РАЗДЕЛ II
УПРАВЛЕНИЕ КИБЕРРИСКАМИ
И КИБЕРБЕЗОПАСНОСТЬЮ

Интеллектуальные механизмы управления кибербезопасностью

И. В. Котенко

Кибербезопасность в условиях глобальной информатизации общества рассматривается сегодня как одна из основных компонент национальной безопасности. Однако используемым в настоящее время подходам к обеспечению кибербезопасности присущ целый ряд недостатков. Эти недостатки обусловлены, главным образом, узкой специализацией отдельных средств киберзащиты и недостаточным уровнем их взаимодействия (кооперации), неразвитыми механизмами верификации защиты на этапах создания и поддержки, неадекватными механизмами определения уязвимостей, анализа рисков и определения уровня защищенности, мониторинга состояния сетей и адаптации к изменению условий их функционирования [1]. *В работе рассматривается подход к разработке и использованию систем киберзащиты (СКЗ), основанный на выделении интеллектуальной надстройки над традиционными механизмами защиты и построении единой унифицированной среды для создания и поддержки функционирования систем защиты. Представляются отдельные механизмы управления кибербезопасностью.*

1. Введение

В связи с беспрецедентно быстрым развитием компьютерных и телекоммуникационных технологий, в том числе появлением сети Интернет, объединяющей огромное количество разнородных сетей (от локаль-

ных до транснациональных), и переходом к информационному обществу проблема обеспечения кибербезопасности и построения информационно-безопасных распределенных вычислительных систем стала одной из наиболее актуальных проблем [1].

В соответствии с современными представлениями перспективная система киберзащиты (СКЗ) должна представлять собой взаимоувязанную, многоэшелонированную и непрерывно контролируемую систему, способную оперативно реагировать на удаленные и локальные кибератаки и несанкционированные действия (НСД), накапливать знания о способах противодействия, обнаружения и реагирования на атаки и НСД и использовать их для усиления защиты.

Такая СКЗ должна предоставлять, по крайней мере, три уровня защиты [2]. Первый уровень защиты составляют «традиционные» средства защиты, реализующие функции идентификации и аутентификации, криптографической защиты, разграничения доступа, контроля целостности, регистрации и учета, межсетевого экранирования. Второй уровень включает средства проактивной защиты, обеспечивающие сбор необходимой информации, анализ защищенности, мониторинг состояния сети, обнаружение атак, противодействие их реализации, введение злоумышленника в заблуждение и т. п. Третий уровень соответствует средствам управления защитой, которые осуществляют интегральную оценку состояния сети, управление защитой и адаптацию политик безопасности и компонентов СКЗ.

Первый уровень достаточно широко представлен в существующих исследованиях. Разработка механизмов киберзащиты, относящихся ко второму и особенно третьему уровню, реализующих по существу интеллектуальную надстройку над традиционными механизмами защиты (для управления ими), составляет в настоящее время приоритетную задачу в области теоретических и прикладных исследований по построению информационно-безопасных распределенных вычислительных систем.

В статье рассматривается подход к разработке и использованию СКЗ, основанный на выделении такой интеллектуальной надстройки над традиционными механизмами защиты и построении единой унифицированной среды для создания и поддержки функционирования систем киберзащиты.

2. Интеллектуализация механизмов защиты

В рамках решения задачи киберзащиты авторами исследуется комплекс формальных методов, моделей, алгоритмов и построенных на их основе программных прототипов, реализующих различные интеллектуальные механизмы защиты:

1) сбор информации о состоянии информационной системы и ее анализ за счет механизмов обработки и слияния информации из различных источников;
2) проактивное предупреждение атак и препятствование их выполнению;
3) обнаружение аномальной активности и явных атак, а также нелегитимных действий и отклонений работы пользователей от политики безопасности, предсказание намерений и возможных действий нарушителей;
4) активное реагирование на попытки реализации действий нарушителей путем автоматической реконфигурации компонентов защиты для отражения действий нарушителей в реальном масштабе времени;
5) дезинформацию злоумышленника, сокрытие и камуфляж важных ресурсов и процессов, «заманивание» злоумышленника на ложные (обманные) компоненты с целью раскрытия и уточнения его целей, рефлексивное управление поведением злоумышленника;
6) мониторинг функционирования сети и контроль корректности текущей политики безопасности и конфигурации сети;
7) поддержку принятия решений по управлению политиками безопасности, в том числе по адаптации к последующим вторжениям и усилению критических механизмов защиты.

Рассмотрим ниже ряд интеллектуальных механизмов киберзащиты, предложенных автором работы.

Механизмы управления кибербезопасностью, основанные на интеллектуальных агентах

Перспективным подходом к построению интеллектуальных механизмов киберзащиты является *технология интеллектуальных многоагентных систем*. Этот подход позволяет по сравнению с традиционными методами существенно повысить эффективность защиты информации, в том числе ее адекватность, отказоустойчивость, устойчивость к деструктивным действиям, универсальность, гибкость и т. д.

В соответствии с данным подходом предполагается, что компоненты систем киберзащиты, специализированные по типам решаемых задач, тесно взаимодействуют друг с другом с целью обмена информацией и принятия согласованных решений, адаптируются к изменению трафика, реконфигурации аппаратного и программного обеспечения, новым видам кибератак [3–6].

В рамках предлагаемого подхода компоненты многоагентной системы киберзащиты представляют собой интеллектуальные автономные программы (агенты защиты), реализующие определенные функции защиты с целью обеспечения требуемого класса защищенности. Они позволяют реализовать комплексную надстройку над механизмами безопасности используемых сетевых программных средств, операционных систем и приложений, повышая защищенность системы до требуемого уровня.

В рамках данного направления исследований разработаны архитектуры, модели и программные прототипы нескольких многоагентных систем, в том числе агентно-ориентированная система моделирования атак, многоагентная система обнаружения вторжений, многоагентная система обучения обнаружению вторжений и др.

Согласно разработанной технологии процесс создания многоагентных систем для любой предметной области, в том числе киберзащиты, предполагает решение двух высокоуровневых задач [5, 7]:

1) создание «Системного ядра» многоагентной системы;
2) клонирование программных агентов и отделение сгенерированной многоагентной системы от «Системного ядра».

Для спецификации «Системного ядра» используются два компонента программного инструментария создания многоагентных систем MASDK («Multi-agent System Development Kit») [7]. Первый из них — это так называемый «Типовой агент» («Generic Agent»), который предназначен для создания высокоуровневой спецификации класса агента. Второй компонент служит для формирования проблемно-ориентированной архитектуры приложения, заполнения данных, знаний, а также определения коммуникационного компонента.

Сформированные агенты имеют аналогичную архитектуру (рис. 1). Различия отражаются в содержании данных и баз знаний агентов. Каждый агент взаимодействует с другими агентами, средой, которая воспринимается и, возможно, изменяется агентами, а также пользователем, общающимся с агентами через пользовательский интерфейс.

В предложенной формальной модели и прототипе *агентно-ориентированной системы моделирования атак* (АСМА) распределенные скоординированные атаки на компьютерную сеть рассматриваются в виде последовательности совместных действий агентов-хакеров, которые выполняются с различных хостов [4, 8]. Предполагается, что хакеры координируют свои действия согласно некоторому общему сценарию. На каждом шаге сценария атаки они пытаются реализовать некоторую

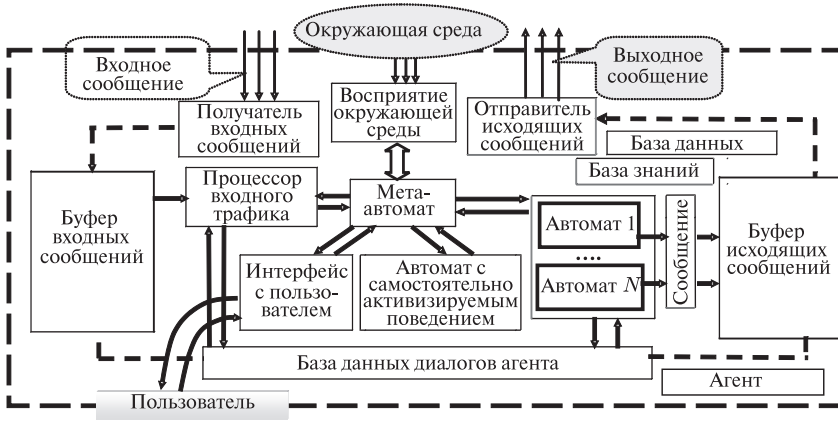

Рис. 1. Архитектура типового агента

частную подцель. АСМА построена на основе предложенной формальной модели реализации атак.

