оптимальных конфигураций и, во-вторых, для оценки эффективности и сравнения различных стратегий конфигурирования, включая стратегию «произвольного конфигурирования».

Программный прототип механизма конфигурирования, разработанный в рамках среды «Java 2» с использованием принципов объектно-ориентированного проектирования и языка моделирования UML, демонстрирует особенности процесса конфигурирования, включая возможность задания определенного критерия оптимальности, а также функцию проверки эффективности выбранной конфигурации.

SUMMARY

Desnitsky V.A. **Configuring embedded and mobile devices on the basis of solving an optimization problem.**

The paper is devoted to design and investigation of a configuration model aimed at constructing combined protection mechanism for information systems containing embedded and mobile devices reasoning from solving an optimization problem. The model proposed describes the configuring process embracing finding optimal bundle of special software security building blocks used to support security of the information system.

The activities conducted are targeted on development and application of combined protection mechanisms for a wide range of distributed information systems with embedded and mobile devices. Such systems are characterized by their topology changing dynamically, varying types of communications between their particular nodes and a sphere of functioning agents and devices that cannot be determined beforehand. Commonly as information system composite elements are embedded and mobile devices are in quite weak computational capabilities they produce lower performance. Therefore, sound protection for such systems demands materially novel solutions, besides implementation of protection against determined threats taking into account non-functional (resource based) requirements to the protection mechanism.

The proposed approach to finding optimal configuration of software building blocks is based on obtaining a series of numerical non-functional metrics and through settling and solving optimization problem with constraints on these metrics values and a specific objective function allows us to construct the most effective configurations to provide the system with firm security.

Starting from functional and non-functional properties and numerical metrics optimality criteria are formed and used firstly to find optimal configurations and secondly to evaluate effectiveness and comparison of various configuring strategies, including a strategy of “random configuring”. A software prototype of the configuration mechanism developed within Java 2 platform applying principles of object-oriented design and modeling language UML demonstrates peculiarities of the configuring process, and capabilities of optimality criterion determination and a function of checking effectiveness of a selected configuration in particular.