КОМПЛЕКСНАЯ СИСТЕМА ЗАЩИТЫ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ, УПРАВЛЯЕМЫХ МЕТЕДАННЫМИ

Денис Курилов, Людмила Лядова

Abstract: Предлагается описание архитектуры и подходов к реализации многоуровневой комплексной системы защиты динамически настраиваемых распределенных информационных систем, основанных на использовании метаданных. Исследуются возможности различных механизмов защиты. Описанная в работе система защиты представляет собой многоуровневый комплекс, построенный на базе мультиагентной системы, объединяющей функциональные возможности современных систем обнаружения вторжений (IDS – intrusion detection systems) с механизмами защиты структуры и программной логики информационных систем.

Keywords: адаптируемые информационные системы, механизмы защиты, метаданные, мультиагентная система.

ACM Classification Keywords: D.2 Software Engineering: D.2.0 General – Protection mechanisms; K.6 Management of Computing and Information Systems: K.6.5 Security and Protection – Authentication, Insurance, Invasive software (e.g., viruses, worms, Trojan horses), Unauthorized access (e.g., hacking, phreaking); I.2 Artificial Intelligence: I.2.11 Distributed Artificial Intelligence – Multiagent systems.

Введение

Характерными особенностями современных информационных систем (ИС), разрабатываемых для различных предметных областей, влияющими на степень их защищенности, надежности функционирования, являются [1]:

- *Сложность*. С возрастанием сложности растёт количество уязвимостей, обнаружение и устранение которых затруднено.
- *Открытость и интегрируемость*. Открытость информационных систем и их интегрируемость, реализация различных механизмов взаимодействия с внешними ИС является потенциальным источником уязвимостей.
- *Адаптируемость и расширяемость*. Возможность гибкой настройки ИС на конкретные условия работы и потребности пользователей, расширения функциональности сторонними разработчиками также создаёт опасность внедрения вредоносного кода.
- *Распределённость*. Возможность взаимодействия подсистем ИС через сеть создаёт дополнительные угрозы безопасности, такие как атака на серверные компоненты ИС с использованием клиентских компонентов.

Существующие на сегодняшний день методы защиты не позволяют защитить динамически настраиваемые информационные системы, функционирующие в распределенной среде, в комплексе: они либо защищают программный код, либо данные ИС. Возможности настройки ИС основаны на использовании средств динамической реструктуризации баз данных (БД); автоматической генерации и настройки пользовательского интерфейса; средств генерации запросов и отчетов; средств управления бизнес-процессами; средств подключения программных компонентов, созданных сторонними разработчиками. Все это делает еще более важной проблему защиты ИС, их ресурсов и программного обеспечения (ПО) от несанкционированного доступа и распространения, так как, используя предоставленные в их распоряжение средства, недобросовестные пользователи, обладающие

достаточной квалификацией, фактически могут использовать возможности технологий создания адаптируемых систем в своих целях.

Предлагаемый в данной статье подход рассматривает программное обеспечение ИС, функционирующее в распределенной среде, как цельный, неделимый программный продукт, который необходимо защищать именно в комплексе с данными и метаданными, описывающими ИС. Под комплексной защитой понимается защита данных и программного кода ИС и выбор наилучшей схемы лицензирования.

Устоявшийся взгляд на ИС как на сложные программные комплексы, компоненты которых установлены в узлах сети и взаимодействуют посредством передачи информации через линии связи, породил соответствующий подход к организации защиты ИС. Основа этого подхода – в реализации *различных механизмов защиты узлов сети от несанкционированного доступа и использования их ресурсов* (в частности, от вредоносного программного обеспечения, включающего различные вирусы и троянские программы), а также *защиты каналов взаимодействия в сети* при помощи различных технических и программных средств (например, экранирования, анализа сетевого трафика и т.п.). Такой подход к организации защиты вполне применим и оправдан, если речь идёт о защите программных систем, исполняемых на отдельных компьютерах или в рамках небольшой сети. В случае же распределенных ИС, масштаб которых выходит за рамки отдельного компьютера или небольшой локальной сети, проявляются существенные недостатки традиционного подхода, среди которых можно выделить следующие:

- *Сложность поддержания защиты ИС на должном уровне*. В силу ориентации подавляющего большинства коммерческих средств поддержки информационной безопасности (таких как Symantec Intruder Alert и NetProwler, семейство систем ISS RealSecure и др.) на сигнатурный метод выявления атак, требуется их постоянное обновление на всех узлах ИС. Предлагаемый в данной работе подход к организации системы защиты не требует отказа от средств такого рода, но значительно снижает зависимость от них.
- *Существенная уязвимость к новым типам атак*, например, основанных на выявлении и использовании ранее не применявшихся для этих целей уязвимостей самой ИС, операционной системы и сетевого программного обеспечения. Причина этого – опять же в преобладании сигнатурных методов анализа на современном рынке систем поддержки информационной безопасности.
- *Практически полное отсутствие защиты от атак, разработанных специально для взлома данной ИС*, основанных, в частности, на уязвимостях и ошибках в программной реализации её модулей. Как следствие – необходимость в дополнительных средствах защиты от атак, проводимых «изнутри» ИС, с использованием предварительно взломанных модулей.
- *Необходимость в дополнительных средствах контроля входящего и исходящего потоков информации*. Под контролем входящего потока подразумевается защита от спама, фишинга, вредоносных ad-ware и других аналогичных внешних угроз. Под контролем исходящего потока понимается сканирование всей исходящей информации, передаваемой во внешние системы, на предмет нахождения в ней защищенной корпоративной информации.
- *Сложность обеспечения достаточного уровня аутентификации пользователей*. В крупных ИС, содержащих защищённые данные и сервисы, стандартный механизм аутентификация пользователей, основанный на проверке знания некоторого секретного ключа, является малоэффективным в виду возможности утечки информации.

Причина многих недостатков традиционного подхода к обеспечению безопасности при применении его к защите распределённых ИС заключается в том, что подавляющее большинство средств защиты *не использует информацию о структуре и семантике защищаемой ИС*.

Архитектура системы защиты

В данной работе предлагается принципиально иной взгляд на построение систем защиты. Информационная система рассматривается не как совокупность взаимодействующих при выполнении своих функций вычислительных узлов, а как *совокупность сервисов*, предоставляемых компонентами ИС, реализуемых на базе нескольких взаимосвязанных узлов сети, требующая защиты в целом, а не на уровне отдельных структурных единиц и каналов связи. Оправданность такого подхода особенно проявляется на фоне тенденции расширения функциональных возможностей крупных ИС, имеющей целью обеспечить всю необходимую для повседневной работы пользователей функциональность в рамках одной интегрированной ИС (например, выполнение типовых операций по вводу и редактированию данных в рамках бизнес-процессов, автоматизируемых компонентами ИС, отправка и получение писем, обработка данных и генерация отчетов и др.).

Авторами предлагается подход к проектированию *комплексной системы защиты динамически адаптируемых распределенных информационных систем*, *основанных на использовании метаданных*, описывающих все стороны функционирования ИС. Архитектура комплексной системы защиты включает несколько уровней и объединяет различные механизмы защиты, интегрированные с защищаемой информационной системой на основе использования метаданных, описывающих эту ИС.

Предлагаемая система защиты представляет собой многоуровневый комплекс, построенный на базе *мультиагентной системы* (МАС). Комплекс сочетает в себе функциональные возможности современных систем *обнаружения вторжения* (IDS – *intrusion detection systems*) с механизмами *защиты структуры и программной логики* ИС.

Основой системы защиты является *распределённая МАС*. Сообщество агентов системы является *закрытым* и *защищенным* от злоумышленного влияния извне как при помощи механизмов, реализуемых на уровне собственной безопасности, так и за счёт самого способа организации работы агентов. Каждый *агент является независимой сущностью, скрыто функционирующей в рамках защищаемой системы*. Информация об агентах не содержится нигде в системе за пределами сообщества агентов, оперирующих в системе и потоках, в которых они исполняются. Сокрытие осуществляется двумя основными способами: исполнение агентов в потоках, *скрытых на уровне ядра ОС при помощи драйвера защиты*, и исполнение агентов в потоках защищаемой системы при помощи *механизма переключения контекста потока*, активно используемого самой ОС [2]. Все агенты МАС делятся на два класса: *агенты-аналитики* и *агенты-сенсоры*. Агенты-аналитики – это интеллектуальные агенты, построенные на базе архитектуры InteRRaP, относящейся к классу многослойных архитектур с вертикальным делением на слои [4]. Каждый слой реализует определённый тип взаимодействия агента со средой (областью системы, в которой оперирует данный агент). Информация о текущем состоянии системы передаётся с низших слоёв на высшие, управление – с высших на низшие.