Отличительные черты реализованного в АСМА подхода к моделированию атак: моделирование атак базируется на спецификации задач хакеров и иерархии их намерений; многоуровневое описание атаки представляется в последовательности «общий сценарий распределенной атаки → намерения хакеров → простые атаки → входной трафик или данные аудита»; разработка планов действий хакеров и моделей отдельных атак основывается на задании онтологии предметной области «Атаки на компьютерные сети»; формальное описание сценариев взаимодействия агентов и реализации распределенных атак выполнено на базе семейства стохастических атрибутных грамматик, связанных операциями подстановки; в алгоритмической интерпретации процедур генерации атак каждой из грамматик ставится в соответствие автомат; генерация действий (атак) хакеров происходит в зависимости от реакции атакуемой сети в реальном масштабе времени.

Разработанный к настоящему времени программный прототип АСМА состоит из следующих компонентов (агентов): множества агентов хакеров, каждый из которых реализует модель атакующего, агента — модели атакуемой компьютерной сети и генератора фонового «нормального» трафика. В процессе атаки агенты обмениваются сообщениями с целью координации своих действий.

На рис. 2 зафиксирован процесс генерации одной из атак. На рисунке данные о реализуемой атаке разбиты на четыре группы:

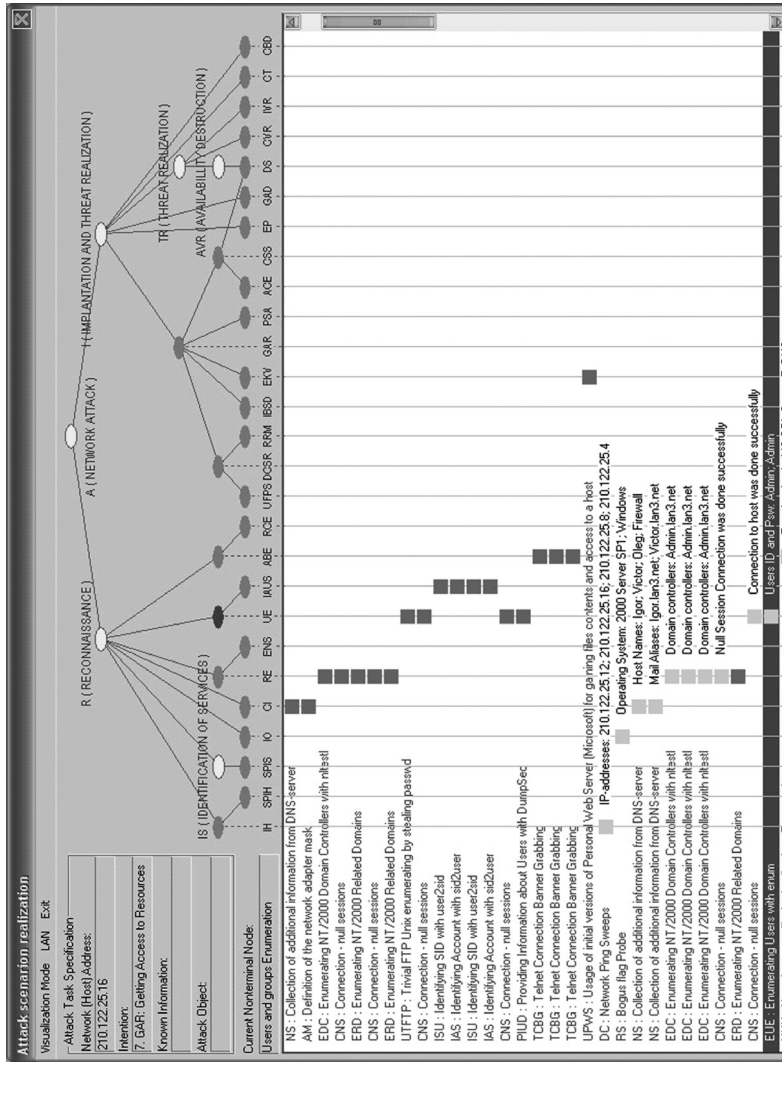

Рис. 2. Окно визуального представления развития сценария атаки

1) в левой верхней части экрана отображается элементы спецификации задачи атакующего;
2) справа от них визуализируется дерево генерации атаки;
3) в левой части экрана ниже данных о спецификации задачи размещаются строки генерируемых действий злоумышленника;
4) для каждого действия злоумышленника справа отображаются признак успеха (неуспеха) в виде квадрата зеленого (черного) цвета и данные, полученные от атакуемого хоста (реакция хоста).

Компоненты *многоагентной системы обнаружения вторжений* (МСОВ) — это взаимодействующие между собой агенты, совместно решающие общую задачу обнаружения вторжений в компьютерную сеть [3, 7, 9, 10]. Архитектура МСОВ включает один или несколько экземпляров агентов разных типов, специализированных для решения подзадачи обнаружения вторжений. Агенты распределены по хостам защищаемой сети, специализированы по типам решаемых задач и взаимодействуют друг с другом с целью обмена информацией и принятия согласованных решений. В принятой архитектуре исследуемого прототипа МСОВ в явном виде отсутствует «центр управления» семейством агентов — в зависимости от сложившейся ситуации ведущим может становиться любой из агентов, инициирующий и (или) реализующий функции кооперации и управления. В случае необходимости агенты могут как клонироваться (образовывать новые сущности), так и прекращать свое функционирование. В зависимости от ситуации (вида и количества атак на компьютерные сети, наличия вычислительных ресурсов для выполнения функций защиты) может потребоваться генерация нескольких экземпляров агентов каждого класса. Предполагается, что архитектура МСОВ может адаптироваться к реконфигурации сети, изменению трафика и новым видам атак, используя накопленный опыт.

Представляется, что наиболее действенный путь обнаружения распределенных многофазных атак, направленных на компьютерные сети, состоит в кооперации множества агентов защиты, распределенных по хостам сети. Поэтому основное достоинство МСОВ заключается в возможности относительно «легких» компонентов системы сотрудничать и совместно решать сложную задачу обнаружения таких атак. Базовые черты подхода, реализованного в МСОВ, таковы:

1) расширяемая и адаптивная многоагентная архитектура;
2) центральное внимание уделяется обнаружению многофазных распределенных атак;

3) обеспечение безопасности и робастности (обработка сетевых событий, важных с точки зрения защиты информации, и функции управления распределены среди множества агентов различных хостов).

Базовые типы компонентов разработанной МСОВ, размещаемые на каждом из хостов защищаемой компьютерной сети, представлены на рис. 3.

Агент-демон AD-E (*AD-Events*) осуществляет предварительную обработку поступающих на хост сообщений, фиксируя значимые для защиты информации события, и переадресует выделенные сообщения соответствующим специализированным агентам. *Агент-демон идентификации и аутентификации* AIA ответствен за идентификацию источников сообщений и подтверждение их подлинности. *Агент-демон разграничения доступа* ACA регламентирует доступ пользователей к ресурсам сети в соответствии с их правами и метками конфиденциальности объектов защиты. Агенты AIA и ACA обнаруживают несанкционированные действия по доступу к информационным ресурсам хоста, прерывают соединения и процессы обработки событий, отнесенные к числу несанкционированных, а также посылают сообщения агентам обнаружения вторжений. *Агенты-демоны* AD-P1 и AD-P2 (*AD-Patterns*) отвечают за обнаружение отдельных «подозрительных» событий или очевидных фактов вторжения и принятие решений относительно реакции на данные события (факты). *Интеллектуальные агенты обнаружения вторжений* IDA1 и IDA2 реализуют более высокий уровень обра-

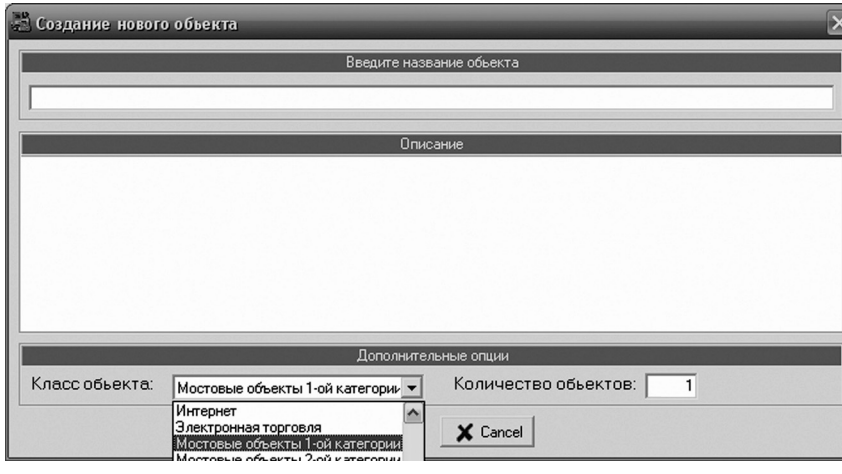

Рис. 3. Архитектура компонентов МСОВ на хосте

ботки и обобщения обнаруженных фактов. Они принимают решения на основе сообщений об обнаруженном подозрительном поведении и явных атаках как от агентов-демонов своего хоста, так и от агентов других хостов.

