МНОГОАГЕНТНОЕ МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕННЫХ АТАК «ОТКАЗ В ОБСЛУЖИВАНИИ» И МЕХАНИЗМОВ ЗАЩИТЫ ОТ НИХ

И. В. Котенко[1], А. В. Уланов[2]

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

[1]<ivkote@iias.spb.su>, [2]<ulanov@iias.spb.su>

УДК 681.3.053:681.32:007.5

Котенко И. В., Уланов А. В. **Многоагентное моделирование распределенных атак «отказ в обслуживании» и механизмов защиты от них** // Труды СПИИРАН. Вып. 3, т. 1. — СПб.: Наука, 2006.

Аннотация. *Рассматривается подход к моделированию кибернетического противоборства команд интеллектуальных агентов на примере распределенных атак «отказ в обслуживании» и механизмов защиты от них. Разработана программная среда моделирования на базе OMNeT++ INET Framework. Среда включает агентские компоненты и библиотеки атак и механизмов защиты. Описаны проведенные эксперименты.* — Библ. 58 назв.

UDC 681.3.053:681.32:007.5

Kontenko I. V., Ulanov A. V. **Multi-agent modeling of distributed "Denial of Service" attacks and defense mechanisms** // SPIIRAS Proceedings. Issue 3, vol. 1. — SPb.: Nauka, 2006.

Abstract. *Paper describes the approach for modeling of cybernetic counteraction of intelligent agents on the example of distributed denial of service attacks and defense mechanisms. The simulation software environment based on OMNeT ++ INET Framework is developed. The environment includes agency components and libraries of attacks and protection mechanisms. The fulfilled experiments are described.* — Bibl. 58 items.

1. Введение

В настоящее время Интернет постоянно находится под воздействием атак различных злоумышленников, зачастую достигающих своих целей [1]. К сожалению, существующая теоретическая база для обеспечения информационной безопасности в Интернет не предоставляет возможности адекватно формализовать комплекс процессов, связанных с противодействием систем защиты и средствами нападения злоумышленников. Хотя исследователи в состоянии представить отдельные механизмы защиты, понимание системы обеспечения информационной безопасности как единой (холической) системы, зависящее от учета множества взаимодействий между отдельными процессами ее функционирования и киберпротивостояния между различными элементами, а также развивающегося динамического характера этих процессов и отдельных компонентов информационных систем, чрезвычайно затруднено.

Рассмотрим указанную выше проблему на примере исследования и реализации механизмов защиты от одного из наиболее критичных по последствиям классов компьютерных атак — «Распределенный отказ в обслуживании».

В результате известной атаки «отказ в обслуживании» (Denial of Service, DoS), сводящейся, как правило, к передаче большого количества сетевых пакетов с одного их хостов сети, законный пользователь не может получить доступ к необходимой ему информации. Большинство операционных систем, маршрутизаторов и компонентов сетей подвержены атакам DoS, предотвратить которые очень сложно.

Для проведения распределенных атак «отказ в обслуживании» (Distributed Denial of Service, DDoS) [2] злоумышленник должен сначала взломать ряд компьютеров для запуска на них средств DoS и последующего одновременного нападения на некоторый компьютер или сеть. Это существенно усложняет как обнаружение, так и защиту от атак данного класса. Известно множество различных видов атак DDoS. Условно их можно разделить на две категории: истощение ресурсов сети и истощение ресурсов хоста. Атаки осуществляются с помощью непосредственной посылки жертве большого количества пакетов или использования для этой цели промежуточных узлов, передачи слишком длинных пакетов, некорректных пакетов или большого количества трудоемких запросов.

Построение эффективной системы защиты от атак DDoS является сложной задачей. Эффективная система защиты должна включать механизмы предупреждения атаки, обнаружения факта атаки, определения источника атаки и противодействия атаке. Стандартной мерой защиты подсети (не только от атак DDoS) является установка правил фильтрации пакетов от зарезервированных IP-адресов (например, для сетевых пакетов, входящих с адресами из внутренней подсети, выходящих с адресами, отличающимися от внутренних, необычных по размеру; к тем и от тех портов, которые не задействованы в системе; по неиспользуемым протоколам и др.). Кроме того, применяется ограничение на трафик для каждого протокола и для входящих/выходящих потоков и множество других мер. Зная эти меры, злоумышленник может таким образом модифицировать параметры атаки DDoS (например, на основе изменения IP-адреса отправителя), что ее будет невозможно отличить от, например, запросов пользователей, вызванных повышенным интересом к данному серверу. Это приводит к усложнению механизмов защиты.

Разработать адекватные методы защиты от атак DDoS и выработать обоснованные рекомендации по выбору механизмов защиты, наиболее действенных в конкретных условиях, можно, используя исследовательское моделирование атак DDoS и механизмов защиты от них.

Формализация, моделирование и исследование противоборства злоумышленников и систем защиты в сети Интернет на примере моделирования процессов реализации распределенных атак «отказ в обслуживании» и механизмов защиты может позволить получить результаты, обобщаемые на другие задачи, в частности, на задачи информационной борьбы в Интернет, конкуренции в сфере электронного бизнеса и др. [3].

В статье на примере атак DDoS и механизмов защиты от них описываются основные аспекты предлагаемого подхода к исследованию противоборства злоумышленников и систем защиты в сети Интернет. Подход основан на агентно-ориентированном моделировании процессов противоборства антагонистических агентов в среде Интернет. В статье рассматриваются методы организации командной работы агентов, механизмы их взаимодействия и планы действий. Дается представление о разработанной среде моделирования атак и механизмов защиты, а также рассматривается пример одного из реализованных сценариев моделирования.

2. Подход к моделированию противоборства

Использование основанного на многоагентных технологиях моделирования процессов обеспечения информационной безопасности в сети Интернет предполагает, что кибернетическое противоборство представляется в виде

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

взаимодействия различных команд программных агентов [4]. Агрегированное поведение системы проявляется посредством локальных взаимодействий отдельных агентов в динамической среде, задаваемой посредством модели сети.

Выделяется две команды агентов, воздействующих на компьютерную сеть, а также друг на друга: команда агентов-злоумышленников по реализации атак DDoS и команда агентов защиты.

Задача многоагентного моделирования процессов кибернетического противоборства представляется как моделирование антагонистического взаимодействия, по крайней мере, одной команды агентов-злоумышленников и одной команды агентов защиты. Цель команды агентов-злоумышленников заключается в определении уязвимостей компьютерной сети и системы защиты и реализации заданного перечня угроз информационной безопасности посредством выполнения распределенных скоординированных атак. Цель команды агентов защиты состоит в защите сети и собственных компонентов от атак.

Агенты различных команд соперничают для достижения противоположных намерений. Агенты одной команды сотрудничают для осуществления общего намерения (по реализации угрозы или по защите компьютерной сети).

Предполагается, что соперничающие агенты осуществляют сбор информации из различных источников, оперируют нечеткими (вероятностными) знаниями, прогнозируют намерения и действия оппонента, оценивают возможные риски, пытаются обмануть друг друга, реагируют на действия оппонента.

Выбор сценария поведения каждой из команд зависит, прежде всего, от выбранной цели функционирования, а конкретная реализация сценария определяется, в первую очередь, непосредственной реакцией противоположной команды. Поэтому выбор каждого очередного шага поведения каждой из команд должен определяться динамически в зависимости от действий противоположной команды и состояния среды.

Каждая команда действует в условиях ограниченной информации, а каждый член команды может обладать различной информацией о действиях других членов команды. Поэтому модель поведения агентов должна быть в состоянии отображать неполноту информации и возможность возникновения случайных факторов. Кроме того, само поведение агентов должно зависеть от информации, которой владеет команда, и от ее распределения на множестве отдельных агентов, входящих в состав команды [3].

Модели функционирования агентов должны предусматривать, что каждый агент «знает», какие задачи он должен решать сам и к какому агенту он должен адресовать свой запрос на информацию или на решение подзадачи с целью получения такой информации, если это вне его компетенции. Сообщения одних агентов представляются в форме и терминах, понятных другим агентам [5].