Структура агентов представлена тремя слоями:

- *Слой поведения* отвечает за реализацию реактивности, поведения в режиме реального времени. Поле ответственности данного слоя – *принятие решений в условиях шаблонных ситуаций*, примерами которых могут служить регистрация пользователя, установка соединения с удалённым узлом сети или попытка взлома известного типа, сигнатура (т.е. сценарное описание) которого уже содержится в системе.
- *Слой планирования (локального планирования)* – это реализация *когнитивной парадигмы* построения МАС. При передаче информации со слоя поведения на слой планирования производится вывод по базе знаний агента, целью которого является определение класса текущей ситуации и выбор адекватного шаблона поведения (набора реакций на изменения состояний среды) с целью дальнейшего его применения на слое поведения.
- *Слой коммуникации (коллективного планирования)* – отвечает за реализацию *механизмов коммуникации агентов*. Данный слой представляет реализацию возможности принятия решения на

основе данных, подготовленных другими агентами системы, отвечает за организацию командной работы.

Представление знаний, используемых агентами для определения ситуации, осуществляется в рамках *фреймовой* парадигмы [3], правила принятия агентами решений представляются *продукционно*.

Агенты-сенсоры служат для сбора данных о текущем состоянии системы защиты, самой ИС и её модулей, а так же сети, в рамках которой функционирует ИС. Данные агенты реализуются на основе реактивного типа архитектуры (Reactive architecture [4]), и служат для сбора статистических данных, регистрации событий и выявление аномалий на базовом уровне.

Взаимодействие агентов основано на модели «заказчик-подрядчик» (Contract Net [4]), предполагающей решение различных задач посредством направления их на выполнение наиболее подходящим для этого агентам. Такой выбор обусловлен наличием у данной модели ряда преимуществ, обеспечивающих наиболее полное её соответствие требованиям решаемой задачи:

- Наличие у каждого агента системы функциональности, позволяющей выполнять некоторые задачи без привлечения других агентов системы (высокая степень самостоятельности агентов).
- Малый промежуток времени между возникновением задачи и началом процесса её решения.
- Малая вероятность неверного решения задач в связи с их назначением наиболее компетентным агентам, содержащим всю необходимую функциональность для их решения.
- Низкие накладные расходы в связи с отсутствием необходимости постоянного анализа каждым агентом текущего состояния среды.
- Высокая эффективность организации контроля системы, следующая из возможности организации агентов в иерархические структуры.

Уровни и механизмы защиты

Защита структуры и программной логики ИС является необходимым элементом защиты, неоправданно игнорируемым современными системами поддержки информационной безопасности по уже указанной выше причине – отсутствия в них информации о семантике защищаемой ИС. *Защита структуры ИС* необходима для пресечения возможности подмены серверных и клиентских компонентов ИС специально подготовленными для проведения дальнейшего взлома программными модулями, а также для более эффективной организации защиты от несанкционированных подключений к сервисам ИС. *Защита программной логики* обеспечивает пресечение попыток несанкционированной модификации программного кода ИС, целью которых может быть внесение ошибок и создание потайных каналов (*back doors* [5]) для организации проведения атак.

Информация о семантике защищаемой ИС представляется при помощи *иерархической трехслойной модели*, в полной мере описывающей все важные с точки зрения организации защиты аспекты ИС.