Возможными высокоуровневыми сценариями, обнаруживаемыми IDA2, являются:

1) разведка — разведывательные действия атакующего (действия по определению конфигурации сети, обнаружению хостов, функционирующих на хосте сервисов, определению операционной системы, приложений и т. п.);
2) внедрение в систему — действия злоумышленника по взлому хоста и внедрению в систему;
3) повышение прав — попытки атакующего, направленные на получение повышенных прав по доступу к объектам хоста;
4) распространение поражения на хосте — нелигитимное распространение злоумышленника по объектам хоста (каталогам, файлам, программам);
5) распространение поражения по сети — распространение атакующего по защищаемой компьютерной сети и др.

Многоагентная система обучения обнаружению вторжений в компьютерные сети (МСООВ) является мультисенсорной системой объединения данных. Она формирует решения на основе многоуровневой модели обработки входных данных (входного трафика сети и данных аудита). На нижнем уровне решения принимаются так называемыми «*базовыми*» классификаторами. Их может быть несколько для одного и того же подмножества атак, но они должны обучаться на различных наборах обучающих и тестовых данных. На более высоком уровне решения базовых классификаторов используются для принятия итогового решения на основе объединения решений базовых классификаторов. Это выполняется мета-классификаторами. Применительно к такому взгляду на обучаемую систему предложена архитектура многоагентной системы обучения обнаружению вторжений [5, 6]. Эта система имеет многоагентную архитектуру, реализующую многоуровневое обучение на основе имеющихся интерпретированных данных из тех же источников и представленных в тех же структурах, которые используются МСОВ. Типовыми классами агентов МСООВ являются: класс агентов управления данными обучения; класс агентов тестирования классификаторов; класс агентов подготовки мета-данных; класс обучающих агентов. В качестве методов (алгоритмов) обучения, которые позволяют ре-

шать рассматриваемую задачу обучения, используются методы ID3, C4.5, бустинг, мета-классификации, FP-growth, метод визуальной классификации, GK2, INFORM и др.

Исследование возможностей агентских технологий и проведенные эксперименты с разработанными программными прототипами показали несомненные преимущества применения интеллектуальных систем киберзащиты и использования многоагентного подхода к построению СКЗ. В пользу этого тезиса можно выдвинуть следующие обстоятельства [5].

- Распределение объектов и средств защиты как в границах хоста, так и в рамках компьютерной сети диктует необходимость использовать распределенную интегрированную систему защиты, к классу которых относятся многоагентные СКЗ.
- Большинство атак реализуются по предварительно заданным сценариям. Каждый сценарий состоит из последовательных стадий, предназначенных для преодоления различных уровней защиты. Сложные атаки на компьютерные сети могут затрагивать сразу несколько хостов сети и иметь целью поражение множества хостов. Они могут реализовываться посредством кооперации большой группы злоумышленников и использования множества хостов для инициирования отдельных фаз атаки из нескольких источников в сети. Реализованная в АСМА агентно-ориентированная технология позволяет адекватно моделировать распределенные скоординированные атаки. Кооперация распределенных агентов обнаружения вторжений может обеспечивать обнаружение атак, реализующих такие сложные сценарии.
- Многоагентный подход обеспечивает повышение оперативности выполнения задач защиты в силу распараллеливания и автоматического выполнения решаемых задач. В разработанном прототипе МСОВ агенты IDA1 и IDA2 осуществляют обобщенный анализ обнаруженных фактов вторжения в рамках всей защищаемой сети. Это позволяет использовать режим автоматического обнаружения сложных координированных атак и минимизировать количество ложных срабатываний и пропусков атак.
- Для больших распределенных систем крайне важна способность продолжать функционировать, когда ее компоненты разрушены или изолированы. СКЗ, имеющие централизованную архитектуру, могут легко поражаться злоумышленником, например, путем атаки «отказ в обслуживании» на хосты управления СКЗ. Как в АСМА, так и в МСОВ совокупность агентов, соответственно выполняю-

щих атаку или реализующих задачу обнаружения, на каждом из хостов может взять на себя необходимые функции генерации или обнаружения распределенной атаки.

- Исключительно важной является способность компонентов СКЗ отслеживать состояние среды функционирования и приспосабливаться к ее изменениям. В разработанных прототипах агенты могут клонироваться для охвата всех необходимых в текущей ситуации задач защиты, обеспечения требуемой избыточности и параллелизма, а также обращаться к агентам других хостов для оказания помощи. Агенты, обладающие указанными характеристиками, автономно и асинхронно выполняющие свои функции, позволяют сформировать робастную и отказоустойчивую систему защиты.
- Для повышения эффективности защиты различные подсистемы СКЗ должны взаимодействовать друг с другом на разных уровнях абстракции решений, сформированных каждой из них. Такой стиль функционирования и взаимодействия подсистем СКЗ, как показало исследование реализованных прототипов, определяет требуемый способ декомпозиции функций защиты и необходимые средства взаимодействия между подсистемами СКЗ. Данный подход естественным образом реализуется с использованием парадигмы многоагентной системы и позволяет препятствовать, обнаруживать и подавлять атаки на более ранних стадиях их развития.
- Многоагентная система может составлять многокомпонентную вычислительную среду, независимую от аппаратных и программных средств, на которых она базируется. Это позволяет реализовать мощную и настраиваемую среду для реализации различных механизмов защиты информации компьютерных сетях.

Технологии дезинформации злоумышленника, сокрытия и камуфляжа важных ресурсов и процессов, «заманивания» злоумышленника на ложные (обманные) компоненты

Для защиты информационных ресурсов компьютерных сетей необходимо не только предупреждать, блокировать, обнаруживать и реагировать на действия нарушителей, но и отвлекать их от основных целей, заманивая на ложные информационные объекты, производить сбор информации о приемах, тактике и мотивации злоумышленников, осуществлять их идентификацию и разоблачение. Для выполнения этих подзадач могут быть использованы так называемые ложные информационные

системы (ЛИС), называемые также системами-имитаторами, обманными системами или системами-ловушками [11, 12].

ЛИС представляют собой программно-аппаратные средства обеспечения информационной безопасности, реализующие функции сокрытия и камуфляжа защищаемых информационных ресурсов, а также дезинформации нарушителей [13].

На основании анализа работ в указанной области в качестве основных функций, которые должны быть реализованы в перспективных ЛИС, можно выделить следующие:

- захват данных («прослушивание» сетевого трафика и фиксация данных для последующего анализа);
- сбор и объединение данных от различных программных и аппаратных компонентов компьютерной сети, в частности сенсоров, межсетевых экранов, систем обнаружения вторжений, маршрутизаторов и др.;
- определение «свой—чужой» и переадресация несанкционированных запросов на ложные компоненты;
- фильтрация событий (для автоматической отбраковки несущественных и фокусировки на значимых событиях);
- обнаружение действий нарушителя;
- выявление источника угроз, трассировка и идентификация нарушителя (определение типа, квалификации и др.);
- обеспечение невозможности использования скомпрометированных компонентов (ресурсов) для атаки или для нанесения вреда другим системам после проникновения нарушителя в ЛИС;
- распознавание плана (стратегии) действий нарушителя; контроль действий нарушителя и реагирование на них, в том числе оповещение администратора о компрометации, блокирование действий нарушителя и др.;
- формирование плана действий компонентов ЛИС по имитации целевой информационной системы;
- заманивание и обман нарушителя (привлечение внимания, сокрытие реальной структуры защищаемой системы и ресурсов, камуфляж, дезинформация) за счет эмуляции сетевых сегментов, серверов, рабочих станций, в том числе передаваемого трафика, и их уязвимостей, автоматическое реагирование на действия нарушителя, в том числе оповещение администратора;

Рис. 4. Обобщенная функциональная структура ЛИС

- удаленное администрирование, документирование, ввод сигнатур, профилей и др. (обеспечивает централизованное управление, основанную на правилах безопасности реакцию системы, подготовку отчетов и анализ тенденций); обеспечение интерфейса с администратором безопасности.