Одним из наиболее перспективных подходов к структуризации распределенных баз знаний такого типа, является использование онтологий, характеризующих предметные знания сами по себе, вне связи с конкретными структурами их представления, алгоритмами вывода в них или эвристиками [5, 6]. Как и для любой другой предметной области, онтология области защиты информации представляет собой описание частично упорядоченного множества понятий, которые должны использоваться соответствующими агентами. Кроме отношения частичного порядка, на узлы этой структуры накладываются и другие отношения, свойственные предметной области. Это различного рода ограничения, правила, количественные и качественные отношения, связывающие понятия рассматриваемой предметной области. Данная онтология определяет подмно-

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

жество понятий, которые используют различные агенты для кооперативного решения поставленных задач. Каждый агент использует определенный фрагмент общей онтологии предметной области.

Специализация каждого агента отражается подмножеством узлов онтологии. Некоторые узлы онтологии могут быть общими для пары или большего количества агентов. Обычно только один из этих агентов обладает детально структурированным описанием этого узла. Именно этот агент является обладателем соответствующего фрагмента базы знаний. В то же время, некоторая часть онтологических баз знаний является общей для всех агентов, и именно эта часть знаний является тем фрагментом, который должен играть роль общего контекста (общих знаний).

Предполагается, что агенты могут реализовать механизмы самоадаптации и эволюционировать в процессе функционирования. Команда агентов-злоумышленников эволюционирует посредством генерации новых экземпляров и типов атак, а также сценариев их реализации с целью преодоления подсистемы защиты. Команда агентов защиты адаптируется к действиям злоумышленников путем изменения исполняемой политики безопасности, формирования новых экземпляров механизмов и профилей защиты.

Взаимодействие между агентами может быть представлено как игра двух соперников, в которой целью агентов является поиск стратегии, которая максимизирует ожидаемый интегральный выигрыш в игре [7–9].

Стратегии функционирования агентов могут быть представлены посредством различных формализмов, например, на основе семейства стохастических атрибутных формальных грамматик (и их интерпретации с использованием автоматов) и скрытых марковских моделей.

Концептуальная модель кибернетического противоборства включает в себя: (1) онтологию приложения в области защиты информации, содержащую множество понятий приложения и отношений между ними; (2) протоколы командной работы агентов различных команд (команд злоумышленников и команд (компонентов) системы защиты информации); (3) модели сценарного индивидуального, группового и общекомандного поведения агентов в рамках конкретных намерений, реализуемых сценариями; (4) коммуникационную компоненту, предназначенную для обмена сообщениями между агентами; (5) модели среды функционирования — компьютерной сети, включающие топологический и функциональные компоненты.

Предлагаемая *технология создания команды агентов* заключается в реализации следующей цепочки этапов [10]: (1) формирование онтологии предметной области; (2) определение структуры команды агентов и механизмов их взаимодействия и координации (в том числе задание ролей и сценариев обмена ролями между агентами); (3) спецификация иерархии планов действий (генерации атак); (4) назначение ролей и распределение планов между агентами.

Для исследовательского моделирования процессов кибернетического противоборства предлагается использовать семейство различных моделей (от аналитических до полунатурных и натурных) (рис. 1) [11].

Выбор конкретных моделей диктуется необходимой точностью и масштабируемостью моделирования. Например, аналитические модели позволяют имитировать глобальные процессы, происходящие в Интернет (в том числе вирусные эпидемии), однако эти модели описывают моделируемые процессы только на абстрактном уровне. Имитационное моделирование на уровне пакетов предоставляет возможность достаточно адекватно воспроизводить проте-

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

кающие процессы, представляя атакующие и защитные действия с помощью обмена сетевыми пакетами, точно имитируя работу по протоколам канального, сетевого, транспортного и прикладного уровней. Наибольшая точность имитации достигается на аппаратных стендах при натурном моделировании, однако при этом удается моделировать достаточно ограниченные фрагменты взаимодействий агентов.

Рис. 1. Семейство моделей, используемых для исследовательского моделирования компьютерного противоборства.

Основное внимание в настоящей работе уделяется применению имитационного моделирования на уровне пакетов с использованием в качестве базового уровня среды моделирования соответствующих средств имитационного моделирования, позволяющих имитировать сетевые процессы.

3. Релевантные работы и сущность подхода

В качестве начального фундамента для исследований в области моделирования противоборства злоумышленников и систем защиты в сети Интернет, используются работы в следующих областях: агентно-ориентированное моделирование; командная работа агентов; системы вывода, основанные на предсказании намерений и планов оппонента; рефлексивные процессы; теоретико-игровое моделирование; моделирование атак на компьютерные сети; моделирование процессов защиты информации; адаптивные системы и эволюционные вычисления.

Методы агентно-ориентированного моделирования являются сравнительно молодой областью применения теории многоагентных систем, поэтому решение поставленной проблемы должно привести, в том числе, и к обогащению этого направления. Основной базис для исследования составляет *теория командной работы агентов*. Известно три широко известных подхода к формализации командной работы агентов: *теория общих намерений* [12], *теория общих планов* [13] и гибридный подход, базирующийся на *комбинировании*

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

теорий общих намерений и общих планов [14]. Многие подходы к организации командной работы агентов воплощены в программных реализациях различных многоагентных систем, например, в системах GRATE*, OAA, CAST, RETSINA-MAS, COGNET/BATON, Team-Soar и др. Важным полигоном для исследования командной работы агентов является «виртуальный футбол» (футбол роботов) и моделирование спасательных действий команд агентов в различных критических ситуациях (при стихийных бедствиях, террористических актах и т.п.).

Еще одной фундаментальной составляющей проводимых исследований являются работы в области *систем вывода, основанных на знаниях о выполняемых действиях и предсказании намерений и планов оппонента* на основе оценки текущей ситуации. Наряду со ставшими уже классическими работами Е.Чарниака [21], сформулировавшего задачу распознавания как задачу абдуктивного вывода, Х.Каутца и Д.Алена [22], рассматривающих распознавание плана на основе идентификации минимального множества высокоуровневых действий, которые достаточны для объяснения наблюдаемых событий, М.Вилейна [23], использующего для распознавания методы грамматического анализа, М.Веллмана и Д.Пинадаса [24], предложивших механизмы байесовского распознавания, и др., сравнительно недавно были опубликованы работы по определению планов злоумышленников при обнаружении вторжений, в частности, работы К.Гейба и Р.Голдмана [25, 26]. Предполагается использовать идеи распознавания планов действий агентов на основе алгоритмов восстановления стохастических формальных грамматик, изученных авторами настоящей статьи в результате предыдущих исследований [27].

Важной компонентой, необходимой для использования в работе, являются методы *теории рефлексивных процессов* [28–30 и др.], *теоретико-игрового информационного моделирования* [7–9, 31 и др.] и *управления в конфликтных ситуациях* [32 и др.].

Используемые авторами методы спецификации сценариев действий агентов, основанные на стохастических атрибутных формальных грамматиках [33], можно соотнести с развиваемой в настоящее время *теорией построения систем (колоний) кооперативных распределенных грамматик* и грамматическими моделями многоагентных систем [34–37].

Команды агентов атаки и защиты должны адаптироваться к реконфигурации аппаратного и программного обеспечения сети, к изменению трафика, а также к новым видам защиты и атакам на основе прошлого опыта и алгоритмов. Поэтому важно учитывать существующие исследования в области *адаптации и самообучения агентов* [38–44 и др.].

Предлагаемый в работе подход к организации командной работы агентов базируется на совместном использовании элементов теории общих намерений, теории разделяемых планов и комбинированных подходов и учитывает опыт программной реализации ряда многоагентных систем [45, 46].

Предполагается, что командная работа агентов организуется с помощью *общего (группового) плана действий*, особенности которого заключаются в следующем [47]: (1) групповой план действий требует, чтобы команда агентов пришла к согласию выполнять предписание (множество заданных инструкций); (2) агенты должны принять на себя обязательства по отношению не только к своим индивидуальным действиям, но также к действиям других агентов и действиям группы в целом; (3) план групповой деятельности может иметь в качестве компонентов как планы индивидуальных агентов для назначенных действий, так и планы подгрупп; (4) при выполнении командной работы агенты команды

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

должны с помощью коммуникаций прийти к согласию с предписанием, а также согласовать собственные намерения друг с другом.