Вся информация о функционировании ИС, ее предметной области [6] распределяется по трём слоям модели защиты системы $S = (Str, Ev, Msg)$, где:

- *Слой структур* Str содержит описание структуры распределенной ИС, включает информацию об узлах сети и доменах приложений (подсистемах ИС), каналах связи, по которым осуществляется взаимодействие подсистем. Слой структур в модели представляется *P-графом* (*графом с полюсами* [7]) $Str = (N, A)$, где N – множество вершин с полюсами, представляющих домены приложений, узлы сети; A – множество связывающих их дуг, представляющих каналы связи.
- *Слой событий* $Ev = \{T, E, Q, Init(Q), Init(E), Ch, Sch\}$, где T – множество моментов времени; E – конечное множество событий; Q – конечное множество состояний; $Init(Q)$: $T \to Q$ – отображение, задающее начальное состояние; $Init(E)$: $T \to E \times T$ – отображение, задающее начальное планирование событий; Ch: $E \times Q \times T \to Q$ – отображение, определяющее новое состояние, в

которое система переходит в результате совершения события; Sch: $E \times Q \times T \to E \times T$ отношение планирования, представляющее причинно-следственные связи между событиями. Слой событий представляет описание работы ИС во времени. Данный слой включает информацию о различных состояниях, в которых может находиться система, и событиях, вызывающих смену состояний. Представлением данного слоя в модели ИС является *ориентированный граф*, вершинам которого ставятся в соответствие *состояния* ИС, в которых может находиться система в различные моменты времени, а дуги представляют *события* (в том числе связанные с получением сообщений), вызывающие смену состояний. Такой набор свяжем с каждой вершиной структуры Str.

– *Слой сообщений* Msg содержит описание данных, которыми подсистемы ИС могут обмениваться между собой, и правила преобразования этих данных. Слой задается как доопределение слоя событий.

Система защиты также является многоуровневой и включает в себя следующие уровни:

– уровень основной защитной логики,

– уровень контроля привилегий,

– уровень собственной безопасности,

– уровень системной безопасности.

Многоуровневый подход к организации защиты позволяет, кроме всего прочего, осуществлять *независимое проектирование различных механизмов защиты*. В частности, появляется возможность реализовывать «высокоуровневую» защитную логику (например, функции проверки вводимых активационных данных), основываясь на предположении невозможности изменения злоумышленником программного кода, реализующего эти функции, т.к. защита этого кода осуществляется на другом уровне.

На уровне *основной защитной логики* реализуется основная функциональность, требуемая от систем обнаружения вторжений согласно стандарту ISO 15408: контроль сетевого трафика, мониторинг работы сервисов ИС, выявление аномальных активностей.

Основным механизмом данного уровня является *подсистема активного аудита*, реализующая статистический и сигнатурный подходы к выявлению и анализу активности, описанные в требованиях FAU_SAA «Анализ данных аудита безопасности» (*Security audit analysis*). Основная функция подсистемы активного аудита – выявление аномалий в работе ИС. В общем случае любая попытка взлома является аномалией, выявляемой на основе статистического анализа работы ИС за продолжительный промежуток времени. В первую очередь имеется в виду анализ работы различных сервисов, расположенных на серверных модулях ИС. В целях компенсации недостатков статистического подхода к анализу активностей, таких как сложность принятия решений в условиях отсутствия устоявшейся эмпирической базы фактов и сложность обнаружения атаки в случае постепенного планомерного изменения параметров активности в сторону характерных для атаки, в рамках подсистемы активного аудита применяется *сигнатурный метод* выявления злоумышленной активности, соответствующий требованиям FAU_SAA.4 «Сложная эвристика атаки» (*Complex attack heuristics*). Под *сигнатурой* в данном случае понимается определённая последовательность событий, характерная для попытки взлома системы. Эффективная реализация механизма активного аудита достигается за счёт использования возможностей *распределённой мультиагентной системы*, лежащей в основе системы защиты. Информация от *агентов-сенсоров* с различных узлов сети, служащей инфраструктурой ИС, стекается к *агентам-аналитикам*, отвечающим за её обработку и формирование вывода о текущем состоянии системы и потенциальных угрозах её безопасности. Анализ производится на основе описанной выше иерархической трёхслойной модели ИС, содержащей также и информацию о самой системе защиты.