Обобщенная функциональная структура разрабатываемой перспективной ЛИС представлена на рис. 4. Жирным шрифтом выделены базовые компоненты ЛИС.

В общем случае ЛИС может обеспечить три уровня введения в заблуждение нарушителя (рис. 5):

1) уровень сегмента (основных компонентов целевой системы) — на данном уровне ЛИС имитирует защищаемую целевую систему в целом, и при обнаружении атаки злоумышленник перенаправляется с целевой системы на компоненты ЛИС;
2) уровень хоста — данный уровень предполагает размещение компонентов ЛИС, имитирующих отдельные хосты, в компьютерной сети целевой системы;
3) уровень сервиса/приложения — в рамках хоста целевой системы каждое приложение/сервис формируется следующим образом: целевой модуль сервиса/приложения вместе с модулем обмана «вкладывается в обертку»; в режиме санкционированного использования при вызове сервиса/приложения управление передается целевому модулю; при обнаружении несанкционированного обращения управление передается модулю обмана.

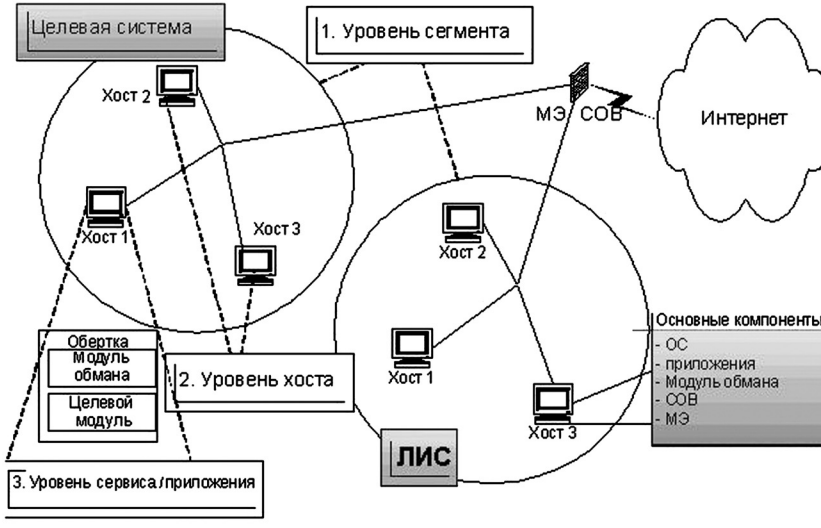

Рис. 5. Обобщенная архитектура ЛИС
и реализуемые уровни введения в заблуждение

Для исследования возможностей перспективных ЛИС разработаны их прототипы и ведутся эксперименты с различными компонентами ЛИС [12, 13].

3. Поддержка жизненного цикла системы киберзащиты

В процессе использования различных механизмов киберзащиты необходимо осуществлять поддержку защищенной информационной среды на различных этапах жизненного цикла, включая этапы их проектирования, конфигурирования, развертывания, функционирования и модификации. Поэтому, кроме создания отдельных перспективных механизмов защиты, необходимо решать задачу разработки моделей и методов построения единой унифицированной системы (среды), осуществляющей поддержку всего жизненного цикла СКЗ, включая адаптивное управление политиками безопасности [2].

В работе предлагается подход к осуществлению непрерывной цепочки различных этапов жизненного цикла распределенных защищенных компьютерных систем (с множеством прямых и обратных связей от одного этапа к другому) (рис. 6, рис. 7 [14]).

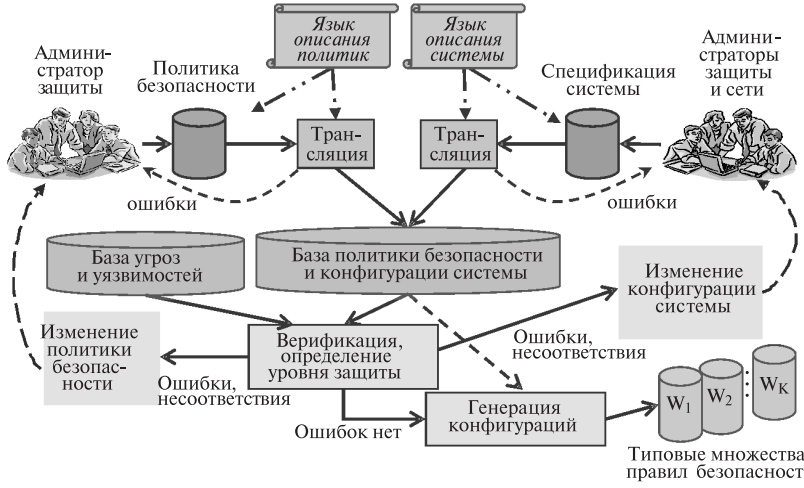

Рис. 6. Начальные этапы поддержки жизненного цикла системы киберзащиты (от спецификации до трансляции сформированных правил безопасности в типовые правила)

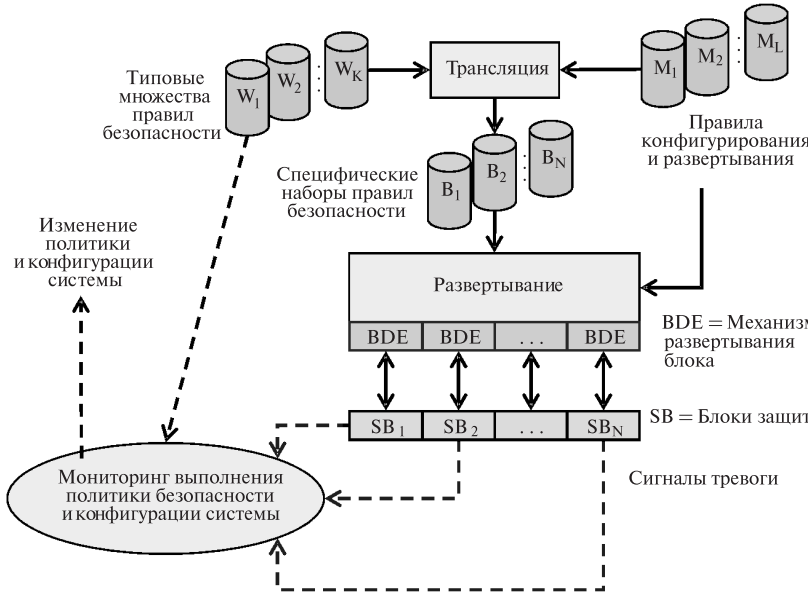

Рис. 7. Последующие этапы поддержки жизненного цикла системы киберзащиты (от трансляции правил безопасности в параметры конфигурации и настройки программно-аппаратного обеспечения до адаптации поведения)

Данный подход предполагает реализацию следующих механизмов:

1) спецификацию политик безопасности и архитектуры (или конфигурации) защищаемой системы;
2) трансформацию политик безопасности с целью их уточнения (детализации) с учетом описания защищаемой системы;
3) верификацию политик безопасности (проверку правильности и устранение конфликтов);
4) определение уровня безопасности и анализ рисков;
5) моделирование поведения системы защиты в различных условиях функционирования;
6) изменение политик в соответствии с требуемым уровнем безопасности и возможностями по использованию различных ресурсов и выделению финансовых средств и на защиту информации;
7) реализацию политик безопасности в системе, в том числе трансляции сформированных правил безопасности в параметры конфигурации и настройки программно-аппаратного обеспечения;
8) проактивный мониторинг выполнения политик безопасности, в том числе обнаружение отклонений работы пользователей от политики безопасности, обнаружение вторжений и анализ уязвимостей;
9) адаптацию поведения распределенных защищенных компьютерных систем и реализованных политик безопасности в соответствии с условиями функционирования.

Ниже рассмотрим некоторые из перечисленных выше механизмов создания и поддержки функционирования системы киберзащиты.