Структура команды агентов описывается в терминах иерархии групповых и индивидуальных ролей. Листья иерархии отвечают ролям индивидуальных агентов, промежуточные узлы — групповым ролям. 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.

Механизмы взаимодействия и координации агентов базируются на трех группах процедур [14, 46]: (1) обеспечение согласованности действий; (2) мониторинг и восстановление функциональности агентов; и (3) обеспечение селективности коммуникаций.

Процедуры обеспечения согласованности действий агентов необходимы для поддержки скоординированной деятельности агентов по некоторому сценарию. Эти процедуры реализуются путем обмена агентами информацией о результатах деятельности, которые непосредственно влияют на выполнение поставленной задачи. До начала реализации атаки DDoS происходит формирование агентов, до их сведения доводятся их роли. Далее агенты сообщают о своей готовности и начинают активные действия в соответствии с заданной ролью. При достижении поставленной цели, обнаружении невозможности выполнить цель или выявлении нерелевантности цели, агент обязан сообщить этот факт оставшимся членам команды. При этих условиях выполняемый сценарий завершается, и должен быть активизирован другой сценарий.

Процедуры мониторинга и восстановления функциональности агентов направлены на сохранение работоспособности и функциональности команды агентов. Их реализация может происходить с использованием различных приемов, например, за счет перераспределения ролей среди оставшихся агентов взамен выбывших или путем генерации новых агентов с соответствующей ролью и функциональностью.

Процедуры обеспечения селективности коммуникаций служат для минимизации количества коммуникативных актов с целью уменьшения вероятности раскрытия агентов и сокращения используемых ресурсов. Эти процедуры реализуются на основании знаний о выгоде коммуникационного акта и «затратах» на его обеспечение.

4. Команды агентов атаки и защиты

Агенты атаки подразделяются, по крайней мере, на два класса: «демоны», непосредственно реализующие атаку, и «мастер», выполняющий действия по координации остальных компонентов системы.

На предварительном этапе демоны и мастер устанавливаются на доступные (уже скомпрометированные) узлы сети Интернет. Здесь важными параметрами являются количество и распределенность агентов. Затем происходит организация команды атаки: демоны посылают мастеру сообщения о том, что они существуют и готовы к работе, а мастер сохраняет информацию о членах команды и об их состоянии.

Злоумышленник задает общую цель команды — начать атаку DDoS в заданный момент времени. Параметры атаки получает мастер. Его цель — разо-

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

слать их всем доступным демонам. Далее в действие вступают демоны. Их локальная цель — исполнить команду мастера. Для этого на указанный узел отсылаются пакеты атаки в заданном мастером режиме. После этого считается, что цель команды на данном этапе достигнута.

Периодически мастер опрашивает демонов, для того, чтобы узнать о том, что они находятся в работоспособном состоянии. Получая сообщения от демонов, мастер контролирует заданный режим выполнения атаки. Если от какого-либо демона не поступает сообщений о состоянии, мастер принимает решение об изменении параметров атаки. Например, он может послать команды всем или только определенным демонам об изменении интенсивности атаки.

Демоны могут выполнять атаку в различных режимах. Это влияет на возможности команды защиты по обнаружению и блокированию атаки, а также прослеживанию и устранению агентов атаки. Режим задается интенсивностью посылки пакетов (пакетов в секунду) и способом подмены адреса отправителя в пакете («IP spoofing»). Способ подмены может быть следующий: (1) без подмены («no») — используется адрес узла, на котором установлен демон; (2) постоянный («constant») — случайным образом выбирается некоторый адрес, который затем используется при посылке всех пакетов; (3) случайный («random») — при посылке каждого пакета случайным образом выбирается новый адрес из диапазона адресов, не используемых в данной сети; (4) случайный настоящий («random real») — при посылке каждого пакета случайным образом выбирается адрес из диапазона адресов, используемых в данной сети.

Злоумышленник может прекратить атаку. Он задает мастеру команду «завершить атаку». Затем мастер рассылает соответствующие команды демонам. Получив эту команду, демоны прекращают атаку.

В соответствии с общим подходом, выделены следующие классы *агентов защиты* [3]: первичной обработки информации («сенсоры»); вторичной обработки информации («сэмплеры»), обнаружения атаки («детекторы»); фильтрации («фильтры»); агенты расследования.

В начальный момент времени агенты защиты устанавливаются на соответствующие их ролям узлы: сенсор — на пути следования трафика для защищаемого узла; сэмплер — также на пути следования трафика для защищаемого узла; детектор — на любой узел в подсети защищаемого узла; фильтры — на входе в подсеть защищаемого узла; агент расследования — за пределами подсети защищаемого узла на любом доступном из Интернета узле.

Общая цель команды агентов защиты — противостояние атаке DDoS. За ее выполнением следит детектор.

Сенсор обрабатывает информацию о сетевых пакетах и собирает статистические данные по трафику для защищаемого узла. Сенсор определяет величину всего трафика (бит в секунду (bit per second) — BPS), а также адреса n узлов, создающих наибольший трафик (в реализованном прототипе — все хосты). Его локальная цель — предоставлять эти данные каждые k секунд детектору на обработку.

Сэмплер обрабатывает информацию о сетевых пакетах и на ее основе составляет модель нормального для данной сети трафика (в режиме обучения). Затем, в нормальном режиме, он анализирует сетевой трафик на соответствие модельному и выделяет IP-адреса нарушителей, которые затем отсылает детектору. Для обнаружения атаки используются методы «Фильтрация по количеству хопов» (Hop counts Filtering — HCF), «Мониторинг исходных IP-адресов»

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

(Source IP address monitoring — SIPM) и «Бит в секунду» (Bit Per Second — BPS).

Локальная цель детектора — на основе данных от сенсора и сэмплера принять решение о наступлении атаки. Для данных от сенсора, если детектор определяет, что параметр *BPS* превышает заданный предел (определяемый как процент от максимальной скорости пропускания канала связи), то он считает происходящее атакой DDoS. Он посылает фильтру и агенту расследования список адресов узлов, создающих наибольший трафик. Детектор также посылает фильтру и агенту расследования адреса, полученные от сэмплера.

Локальная цель фильтра — выполнить фильтрацию трафика на основе данных от детектора. Если в сообщении содержится решение о проведении атаки, то фильтр начинает отбрасывать пакеты от указанных узлов.

Цель агента расследования — идентифицировать и вывести из строя агентов атаки. После приема сообщения от детектора он проверяет указанные адреса на наличие агентов команды атаки и пытается вывести идентифицированных агентов из строя. Для упрощения модели сделано допущение, что вероятность вывода из строя 30%.

При обнаружении атаки детектор посылает фильтру для фильтрации адреса узлов, создающих наибольший трафик. Как только по информации от сенсора детектор решит, что атака прекратилась, цель команды агентов защиты на заданном временном промежутке будет достигнута.

Рассмотрим три примера методов, используемых сэмплером.

Hop counts Filtering (HCF) [8]. Используется предположение, что пакеты из одной и той же сети проходят до получателя одинаковое количество интервалов (скачков, хопов) между точками маршрутизации. Количество хопов, преодоленных пакетами по пути прохождения, оценивается с помощью поля TTL пакета. Каждый маршрутизатор, через который проходит пакет, отнимает от значения TTL единицу. В режиме обучения на основе запросов к защищаемому узлу составляется таблица, в которой IP-адреса узлов группируются по количеству хопов. В нормальном режиме система вычисляет количество хопов пришедшего пакета и сравнивает его с табличным значением. Если количество хопов пакета не совпадает с табличным значением, то пакет отбрасывается.