Уровень *контроля привилегий* содержит функциональность, обеспечивающую поддержку контроля прав пользователей системы на основе хранимых профилей активности в соответствии с требованиями FAU_SAA.2 «Выявление аномальной активности, основанное на применении профилей» (*Profile based anomaly detection*). Как правило, целью любой атаки на крупные корпоративные ИС является получение

доступа к конфиденциальным данным или к защищённым сервисам, что, в конечном счете, подразумевает необходимость получения высокого уровня привилегий в атакуемой системе [5]. Основным механизмом данного уровня является *подсистема анализа активностей пользователей*.

В ИС выделяются *группы пользователей*, каждой из которых соответствует определённый *набор привилегий*. В ходе этапа настройки и тестирования ИС производится сбор статистических данных о *типах активностей*, присущих данным группам пользователей, и формируются *групповые модели*, представляемые *графами активностей*. В групповой модели содержится информация, характеризующая поведение состоящих в группе пользователей при входе в систему, при работе в системе и при выходе из неё. После завершения построения групповых моделей для каждого пользователя строится *индивидуальная модель*.

Модель поведения представляет собой ориентированный граф $G = \{V, A\}$, где $V = \{v_i\}$ – множество вершин, на котором определено отношение порядка по следующему правилу: элемент, включённый в множество V последним, имеет старший номер в нём; $A = \{a_{ij}\}$ – множество дуг графа G. Каждому элементу $a_{ij} \in A$ ставится в соответствие некоторый вес $w_{ij} \in W$, где W – множество допустимых весов дуг. Вершины $v_i \in V$ представляют значения контролируемых параметров. Дуги $a_{ij} \in A$ представляют семантические связи между значениями контролируемых параметров, характеризующие очерёдность добавления вершин, соответствующих значениям параметров, т.е. элементов $v_i \in V$, в граф G. Веса $w_{ij} \in W$, назначенные дугам $a_{ij} \in A$, задают семантические расстояния между значениями контролируемых параметров, соответствующими инцидентным этим дугам вершинам $v_i, v_J \in V$. Семантическое расстояние характеризует различие между значениями контролируемого параметра активности. Построенная модель позволяет контролировать соответствие параметров некоторым эталонным значениям, «накапливая» происходящие изменения для последующего анализа.

В основе моделей лежит анализ различных типов *параметров активности пользователей*:

- *Категориальные параметры*. Примерами категориальных параметров могут служить измененные файлы, записи в БД, используемые сервисы ИС, инициированные команды, типы ошибок и т.п. Анализ категориальных параметров активности носит *событийно-ориентированный* характер.
- *Числовые параметры*. К данному типу относятся любые параметры активности, значения которых можно оценить количественно, например, объём переданной и запрошенной информации, количество сервисов, используемых единовременно, а так же количество вершин и дуг в модели.
- *Параметры интенсивности,* например, количество входов пользователя в систему за фиксированный промежуток времени, интенсивность запросов к БД, и т.п.
- *Параметры распределения событий*. К этому типу можно отнести, например, соотношения частоты таких событий как запрос на просмотр и запрос на изменение, обращёний к определённым сервисам ИС.

Основное применение моделей заключается в реализации *механизма аутентификации*, основанного на сопоставлении текущего поведения пользователя со статистическими сведениями об обычных параметрах его активности. Данный механизм является *дополнением стандартных механизмов аутентификации* и служит для защиты от несанкционированного получения привилегий путём кражи идентификационной информации привилегированных легальных пользователей.

Индивидуальные модели пользователей и групповые модели в динамически настраиваемых ИС могут быть использованы и для целей, не связанных с защитой, таких, например, как автоматическая генерация и настройка пользовательского интерфейса на основе статистической информации о применяемой данным пользователем или группой пользователей функциональности ИС.

Рассмотренные выше уровни защиты проектируются на основании предположения о невозможности модификации злоумышленником лежащего в его основе программного кода. На уровне *собственной*

безопасности реализуются механизмы, при помощи которых осуществляется защита программного кода ИС от анализа и изменения.