Механизмы определения уровня кибербезопасности

В настоящее время актуальной задачей в области кибербезопасности является обнаружение уязвимостей и оценка уровня защищенности киберсистем. Для решения данной задачи служит специальный класс систем, называемых системами анализа защищенности (САЗ). Современные САЗ предназначены для проверки защищаемой системы на соответствие заданной системной конфигурации и политике безопасности, определения уязвимостей для их дальнейшего устранения и уменьшения рисков, вызванных наличием данных уязвимостей.

Предлагаемый подход к построению САЗ на основе активных методов базируется на механизме автоматической генерации и выполнения распределенных сценариев атак с учетом разнообразия целей и уровня знаний злоумышленника [15, 16]. Рассматриваемый подход ба-

зируется на комплексном использовании основанных на экспертных знаниях моделей злоумышленника, вероятностных моделей компьютерной сети, генерации комплекса сценариев атак и оценки уровня защищенности.

Система анализа защищенности, использующая предложенный подход, предназначена для функционирования на различных этапах жизненного цикла компьютерной сети, включая этапы проектирования и эксплуатации. На этапе проектирования САЗ оперирует с моделью анализируемой компьютерной сети, которая базируется на заданной спецификации компьютерной сети и реализуемой политики безопасности. На этапе эксплуатации САЗ взаимодействует с реальной компьютерной сетью.

Результаты генерируемых атак позволяют определить уязвимости, построить трассы (графы) возможных атак, выявить «узкие места» в компьютерной сети, и вычислить различные метрики безопасности, которые могут быть использованы для оценки общего уровня защищенности компьютерной сети (системы), а также уровня защищенности ее компонентов.

Полученные результаты обеспечивают также выработку обоснованных рекомендаций по устранению выявленных узких мест и усилению защищенности системы. На основе данных рекомендаций пользователь САЗ вносит изменения в конфигурацию реальной сети или в ее модель, а затем, если необходимо, повторяет процесс анализа уязвимостей и оценки уровня защищенности. Таким образом, обеспечивается требуемый уровень защищенности компьютерной сети (системы) на всех этапах ее жизненного цикла.

Обобщенная архитектура предлагаемой системы активного анализа защищенности представлена на рис. 8 [17].

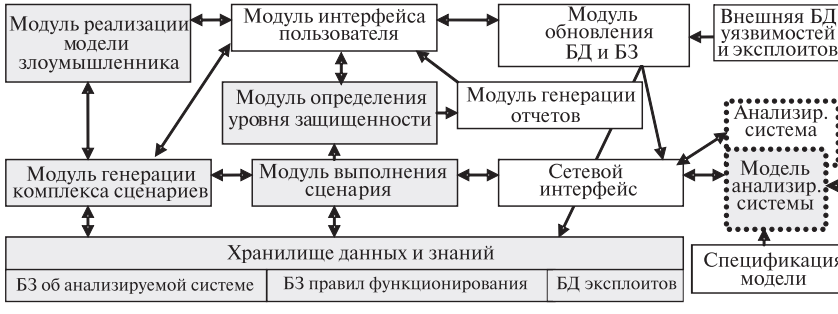

Рис. 8. Обобщенная архитектура системы активного анализа защищенности

Модуль реализации модели злоумышленника обеспечивает определение уровня умений злоумышленника, выбор стратегии поведения и определение цели атаки.

Хранилище данных и знаний состоит из базы знаний (БЗ) об анализируемой системе; базы правил функционирования САЗ и базы данных (БД) эксплоитов (программ реализации атак). Хранилище содержит данные и знания, которые используются злоумышленником для планирования и реализации атак.

База знаний об анализируемой системе содержит знания и данные об архитектуре и конкретных параметрах компьютерной сети, которые необходимы для генерации сценариев и выполнения атак (например, для конкретного хоста эти данные могут задавать тип и версию операционной системы, список открытых портов, запущенные приложения и т. п.). Эти данные обычно могут быть получены злоумышленником при реализации этапа разведки с помощью программных средств и методов социальной инженерии.

База правил функционирования содержит мета- и низкоуровневые правила вида «ЕСЛИ—ТО», определяющие действия САЗ на различных уровнях детализации. Метаправила определяют сценарии атак на высоком уровне. Низкоуровневые правила определяют атакующие действия на основе внешней базы уязвимостей. Часть «ЕСЛИ» каждого правила содержит цель действия и (или) условия выполнения данного действия. Цель выбирается согласно типу сценария и высокоуровневой цели, которая определяется метаправилом более высокого уровня. Условия выполнения действия сравниваются с данными, хранимыми в базе знаний об анализируемой системе. Часть «ТО» содержит идентификатор атаки, которая может быть выполнена при данных условиях, и (или) ссылку на эксплоит. Низкоуровневые правила данной базы создаются на основе одной из баз данных уязвимостей, например OSVDB (Open Source Vulnerability Database). База данных эксплоитов содержит программы реализации действий злоумышленника и параметры их использования.

Модуль генерации комплекса сценариев производит выбор данных об анализируемой системе из хранилища данных и знаний, генерирует комплекс сценариев атаки с использованием базы правил функционирования САЗ, осуществляет контроль выполнения комплекса сценариев и его изменение в процессе работы, а также выполняет обновление данных об анализируемой системе. Модуль выполнения этапа сценария осуществляет выбор следующего действия и эксплоита, прогнозирует ожидаемый отклик анализируемой компьютерной сети, реализует запуск эксплоита и распознавание отклика сети.

В случае взаимодействия с компьютерной сетью генерируется реальный сетевой трафик. При работе с моделью анализируемой системы обеспечивается два уровня эмуляции атак: (1) на первом уровне каждое низкоуровневое действие представляется идентификатором, описывающим тип атаки и (или) используемый эксплоит, а также параметрами атаки; (2) на втором (низком) уровне каждое действие представляется множеством сетевых пакетов.

Сетевой интерфейс обеспечивает: (1) в случае работы с моделью анализируемой системы — передачу идентификаторов и параметров атак (или сетевых пакетов в случае моделирования с большей степенью детализации), а также получение результатов атак и реакции системы; (2) при взаимодействии с реальной компьютерной сетью — передачу, захват и анализ сетевого трафика.

Модуль определения уровня защищенности использует разработанную таксономию метрик безопасности. Это основной модуль, который фиксирует сценарии атак в виде трасс прохождения различных компонентов системы, производит подсчет метрик безопасности, основываясь на информации о результате атак, и определяет «узкие места».

Модуль обновления баз данных и баз знаний использует открытые базы данных уязвимостей (например, OSVDB или NVD) и транслирует их в базу правил функционирования САЗ на низком уровне.

Окно интерфейса пользователя одного из разработанных прототипов САЗ (рис. 9) разделено на четыре функциональные части.

Левая верхняя часть («Network Model») отображает в виде дерева заданную системным администратором конфигурацию анализируемой компьютерной сети. Данная конфигурация изменяется в процессе выполнения атак (например, отображается остановка сетевого сервиса), и возвращается в исходное состояние после окончания каждого сценария. Правая верхняя часть («Malefactor's Network Model») отображает в виде дерева конфигурацию компьютерной сети так, как ее представляет себе злоумышленник. Изначально она пуста и заполняется в процессе выполнения атак. Эта конфигурация может иметь различия с заданной администратором конфигурацией, так как злоумышленник, как правило, обладает не всей информацией о сети, например, злоумышленник может узнать, что в сети функционирует 4 компьютера, а не 5, как задано в спецификации. Левая нижняя часть («Attack Tree») представляет собой сгенерированный системой сценарий выполнения атаки. Правая нижняя часть содержит три вкладки: (1) журнал выполняемых действий и результатов атак (лог); (2) обнаруженные уязвимости и трассы успешных атак; (3) вычисленные метрики безопасности.