Source IP address monitoring (SIPM) [31]. Используется предположение, что в начале атаки появляется большое количество новых IP-адресов. В режиме обучения составляется таблица легитимных IP-адресов при обращении клиентов к защищаемому узлу. В нормальном режиме и режиме обучения по входящим пакетам собирается статистика по количеству новых для системы IP-адресов за заданные интервалы времени dt с определенным сдвигом $tshift$. Это значит, что статистика вычисляется каждые $tshift$ секунд по предыдущим dt секундам. В режиме обучения определяется максимальная величина (порог) количества новых IP-адресов. Затем, в нормальном режиме, если количество новых IP-адресов остается в пределах нормы, то эти адреса заносятся в базу. Если данная величина в течение нескольких интервалов времени (NIP) была выше нормы (такая агрегация называется методом кумулятивных сумм, CUSUM), то пакеты от новых IP-адресов отбрасываются.

Bit Per Second (BPS). Используется предположение, что трафик от одного IP-адреса не должен превышать некоторое критическое значение (порог). В режиме обучения для каждого клиента, обращающегося к защищаемому узлу, формируются статистические данные о количестве переданных бит за интервал времени. Определяется наибольший показатель за время обучения. В нор-

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

мальном режиме, если BPS для какого-то IP-адреса превышает заданное значение (порог), то пакеты от этого адреса блокируются. Здесь параметры вычисляются также каждые *tshift* секунд по предыдущим *dt* секундам.

Ключевыми параметрами для функционирования сэмплера являются величины пороговых значений, интервалы времени *dt* и сдвиг *tshift* для SIPM и BPS. Для SIPM ключевым параметром является также максимальное количество интервалов, когда порог был выше нормы.

5. Среда моделирования и архитектура агентов

Пример многооконного пользовательского интерфейса среды моделирования показан на рис. 2 и рис. 3. На основном окне визуализации (рис. 3, вверху справа) отображается компьютерная сеть для проведения моделирования.

Окно управления процессом моделирования (рис. 3, внизу справа) позволяет просматривать и менять параметры моделирования. Более детальное представление этого окна отображено на рис. 2. В данном окне на шкале времени можно наблюдать события, значимые для понимания атак и механизмов защиты. Шкала времени отображается над окном с текстовым описанием событий. На рис. 2 можно видеть, например, события посылки пакета ACK, действие сенсора, инициирование атаки и др.

Для отображения текущего состояния команд агентов служат соответствующие окна состояний (рис. 3, сверху посередине; а также рис. 4). Можно открывать различные окна, характеризующие функционирование (статистические данные) отдельных хостов, протоколов и агентов, например, на рис. 3 (слева) отображено несколько окон, характеризующих в графическом и текстовом виде (в том числе в форме графика зависимости количества переданных бит от времени) функционирование одного из хостов.

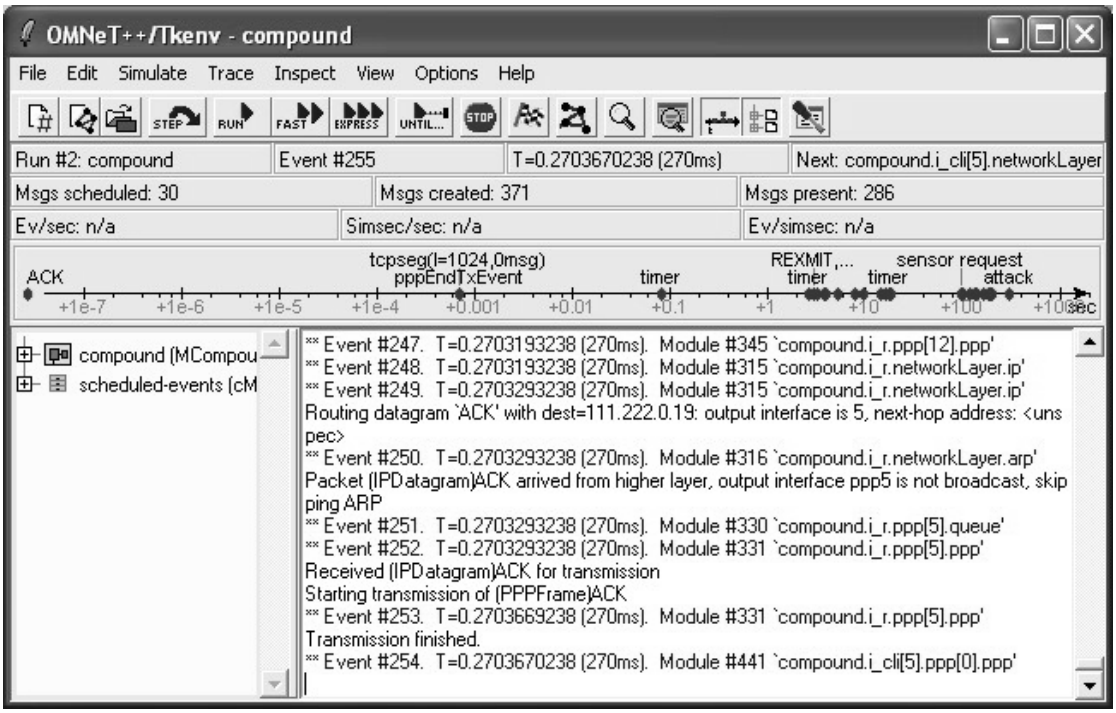

Рис. 2. Окно управления процессов моделирования.

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

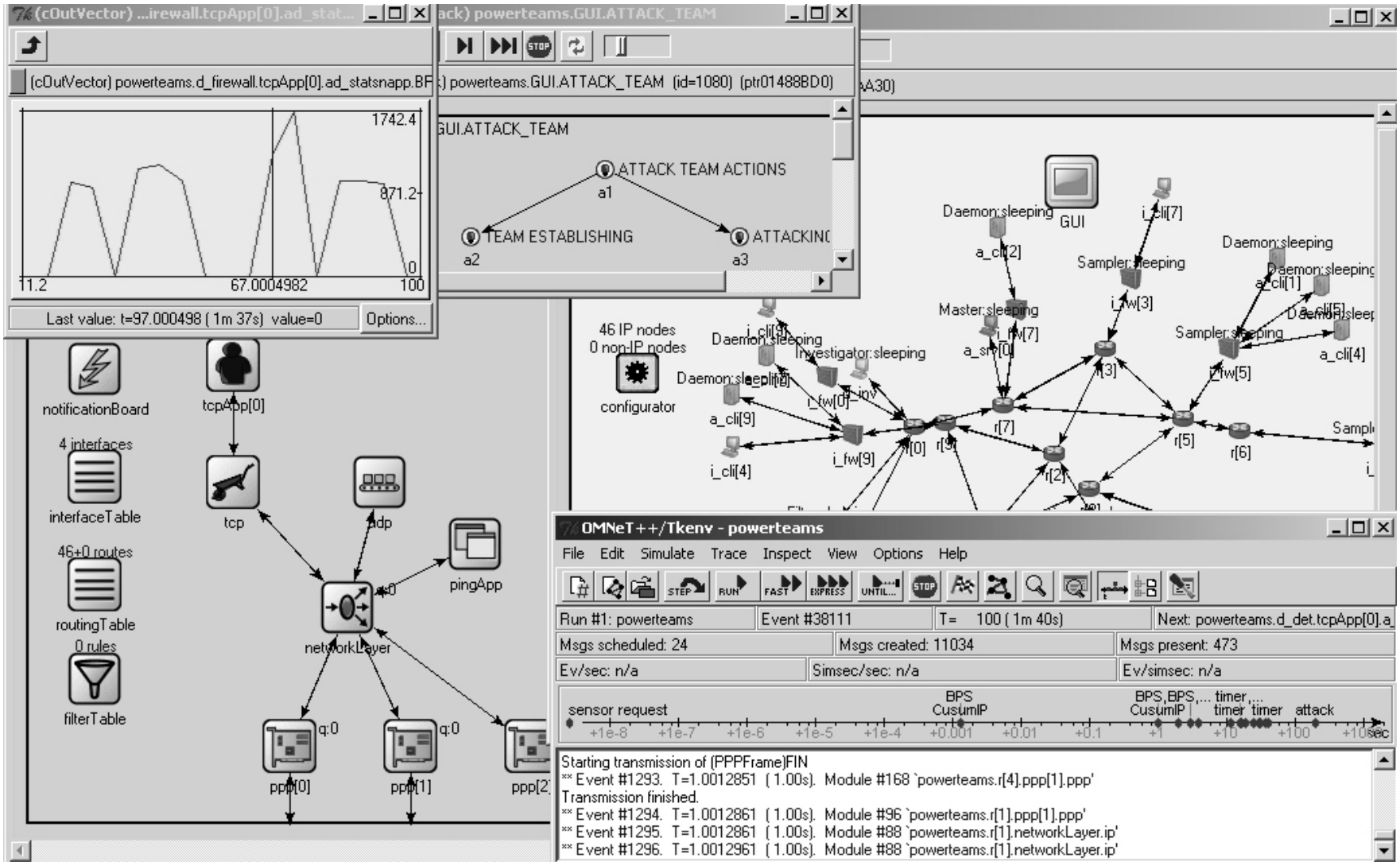

Рис. 3. Пример пользовательского интерфейса среды моделирования.