Основные механизмы данного уровня:

- Механизм *явного контроля целостности программного кода*, инициирует мгновенную реакцию системы защиты. В рамках данного механизма реализуются проверки программного кода приложения на предмет присутствия в нем несанкционированных изменений, а так же криптографические средства защиты программных модулей.
- Механизм *неявного контроля* используется для организации отложенной реакции системы на факт взлома с целью предотвращения возможности использования взломанного приложения. При обнаружении факта внесения злоумышленником изменений в программный код или деактивации им механизмов защиты первых уровней и/или механизма явного контроля система защиты переводится в имитирующий режим. При этом отсутствуют какие-либо внешние проявления обнаружения попытки взлома, но модули ИС, подвергшиеся злоумышленному воздействию, фактически изолируются в том смысле, что предотвращается возможность обращения с их помощью к ключевым данным и сервисам ИС.
- Механизм *сокрытия местонахождения функций системы защиты*. Данный механизм направлен в первую очередь на функции, отвечающие за обратную связь с пользователями ИС и, в частности, вывод сообщений об ограничении доступа блокировке защищённых сервисов в случае обнаружении попыток взлома (так называемые nag screens). Функции обратной связи, генерирующие такого вида сообщения, в большинстве случаев являются наиболее удобной отправной точкой для взлома системы [8]. Сокрытие производится путём вынесения всех потенциально опасных с точки зрения угрозы их обнаружения функций в динамически генерируемые программные модули. При этом функции, генерирующие «опасные» сообщения, ни в каком виде не хранятся в файлах приложения и их выявление и изменение становятся достаточно сложной задачей.

Уровень системной безопасности. Функционирование подавляющего большинства вредоносных программ невозможно без получения определённых привилегий, дающих возможность доступа к защищённым системным функциям. Доступ к системным функциям необходим для таких задач как открытие сетевых портов (например, для взаимодействия с внедрённым в атакуемую систему троянским модулем), исполнение программ в режиме отладки (в целях выявления брешей в защите), получение доступа к защищённым разделам внешней памяти и к адресным пространствам исполняемых программ, а так же к контроллерам ввода/вывода. Идеальным вариантом является получение возможности исполнения кода на нулевом уровне привилегий – это даёт возможность получения прямого доступа к любым ресурсам атакуемой системы, в том числе и к функциям ядра операционной системы и физическим устройствам. Защита уровня ядра служит для предотвращения возможности получения злоумышленником доступа к защищённым функциям операционной системы и, в частности, к ядру операционной системы.

Основными механизмами данного уровня являются: механизм выявления скрытых процессов, механизм контроля сетевого взаимодействия, механизм защиты нулевого уровня привилегий.

Механизм *контроля сетевого взаимодействия* служит для анализа состояния сетевых портов с целью выявления несанкционированных попыток открытия новых и изменения режима работы активных портов.

Данный механизм реализуется путём отслеживания вызовов соответствующих функций ядра ОС (в случае ОС Windows это Native API [8]), осуществляемого путём установки на данные функции оболочек, реализующих интерфейсы обратного вызова. Отслеживание вызовов функций ядра ОС является достаточным условием обнаружения несанкционированных попыток получения доступа к потенциально опасным с точки зрения организации безопасной работы функциям ОС, т.к. вызов любой функции прикладных программных интерфейсов, в конечном счёте, приводит к вызову некоторой функции ядра ОС. Кроме того, в большинстве случаев одной функции ядра ОС соответствует несколько различных

функций прикладных программных интерфейсов, являющихся, по сути, оболочками данной функции, осуществляющими её вызов с некоторым конкретным набором параметров [8], и только контроль на уровне ядра может гарантировать защиту, не зависящую от возможности появления новых программных интерфейсов и новых способов получения доступа к потенциально опасным функциям ОС.