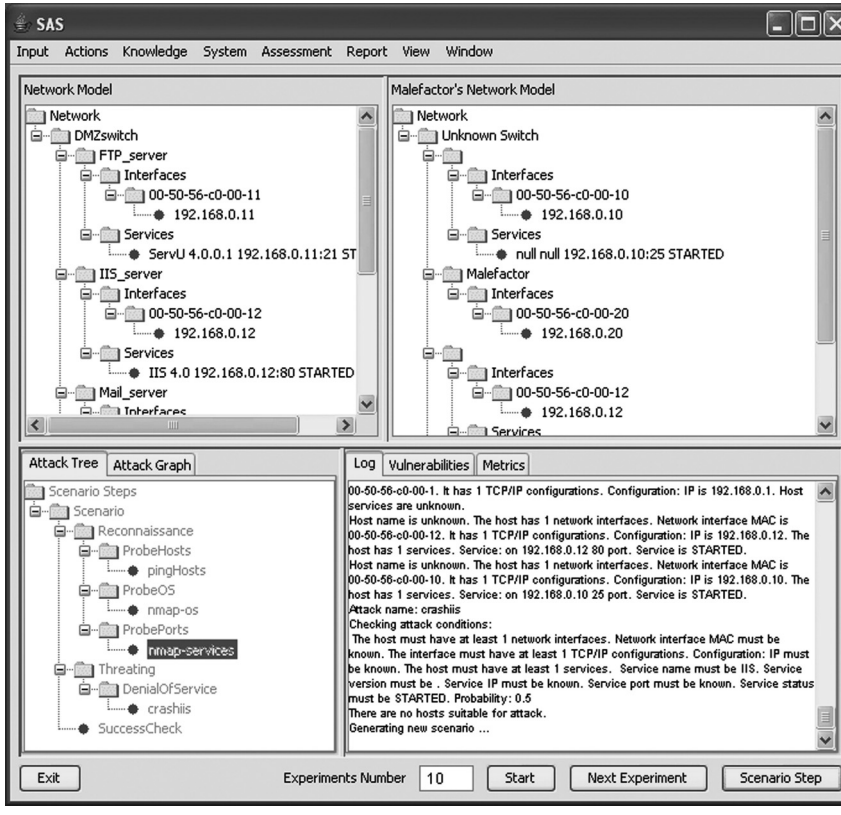

Рис. 9. Окно прототипа САЗ

В рамках работ по созданию архитектур, моделей и прототипов, осуществляющих *пассивный анализ уязвимостей*, ставится задача разработки компонентов, выполняющих следующие функции [18]:

- захват сетевого трафика и его анализ;
- анализ учетных записей (выявление учетных записей со слабыми паролями, количество пользователей с правами администратора, активен ли пользователь guest и т. п.);
- анализ установленного программного обеспечения (определение версий ПО и наличие программных коррекций);
- анализ журналов регистрации событий (операционной системы и приложений); анализ состояния файловой системы (проверка прав доступа и целостности файлов);

- обнаружение несоответствий с заданной политикой безопасности и конфигурацией сети;
- в случае обнаружения последних — генерация сигнала тревоги; определение уровня защищенности и генерация отчетов с советами по его увеличению;
- коррекция обнаруженных уязвимостей и отклонений от заданной политики безопасности; создание отчетов.

Архитектура пассивной САЗ, служащей для решения данной задачи, состоит из единой консоли управления и программных агентов, функционирующих на каждом устройстве сети (рис. 10).

Консоль предназначена для управления агентами, хранения заданной политики безопасности (на специализированном языке Security Policy Language (SPL)) и конфигурации сети (на языке System Description Language (SDL)), слепка их текущего состояния, обнаружения различий в заданной и текущей политике безопасности, взаимодействия с пользователем. *Сетевой интерфейс* поддерживает взаимодействие консоли управления и программных агентов. *Модуль корреляции данных* обеспечивает сбор, упорядочивание и фильтрацию информации от множества агентов. *Хранилище данных* состоит из двух основных частей: (1) заданной спецификации политики безопасности (на языке SPL) и конфигурации сети (на языке SDL); (2) текущего состояния политики безопасности и конфигурации сети. *Модуль обнаружения несоответствий с заданной спецификацией* производит сравнение значений параметров безопасности, полученных от программных агентов с соответствующими значениями, заданными в спецификации. В случае обнаружения различий от *модуля генерации сигнала тревоги* пользователю поступает соответствующее сообщение. Обнаруженное отклонение может быть устранено в двух режимах: в ручном и в автоматическом. В первом случае соответствующая команда поступает в *модуль коррекции* от пользователя; во втором — от модуля генерации сигнала тревоги. *Модуль определения уровня защищенности* производит анализ полученных от агентов данных и с использованием таксономии метрик безопасности определяет уровень защищенности и формирует рекомендации по его увеличению. *Модуль генерации отчетов* в наглядном виде отображает пользователю результаты анализа.

Программный агент использует API операционной системы для доступа к параметрам безопасности, журналам регистрации событий, сетевому трафику, производит первичную фильтрацию полученных данных. *Сетевой интерфейс* агента обеспечивает взаимодействие агента

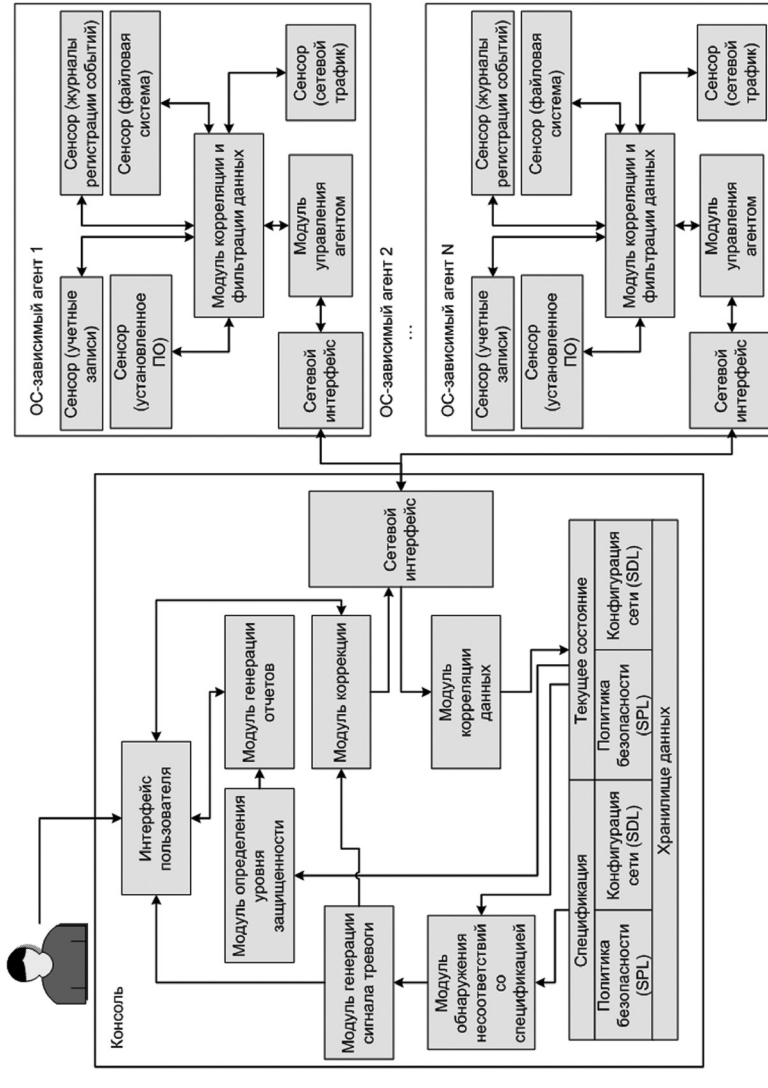

Рис. 10. Обобщенная архитектура компонентов пассивного анализа уязвимостей

с консолью управления и используется сетевым сенсором для захвата трафика. *Модуль управления агентом* организует внутренние процессы программного агента (своевременный опрос сенсоров и т. п.) *Модуль корреляции и фильтрации данных* собирает информацию с различных сенсоров, производит корреляцию и фильтрацию. *Сенсоры* служат для сбора информации из различных источников (файловая система, реестр операционной системы Windows, конфигурационные файлы различных операционных систем и приложений, сетевой трафик и т. п.), преобразуют ее в соответствующие сообщения для модуля корреляции и фильтрации данных.

Прототип пассивной САЗ реализуется с использованием языков программирования Java и C++ (в связке с Java Native Interface).

Механизмы моделирования поведения системы киберзащиты

В проводимых исследованиях развивается агентно-ориентированный подход к моделированию киберпротивоборства злоумышленников и систем защиты в виде антагонистического взаимодействия команд программных агентов, сформулированный в [19]. Выделяется по крайней мере две команды агентов, воздействующих на компьютерную сеть, а также друг на друга (рис. 11): команда агентов-злоумышленников и команда агентов защиты. Агенты различных команд соперничают для достижения противоположных намерений. Агенты одной команды сотрудничают для осуществления общего намерения.