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

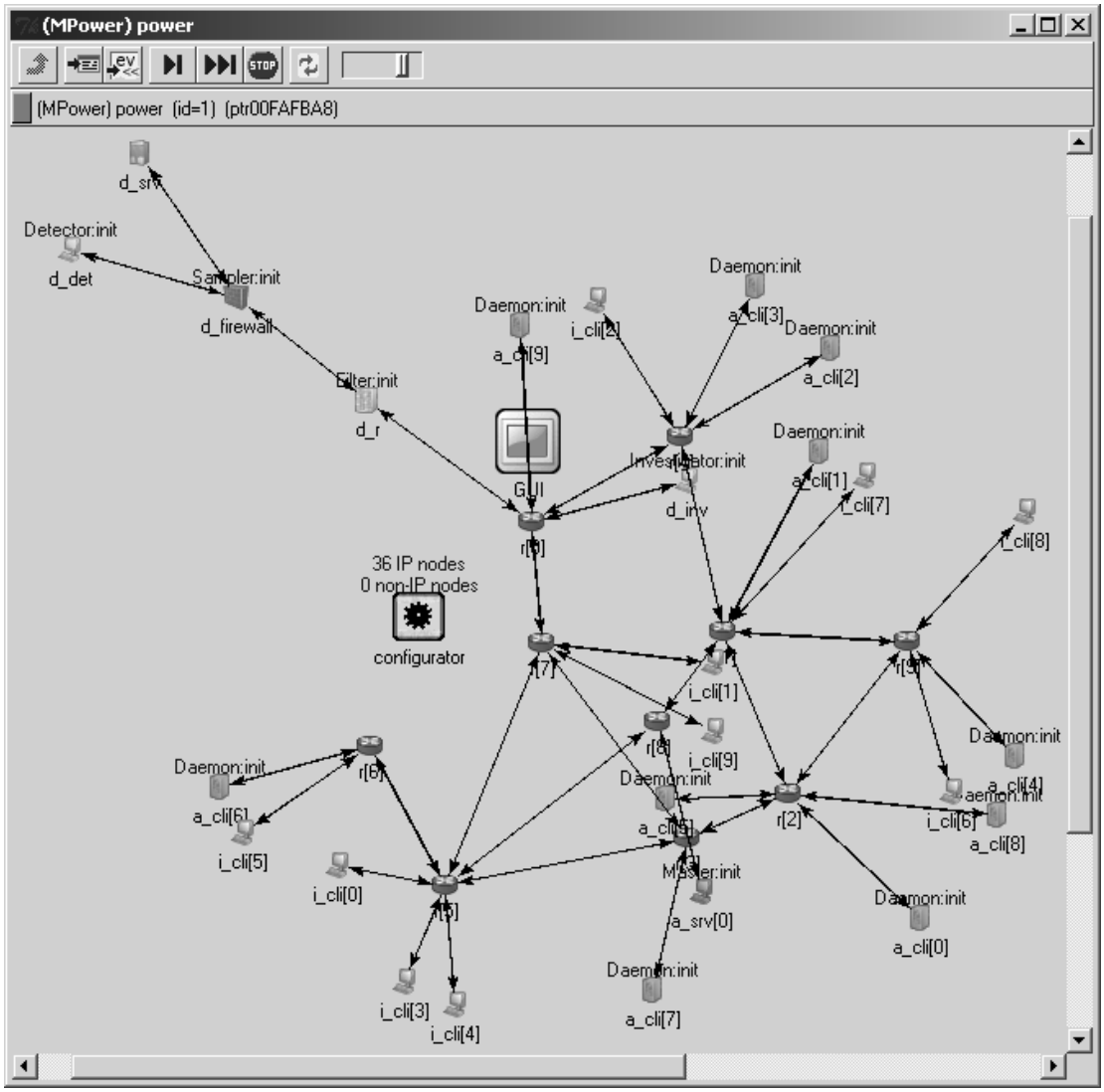

Рис. 4. Пример структуры компьютерной сети для проведения моделирования.

Пример основного окна, на котором отображается компьютерная сеть для проведения моделирования, показан на рис. 4. Исследуемая компьютерная сеть представляет собой набор узлов, соединенных каналами связи. Узлы могут нести различную функциональность в зависимости от их параметров или набора внутренних модулей. Овальным значком обозначены маршрутизаторы. Красным подсвечены узлы, на которых располагаются агенты команды атаки, зеленым — узлы, на которых находятся агенты команды защиты. Над окрашенными узлами есть соответствующие надписи, говорящие о типе агента и его состоянии. Остальные узлы — типовые, создающие стандартный трафик сети.

Узлы сети соединяются между собой каналами связи, параметры которых можно изменять. Примеры базовых параметров: `delay` (задержка распространения сигнала), `datarate` (скорость передачи данных). Стандартный узел сети состоит из следующих модулей: `ppp` — отвечает за канальный уровень; `networkLayer` — отвечает за сетевой уровень; `pingApp` — отвечает за приложения, связанные с ICMP протоколом; `tcp` — модуль, обслуживающий протокол TCP; `udp` — модуль, обслуживающий протокол UDP; `tcpApp[0]` — при-

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ложение TCP; `notificationBoard` — модуль для регистрации событий, происходящих на узле; `interfaceTable` — содержит таблицу сетевых интерфейсов; `routingTable` — содержит таблицу маршрутизации; `filterTable` — содержит таблицу правил фильтрации.

Приложения (в том числе и агенты) устанавливаются на узлы, подключаясь к соответствующим модулям протоколов.

При проектировании и реализации агентов были использованы элементы абстрактной архитектуры FIPA [58]. Основная идея такого представления заключается в обеспечении взаимодействия агентов и возможности их повторного использования. Такое описание позволяет увидеть взаимосвязи между основными элементами многоагентной системы.

Для агентов в разрабатываемой системе были использованы следующие элементы абстрактной архитектуры: язык коммуникаций, транспортный и сетевой уровни, каталог агентов. Для всех агентов была необходима реализация языка взаимодействия и транспортного уровня для передачи сообщений. Для агентов «мастер» и «детектор», координирующих работу агентов в своих командах, необходим также каталог агентов. Для демона необходима реализация двух транспортных модулей: для осуществления коммуникаций и для атаки. Для агента фильтрации необходима реализация сетевого уровня для возможности применения правил фильтрации. Агентам «сенсор» и «сэмплер» также необходим сетевой уровень для обработки и сбора данных в целях построения модели нормального трафика.

Агенты устанавливаются в среду моделирования с помощью подключения к транспортному и сетевому уровням OMNeT++ INET Framework (рис. 5, 6).