Механизм защиты нулевого уровня привилегий обеспечивает защиту функций, выполняемых на нулевом уровне привилегий системы. Наибольшую угрозу безопасности представляют так называемые наборы средств для взлома [5] – руткиты (от англ. *rootkit*), работающие на нулевом уровне привилегий. Это вредоносное программное обеспечение, которое предоставляет злоумышленнику практически полный контроль над инфицированной системой и практически не поддающееся обнаружению и ликвидации. Возможны реализации руткита в виде отдельного драйвера или в виде оболочки некоторой функции ядра ОС. Предотвращение внедрения в защищаемую систему руткита осуществляется путём *контроля вызовов функций ядра ОС*, отвечающих за загрузку драйверов и образов в систему. Обнаружение руткитов, устанавливающих оболочки на функции ядра ОС, является технически несложной задачей, т.к. для обнаружения факта несанкционированной модификации достаточно иметь информацию об исходной структуре функций ядра. В рамках данного механизма осуществляются *периодические проверки соответствия значений хеш-функций, вычисленных от программного кода функций ядра ОС*, эталонным значениям, полученным при развертывании системы защиты и санкционированном внесении изменений в данные функции. Дополнительной мерой может служить отслеживание попыток получения доступа к памяти по адресам, соответствующим функциям ядра ОС, но это неминуемо приведёт к заметному снижению производительности защищаемой системы, и поэтому данная мера может применяться только в ситуациях, требующих обеспечения максимального уровня безопасности.

Механизм выявления скрытых процессов (невидимых на прикладном уровне) служит для обнаружения вредоносного ПО, работающего на прикладном уровне. Выявление скрытых процессов осуществляется при помощи работающего на системном уровне *драйвера защиты*.

Заключение

Основными преимуществами предлагаемой системы защиты являются:

- *Гибкость и возможность быстрой динамической адаптации* системы защиты к новым угрозам путём модификации баз знаний.
- *Универсальность*. Предлагаемая система защиты основывается на детализированной многоуровневой модели защищаемой ИС, что даёт потенциальную возможность интеграции в любую ИС.
- *Простота изменения и расширения функциональности*. Предлагаемая система защиты основана на работе со знаниями, и в большинстве случаев её функциональность можно расширить без внесения изменений в программный код.
- *Высокая производительность*. Метазнания, заложенные в системе защиты, дают возможность максимизировать эффективность её работы на основе данных анализа работы в рамках защищаемой ИС.
- *Возможность интеграции* системы защиты на поздних этапах разработки ИС за счет отсутствия необходимости внедрения защитной логики непосредственно в модули защищаемой ИС.

В настоящее время ведется разработка программных компонентов комплексной системы защиты на основе использования метаданных, управляющих функционированием информационных систем, построенных на базе технологии METAS, созданной сотрудниками АНО «Институт компьютинга».

Библиографический список

[1] Лядова Л.Н. Архитектура информационной системы «Образование Пермской области» // Математика программных систем: Межвузовский сборник научных трудов / Перм. ун-т. Пермь, 2002. С. 25-35.

[2] Кастер Х. Основы Windows NT и NTFS / Пер. с англ.— М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd.», 1996.

[3] Минский М. Фреймы для представления знаний. М.: Энергия, 1979.

[4] Huhns M., Stephens L. Multiagent Systems and Societies of Agents // Weiss G. Multiagent systems: a modern approach to a distributed artificial intelligence / Massachusetts Institute of Technology.

[5] Хогланд Г., Мак-Гроу Г. Взлом программного обеспечения: анализ и использование кода. М.: Вильямс, 2005.

[6] Лядова Л.Н., Мороз А.А. Модель защиты программного обеспечения от несанкционированного распространения // В кн.: Сборник трудов Второй международной научно-технической конференции «Инфокоммуникационные технологии в науке, производстве и образовании» (Инфоком 2) / Кисловодск, 2006. С. 120-124

[7] Миков А.И. Автоматизация синтеза микропроцессорных управляющих систем. Иркутск: Изд-во Иркут. ун-та, 1987.

[8] Касперски К. Техника и философия хакерских атак. М.: СОЛОН-Пресс, 2004.

Сведения об авторах

Денис Курилов – студент кафедры математического обеспечения вычислительных систем Пермского государственного университета; Россия, г. Пермь, 614990, ул. Букирева, 15; e-mail: Denis.Kurilov@mail.ru

Людмила Лядова – заведующий кафедрой математического обеспечения вычислительных систем Пермского государственного университета; Россия, 614990, ул. Букирева, 15; e-mail: LNLyadova@mail.ru