Цель команды агентов-злоумышленников состоит в определении уязвимостей компьютерной сети и системы защиты и реализации заданного перечня угроз информационной безопасности (конфиденциальности, целостности и доступности) посредством выполнения распределенных скоординированных атак. Цель команды агентов защиты состоит в защите сети и собственных компонентов от атак.

Команда агентов-злоумышленников реализует развитые стратегии, включающие сбор информации о системе — цели нападения, обнаружение уязвимостей и используемых средств защиты, моделирование способов преодоления защиты, подавление, обход или обман средств защиты (например, посредством реализации «растянутого» во времени скрытого сканирования, выполнения отдельных скоординированных действий (атак) из нескольких различных источников, вместе составляющих сложную многофазную атаку и др.), использование уязвимостей и получение доступа к ресурсам, повышение полномочий, реализацию определенной угрозы, скрытие следов своей деятельности и создание «черных ходов» для использования их для последующего вторжения.

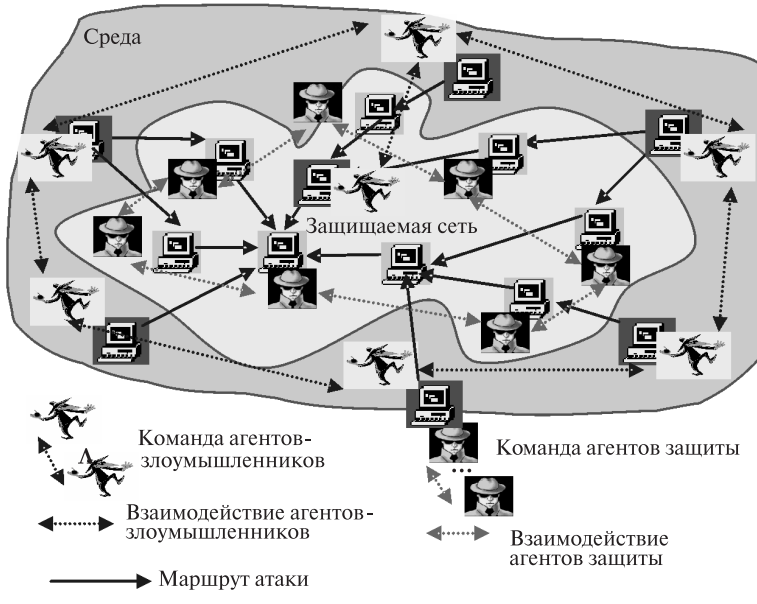

Рис. 11. Представление кибернетического противоборства в виде взаимодействия команд агентов

Примером автоматической стратегии является поражение сети Интернет, возникающее в результате распространения сетевых вирусов и червей, в том числе недавние эпидемии, высвечивающие тенденцию сращивания вирусных и спам-технологий и формирования объединенной, мотивированной сети агентов-злоумышленников.

Команда агентов защиты выполняет в реальном времени последовательность следующих действий: реализация механизмов защиты, соответствующих установленной политике безопасности (в том числе проактивного препятствования вторжениям, блокирования атак и их обнаружения); сбор информации о состоянии защищаемой системы и анализ обстановки; предсказание намерений и возможных действий злоумышленников; заманивание злоумышленников с использованием ложных информационных компонентов с целью введения в заблуждение и уточнения их целей; непосредственное реагирование на вторжения, в том числе усиление критичных механизмов защиты; устранение последствий вторжения, выявленных уязвимостей и адаптация системы обеспечения информационной безопасности к последующим вторжениям.

Структура команды агентов описывается в терминах иерархии групповых и индивидуальных ролей. Конечные узлы иерархии отвечают

ролям индивидуальных агентов, промежуточные узлы — групповым ролям. Механизмы взаимодействия и координации агентов базируются на трех группах процедур: (1) обеспечение согласованности действий; (2) мониторинг и восстановление функциональности агентов; и (3) обеспечение селективности коммуникаций (для выбора наиболее «полезных» коммуникационных актов). Спецификация иерархии планов действий осуществляется для каждой из ролей. Для каждого плана описываются: начальные условия, когда план предлагается для исполнения; условия, при которых план прекращает исполняться; действия, выполняемые на уровне команды, как часть общего плана. Для групповых планов явно выражается совместная деятельность.

Команда агентов-злоумышленников эволюционирует посредством генерации новых экземпляров и типов атак с целью преодоления подсистемы защиты. Команда агентов защиты адаптируется к действиям злоумышленников путем формирования новых экземпляров механизмов и профилей защиты.

Взаимодействие между агентами разных команд представляется как игра двух соперников, в которой целью агентов является поиск стратегии, которая максимизирует ожидаемый интегральный выигрыш в игре.

Чтобы справится с гетерогенностью и распределенностью источников информации и используемых агентов, в работе применяется основанный на онтологии подход и специальные протоколы для спецификации распределенного согласованного тезауруса понятий. Онтология предметной области обеспечения безопасности компьютерных сетей реализуется на базе стандартных языковых средств RDF или DAML+OIL.

Проектирование и реализация рассмотренной многоагентной системы были осуществлены на базе нескольких различных инструментариев: MASDK, JADE, OMNeT++ INET Framework. В настоящее время разработка ведется на базе пакета моделирования OMNeT++ INET Framework.

На основе OMNeT++ INET Framework разработана среда для многоагентного моделирования атак «Распределенный отказ в обслуживании» (DDoS) и механизмов защиты от них [20]. Для этого система INET Framework подверглась нескольким модификациям, в том числе были созданы: таблица фильтрации пакетов на сетевом уровне для моделирования действий стороны защиты; модуль, позволяющий просматривать весь трафик данного узла для ведения статистики, а также для моделирования действий стороны защиты. Подверглись изменению модули, отвечающие за работу Sockets для моделирования механизмов

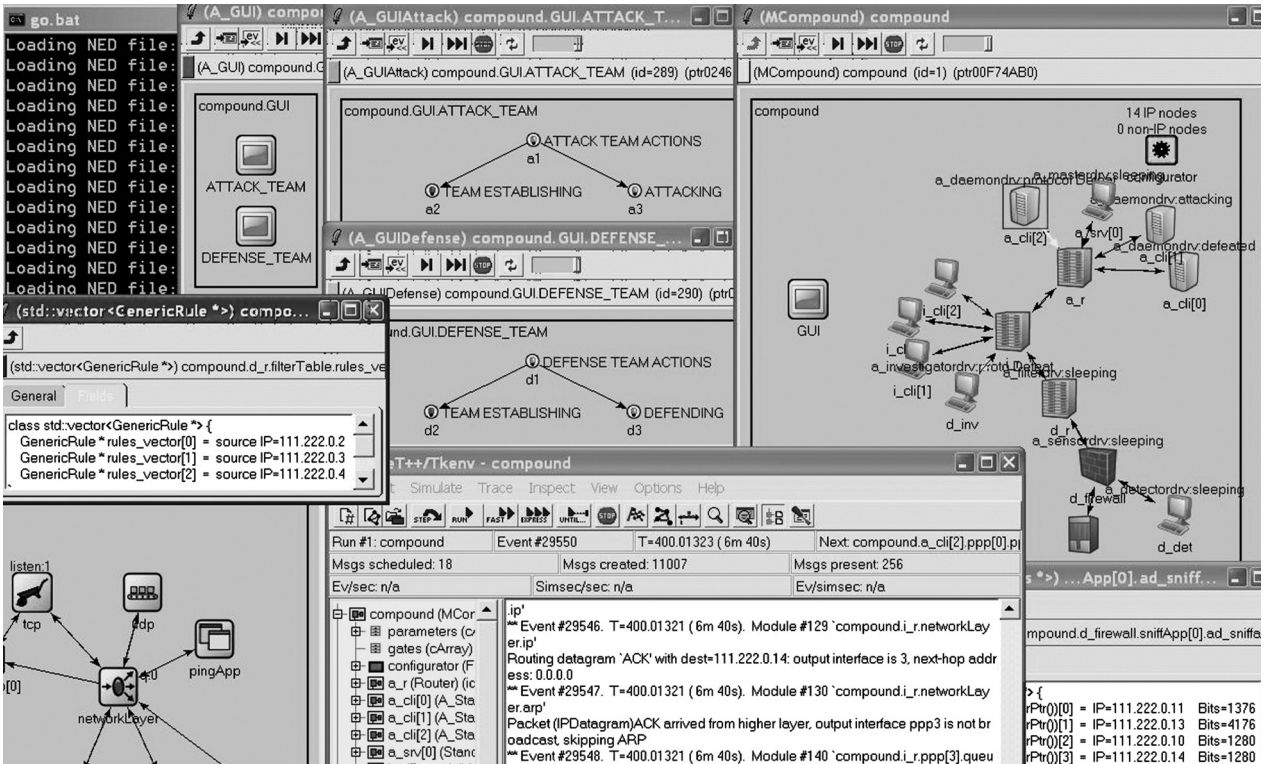

Рис. 12. Пример пользовательского интерфейса среды моделирования

атаки. Ядра агентов были выполнены на основе сопрограмм, так как это удобно для реализации протоколов взаимодействия, на которых основана командная работа агентов. Остальные модули выполнены как обработчики сообщений от ядра и внешней среды.