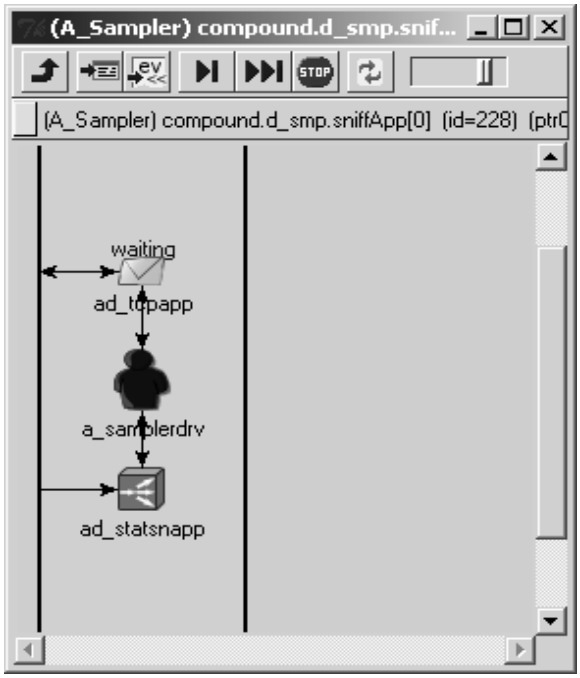

Рис. 5. Представление структуры агента «сэмплер».

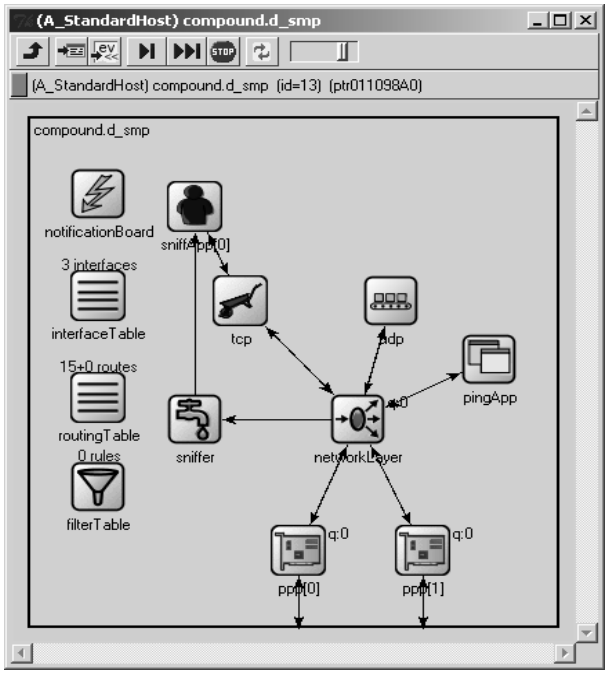

Рис. 6. Агент «сэмплер».

На рис. 5 изображено обобщенное представление структуры агента «сэмплер». Сэмплер включает в себя транспортный уровень (изображен в виде сообщения), необходимый для коммуникации с другими агентами, сетевой уровень (изображен в виде голубого кубика) для сбора данных по трафику и ядро

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

агента (представлен в виде синего образа фигурки человека). Последнее включает язык коммуникаций, базу знаний и обработчики сообщений от соседних модулей. Представление установки агента «сэмплер» в среду моделирования показано на рис. 6. Видно, что агент подключается к узлу сети через модуль `tcp`, обслуживающий протокол TCP, и модуль `sniffer` анализатора сетевых пакетов.

Сеть для многоагентного моделирования состоит из трех подсетей: подсеть защиты, где расположена команда защиты; промежуточная подсеть, в которой расположены узлы, создающие типовой трафик в сети, в том числе к защищаемому узлу; подсеть атаки, где расположена команда атаки.

Подсеть защиты (рис. 4, вверху) состоит из 4 узлов, на которых установлены 4 агента (детектор, сэмплер, фильтр, расследования), и защищаемого узла, на котором установлен атакуемый сервер. Агенты и сервер представляют собой приложения, функционирующие на соответствующих узлах. IP-адреса узлов выдаются автоматически. Остальные параметры приложений необходимо задать перед моделированием. Сервер установлен на узле `d_srv`. Задается порт для взаимодействия и время задержки ответа клиентам. Сервер входит в INET Framework. Детектор размещен на узле `d_det`. Назначается адрес защищаемого сервера, порт для командного взаимодействия, интервал опроса сенсоров, максимально допустимая скорость передачи данных к серверу (*BPS*). Сенсор расположен на узле `d_firewall` (на входе в подсеть сервера). Задается собственный порт, IP-адрес и порт детектора для командного взаимодействия. Фильтр размещен на узле `d_r` (маршрутизатор). Назначается собственный порт, IP-адрес и порт детектора для командного взаимодействия. Агент расследования установлен на узле `d_inv` (во внешней сети). Задается собственный порт, IP-адрес и порт детектора для командного взаимодействия.

Промежуточная подсеть (рис. 4, в центре) состоит из *N* узлов `i_cli[…]` с типовыми клиентами, соединенных с маршрутизатором `i_r`. Клиент входит в INET Framework. Количество узлов *N* определяется параметром моделирования. Задаются следующие параметры клиентов: адрес и порт сервера, время начала работы, количество и размер запросов при соединении с сервером, размер ответа, время подготовки ответа и интервал бездействия.

Подсеть атаки (рис.4, внизу и в центре) включает *M* узлов `i_cli[…]`, на которых размещаются демоны, и один узел с мастером. Они соединены с маршрутизатором `i_r`. Количество узлов *M* задается параметром моделирования. Мастер имеет следующие параметры: порт для командного взаимодействия, адрес и порт цели атаки, время начала атаки, ее интенсивность (пакетов в секунду). Для демона задаются собственный порт, IP-адрес и порт мастера для командного взаимодействия.

6. Основные параметры моделирования

Для моделирования используется генератор топологий сетей, максимально приближенных к реально существующим в Internet. Были заданы следующие параметры топологии опорной сети: минимальное количество связей у каждого узла — 2, количество узлов — 10, параметр вероятностного распределения $\gamma = 2{,}25$. Структура сформированной сети для проведения моделирования, а также размещение агентов команд атаки и защиты показаны на рис. 4.

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Маршрутизаторы соединены между собой волоконно-оптическими каналами связи со следующими параметрами: `delay 1us` (задержка распространения сигнала — 1 мс); `datarate 512*1000000` (скорость передачи данных — 512 Мбит).

Остальные узлы соединены Ethernet каналами связи с параметрами: `delay 0.1us` (задержка распространения сигнала — 0,1 мс); `datarate 10*1000000` (скорость передачи данных — 10 Мбит).

10 клиентов подключены случайным образом к маршрутизаторам опорной сети. Защищаемый сервер — `d_srv`.

Базовые параметры клиентов сети: `connectAddress="d_srv"` (адрес сервера); `connectPort=80` (порт сервера); `startTime=exponential(5)` (время начала работы является случайной величиной с экспоненциальной функцией распределения и средним значением 5 секунд); `numRequestsPerSession = 1` (количество запросов за соединение к серверу); `requestLength = truncnormal(350,20)` (размер запроса является случайной величиной с нормальной функцией распределения, средним значением 350, дисперсией 20 бит); `replyLength = exponential(2000)` (размер запроса является случайной величиной с экспоненциальной функцией распределения и средним значением 2000 бит); `thinkTime=truncnormal(2,3)` (время обработки является случайной величиной с нормальной функцией распределения, средним значением 2 секунд, дисперсией 3 секунды (округляется до положительной величины)); `idleInterval=truncnormal(36,12)` (время простоя является случайной величиной с нормальной функцией распределения, средним значением 36 секунд, дисперсией 12 секунд); `reconnectInterval=30` (интервал соединения с сервером – 30 секунд).

Структура команды защиты: детектор — установлен на узле `d_det`, сэмплер — `d_firewall`, фильтр — `d_r`, агент расследования — `d_inv`.

Базовые параметры команды защиты: `target_host="d_srv"` (адрес защищаемого сервера); `detector_ip="d_det"` (адрес детектора для командного взаимодействия); `detector_port="2000"` (порт детектора для взаимодействия).

Параметры сэмплера: `dt=10` (интервал для методов SIPM и BPS); `tshift=3` (сдвиг интервала для методов SIPM и BPS).

Команда атаки включает 10 демонов, установленных на стандартные узлы сети, которые подключены случайным образом к маршрутизаторам опорной сети, а также агента-мастера, расположенного на узле `a_srv`.

Начальные параметры команды атаки: `a_n=10` (количество демонов – 10 штук); `master_ip="a_srv[0]"` (адрес мастера для командного взаимодействия); `port=2000` (порт для командного взаимодействия); `ad_udpapp.port=2001` (порт демонов для посылки пакетов атаки); команда бездействует.

7. Пример сценария моделирования

Режим обучения. Суть режима обучения заключается в том, чтобы составить модель типового для исследуемой сети трафика. Клиенты обращаются к серверу, а он им отвечает. В это время сэмплер регистрирует эти запросы и использует их для формирования параметров методов SIPM, HCF и BPS.

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Во время обучения можно наблюдать за изменением моделей трафика для каждого из предложенных методов (рис. 7, 8).

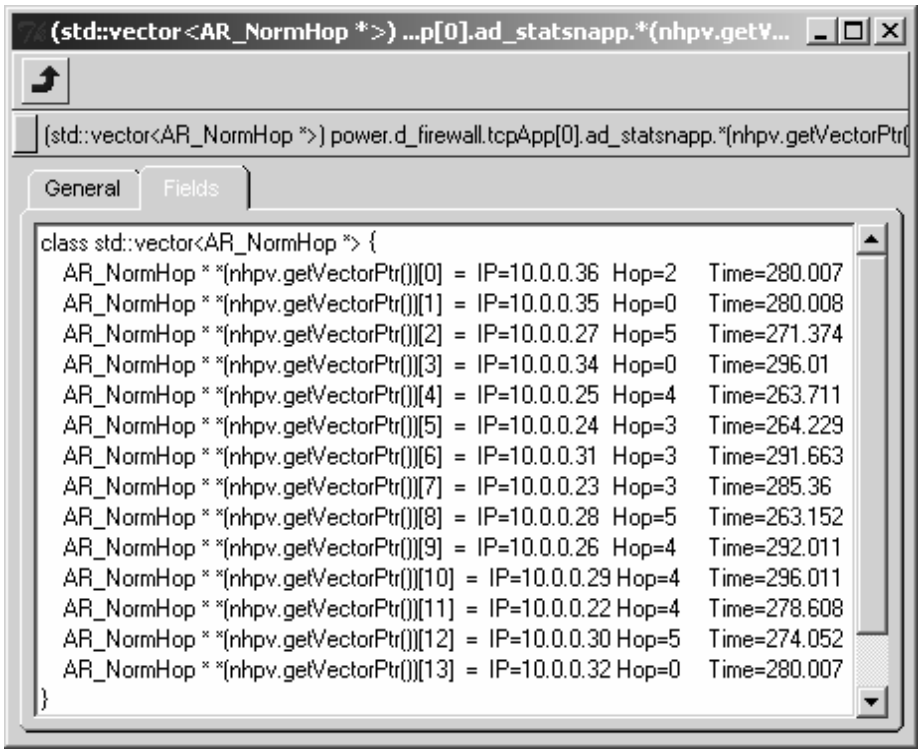

Рис. 7. Список узлов, обращавшихся к серверу, и хопов между ними и сервером после 300 секунд обучения.

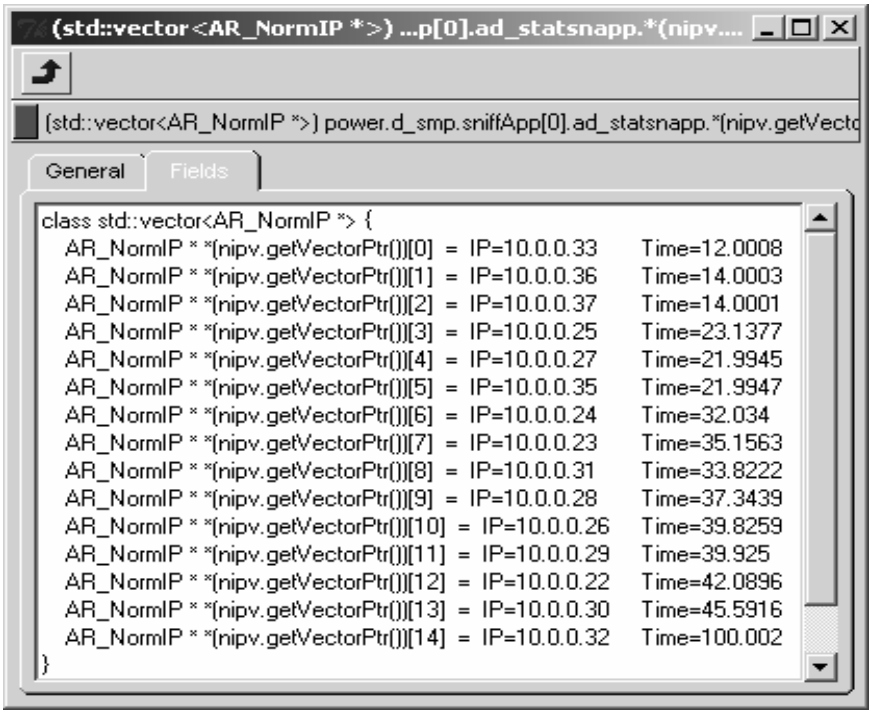

Рис. 8. Список узлов, обращавшихся к серверу и признанных легитимными клиентами после 300 секунд обучения.

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

На рис. 7 приведен список узлов, обращавшихся к серверу, и скачков (хопов; в общем случае равно количеству пройденных пакетом маршрутизаторов) до них после 300 секунд обучения, а также время последнего обращения. Как уже упоминалось, количество хопов вычисляется с помощью поля TTL (Time To Live) пакета.

На рис. 8 изображен список обращавшихся к серверу и признанных легитимными клиентами после 300 секунд обучения. Здесь также можно увидеть, что в интервале от 0 до 50 секунд было много новых для сервера адресов, что соответствует графику на рис. 9.

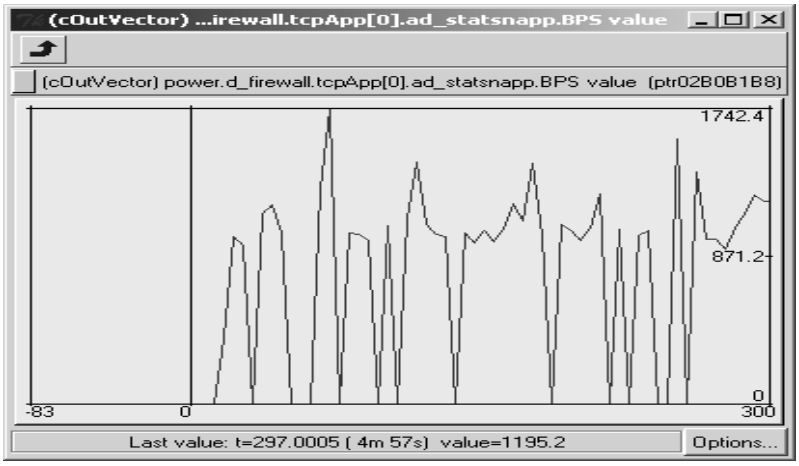

Рис. 9. Изменение параметра BPS.

На рис. 8 представлен график изменения максимального параметра BPS за интервал 10 секунд со сдвигом 3 секунды через 300 секунд после начала обучения. Максимальное значение было зарегистрировано в районе 100 секунд, оно равно 1742.4 бит/с. Можно также увидеть значения BPS для адресов клиентов, обратившихся за заданный интервал к серверу.

Режим противодействия. Сценарий реализуется при такой же конфигурации, как была использована при обучении. Главное отличие заключается в том, что теперь задействована команда атаки.

Параметры команды атаки: `target_ip=″d_srv″` (цель атаки — сервер `d_srv`); `target_port=″2001″` (порт цели атаки); `t_ddos=300` (время начала атаки); `attack_rate=5` (интенсивность атаки в пакетах в секунду); `ip_spoofing=″no″` (атака без подмены адреса отправителя).