Пример пользовательского интерфейса среды моделирования показан на рис. 12. На основном окне визуализации (рис. 12, справа вверху) отображается компьютерная сеть для проведения моделирования. Окно управления процессом моделирования (рис. 12, внизу посередине) позволяет просматривать и менять параметры моделирования. Для отображения текущего состояния команд агентов служат соответствующие окна состояний (рис. 12, сверху посередине). Можно открывать различные окна, характеризующие функционирование (статистические данные) отдельных хостов, протоколов и агентов, например, на рис. 12 внизу слева отображено окно функционирования одного из хостов.

Компьютерная сеть для проведения моделирования состоит из трех подсетей:

1) подсеть защиты, на K узлах которой устанавливаются агенты защиты, и в которой можно выделить защищаемые серверы;
2) промежуточная подсеть, состоящая из N хостов с типовыми клиентами, генерирующими нормальный трафик;
3) подсеть атаки, включающая M узлов с демонами и один узел с мастером.

Характеристики подсетей задаются соответствующими параметрами моделирования.

На примере моделирования процессов реализации распределенных атак «отказ в обслуживании» проведен ряд экспериментов. Эксперименты показали эффективность предлагаемого подхода и возможность его использования для исследования перспективных механизмов защиты и анализа уровня защищенности проектируемых сетей. В дальнейшем планируется реализация большего количества механизмов защиты и атак, а также исследование механизмов внутрикомандного взаимодействия агентов.

4. Заключение

В статье предложен подход к разработке и использованию интеллектуальных адаптивных систем киберзащиты. Подход основан на реализации интеллектуальных механизмов управления защитой и построе-

нии единой унифицированной среды для создания и поддержки функционирования систем защиты на всем их жизненном цикле, включая адаптивное управление политиками безопасности.

В статье более детально охарактеризованы предложенные авторами работы интеллектуальные механизмы киберзащиты, в частности механизмы, основанные на использовании интеллектуальных агентов, механизмы дезинформации злоумышленника, сокрытия и камуфляжа важных ресурсов и процессов, «заманивания» злоумышленника на ложные (обманные) компоненты.

Представлены также механизмы создания и поддержки функционирования системы киберзащиты, в том числе механизмы определения уровня кибербезопасности и моделирования поведения системы киберзащиты.

Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00547), программы фундаментальных исследований ОИТВС РАН (контракт № 3.2/03), Фонда содействия отечественной науке, проекта Евросоюза RE-TRUST (контракт № 021186-2) и других проектов.

Литература

1. *Котенко И. В., Юсупов Р. М.* Технологии компьютерной безопасности // Вестник РАН. Т. 77. № 4. 2007. С. 323–333.
2. *Котенко И. В.* Модели и методы построения и поддержки функционирования интеллектуальных адаптивных систем защиты информации // Математические методы распознавания образов: 13-я Всероссийская конференция (ММРО-13). Ленинградская обл., г. Зеленогорск, 30 сентября – 6 октября 2007 г.: Сборник докладов. М.: МАКС Пресс, 2007. С. 599–602.
3. *Gorodetski V., Kotenko I., Skormin V.* Integrated Multi-Agent Approach to Network Security Assurance: Models of Agents' Community // Information Security for Global Information Infrastructures. IFIP TC11 Sixteenth Annual Working Conference on Information Security / Ed. by S. Qing, J. H. P. Eloff. Beijing, China, August 21–25, 2000. P. 291–300.
4. *Gorodetski V., Kotenko I.* The Multi-agent Systems for Computer Network Security Assurance: frameworks and case studies // IEEE ICAIS-02. IEEE International Conference «Artificial Intelligence Systems». Proceedings. IEEE Computer Society. 2002. P. 297–302.
5. *Gorodetsky V., Kotenko I., Karsayev O.* The Multi-agent Technologies for Computer Network Security: Attack Simulation, Intrusion Detection and Intrusion Detection Learning // The International Journal of Computer Systems Science & Engineering, 2003, No 4, P. 191–200.

6. *Котенко И. В.* Многоагентные технологии анализа уязвимостей и обнаружения вторжений в компьютерных сетях // Конфидент. 2004. № 2. С. 72–76; № 3. С. 78–82.
7. *Gorodetski V., Karsayev O., Kotenko I., Khabalov A.* Software Development Kit for Multi-agent Systems Design and Implementation // Lecture Notes in Artificial Intelligence, Vol.2296, Springer Verlag, 2002. P. 121–130.
8. *Gorodetski V., Kotenko I.* Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool // Recent Advances in Intrusion Detection. Fifth International Symposium. RAID 2002. Zurich, Switzerland. Lecture Notes in Computer Science. V. 2516. 2002. P. 219–238.
9. *Городецкий В. И., Котенко И. В., Карсаев О. В.* Интеллектуальные агенты для обнаружения атак в компьютерных сетях // КИИ-2000. VII Национальная конференция по искусственному интеллекту с международным участием. Труды конференции. М.: Издательство Физико-математической литературы, 2000. С. 771–779.
10. *Котенко И. В., Карсаев О. И.* Использование многоагентных технологий для комплексной защиты информации в компьютерных сетях // Известия ТРТУ, № 4. 2001. С. 38–50.
11. *Котенко И. В., Степашкин М. В.* Обманные системы для защиты информационных ресурсов в компьютерных сетях // Труды СПИИРАН. Вып. 2. СПб.: СПИИРАН, 2004. С. 211–230.
12. *Котенко И. В., Степашкин М. В.* Использование ложных информационных систем для защиты информационных ресурсов компьютерных сетей // Проблемы информационной безопасности. Компьютерные системы. 2005. № 1. С. 63–73.
13. *Котенко И. В., Степашкин М. В.* Системы-имитаторы: назначение, функции, архитектура и подход к реализации // Изв. вузов. Приборостроение. Т. 49. № 3. 2006. С. 3–8.
14. *Tishkov A., Kotenko I., Sidelnikova E.* Security Checker Architecture for Policy-based Security Management // Lecture Notes in Computer Science. Springer-Verlag. V. 3685. 2005. P. 469–474.
15. *Kotenko I., Stepashkin M.* Analyzing Vulnerabilities and Measuring Security Level at Design and Exploitation Stages of Computer Network Life Cycle // Lecture Notes in Computer Science. Springer-Verlag. V. 3685. 2005. P. 317–330.
16. *Богданов В. С., Котенко И. В., Степашкин М. В.* Активный анализ защищенности компьютерных сетей // Методы и технические средства обеспечения безопасности информации. Материалы XIV Общероссийской научно-технической конференции. СПб.: Издательство Политехнического университета, 2005.
17. *Котенко И. В., Степашкин М. В., Богданов В. С.* Архитектуры и модели компонентов активного анализа защищенности на основе имитации действий злоумышленников // Проблемы информационной безопасности. Компьютерные системы. 2006. № 2. С. 7–24.

18. *Степашкин М. В., Богданов В. С., Котенко И. В.* Подсистема пассивного анализа защищенности компьютерных сетей // Методы и технические средства обеспечения безопасности информации. Материалы XIV Общероссийской научно-технической конференции. СПб.: Издательство Политехнического университета, 2005.
19. *Котенко И. В., Уланов А. В.* Моделирование адаптивной кооперативной защиты от компьютерных атак в сети Интернет // Проблемы управления рисками и безопасностью. Труды Института системного анализа Российской академии наук (ИСА РАН). Т. 31. М.: URSS, 2007. С. 103–125.
20. *Kotenko I., Ulanov A.* Packet Level Simulation of Cooperative Distributed Defense against Internet Attacks // 16th Euromicro International Conference on Parallel, Distributed and network-based Processing (PDP 2008). Toulouse, France. February 13–15 2008. IEEE Computer Society. 2008. P. 565–572.