После начала моделирования (в интервале от 0 до 300 секунд) клиенты начинают посылать запросы серверу, а он на них отвечать. Таким образом происходит генерация обычного сетевого трафика.

Через некоторое время после начала моделирования происходит составление команды защиты. Агенты расследования, сэмплер и фильтр соединяются с детектором и посылают ему сообщения о своей работоспособности. Детектор заносит данные о них в свою память. Аналогичным образом происходит формирование команды атаки: с мастером соединяются демоны и сообщают о своей работоспособности.

После формирования команды защиты, она начинает свое функционирование. Сэмплер собирает данные по сетевому трафику и сравнивает их с «модельными» данными, полученными в режиме обучения. Адреса, от которых ис-

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ходят аномалии, передаются детектору каждые n секунд (в описываемом сценарии моделирования $n = 60$). Детектор принимает решение о том, происходит атака или нет, и отсылает фильтру и агенту расследования IP-адреса подозрительных узлов.

Графики изменения пропускной способности канала на входе в защищаемую сеть до (темный) и после фильтра (светлый) показаны на рис. 10 (до 300 с. можно увидеть величину трафика, создаваемого клиентами).

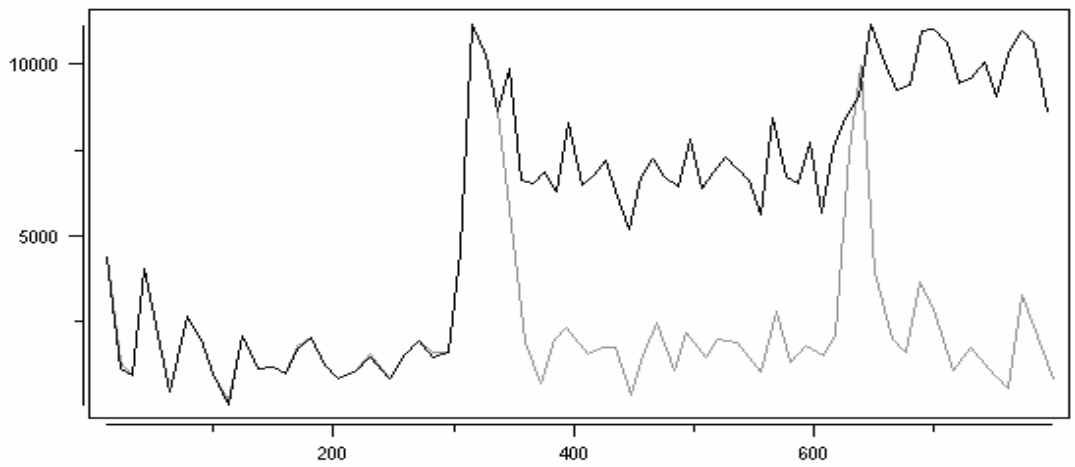

Рис. 10. Графики изменения трафика на входе в защищаемую сеть (зависимость бит/с от времени) до (темный) и после фильтра (светлый).

Через 300 секунд после начала моделирования команда атаки приступает к атакующим действиям. Мастер опрашивает всех известных ему демонов. Затем, работоспособным демонам отсылается команда атаки: адрес и порт цели атаки, интенсивность (распределенная между всеми демонами) и способ подмены адреса отправителя. Получив команду, демоны приступают к посылке пакетов атаки (рис. 10, отметка 300 секунд).

Через некоторое время, сэмплер по методу BPS определяет узлы, которые создают аномально большой трафик. Детектор получает от него эти адреса и пересылает их фильтру и агенту расследования. Фильтр применяет правила фильтрации, и пакеты от выбранных узлов начинают отбрасываться (рис. 10, отметка от 400 до 600 секунд, светлый график).

Агент расследования пытается проследить указанные узлы и обезвредить агентов атаки, установленных на них. Ему удается обезвредить 4 демона. В окне структуры компьютерной сети над обезвреженными демонами появляется надпись «Defeated». Однако остальные агенты-демоны продолжают атаку (рис. 10, после 400 и до 600 секунд, темный график).

Мастер, видя, что ряд демонов был обезврежен, перераспределяет интенсивность атаки между оставшимися демонами и задает метод подмены адресов — «случайный». Детектор, определяя, что атака не прекращается, посылает сэмплеру запрос на применение нового механизма защиты: SIPM. Благодаря этому пакеты атакующих стали отбрасываться (рис. 10, после 700 секунд, светлый график), и в защищаемой подсети установился порядок. Однако агенту расследования не удается проследить оставшихся атакующих, и за пределами подсети, атака продолжается.

Труды СПИИРАН. 2006. Вып. 3. Т. 1. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2006. Issue 3. V. 1. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

8. Заключение

Основные результаты работы заключаются в разработке базовых идей по многоагентному моделированию механизмов атаки и защиты от DDoS и реализации среды моделирования для воплощения этих идей. Среда моделирования реализована на C++ и OMNeT++. Она позволяет моделировать большой спектр реальных атак DDoS и механизмов защиты от них. Был проведен ряд экспериментов для исследования противостояния различных механизмов защиты нескольким типам атак.

Результаты работы могут являться необходимым базисом для разработки целого класса интеллектуальных систем, позволяющих на основе исследовательского компьютерного моделирования осуществлять создание и анализ систем защиты информации. Полученные в ходе исследования результаты, в том числе разработанная программная среда моделирования, могут быть использованы для проектирования и анализа систем защиты информации современных компьютерных сетей (выполняемых проектировщиками и администраторами защиты компьютерных сетей), а также для исследования как существующих, так и перспективных компьютерных атак и механизмов защиты от них. Другим важным направлением использования полученных результатов является обучение специалистов в области защиты информации.

Дальнейшее развитие работы связано с разработкой формальных моделей кибернетического противоборства в Internet, улучшением среды моделирования (в том числе за счет реализации большего количества типов атак, механизмов защиты, сценариев взаимодействия), проведением экспериментов для сравнения и оценки эффективности систем защиты.

Работа выполнена при финансовой поддержке РФФИ (проект №04-01-00167), программы фундаментальных исследований ОИТВС РАН (контракт №3.2/03) и при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза POSITIF (контракт IST-2002-002314).

Литература

1. Nomad Mobile Research Centre [Электронный ресурс] // <http://www.nmrc.org> (по состоянию на 24.03.2006).
2. *Mirkovic J., Dietrich S., Dittrich D., Reiher P.* Internet Denial of Service: Attack and Defense Mechanisms. Prentice Hall PTR, 2004. 400 p.
3. *Городецкий В. И., Котенко И. В.* Концептуальные основы стохастического моделирования в среде Интернет // Труды Института системного анализа РАН. Москва, 2005. С. 20–35.
4. *Котенко И. В.* Многоагентные модели противоборства злоумышленников и систем защиты в сети Интернет // Математика и безопасность информационных технологий: Материалы конференции в МГУ. М., 2005. С. 44–58.
5. *Котенко И. В., Карсаев О. И.* Использование многоагентных технологий для комплексной защиты информации в компьютерных сетях // Известия ТРТУ. 2001. № 4. С. 12–18.
6. *Gorodetski V., Kotenko I., Karsaev O.* Framework for Ontology-based Representation of Distributed Knowledge in Multiagent Network Security System // Proceedings of the 4th World Multi-conference on Systems, Cybernetics and Informatics (SCI-2000). Orlando, 2000. P. 44–56.
7. *Whittaker G. M.* Asymmetric Wargaming: Toward A Game Theoretic Perspective. MITRE, 2000. 86 p.
8. *Новиков Д. А., Чхартишвили А. Г.* Рефлексивные игры. М.: СИНТЕГ, 2003. 212 с.
9. *Чхартишвили А. Г.* Теоретико-игровое моделирование информационного управления в активных системах // Человеческий фактор в системах управления: Сб. науч. работ. Москва, 2005. С. 17–40.

10. *Городецкий В. И., Котенко И. В.* Командная работа агентов-хакеров: применение многоагентной технологии для моделирования распределенных атак на компьютерные сети // VIII Национальная конференция по искусственному интеллекту с международным участием: Труды конференции. М.: Физматлит, 2002. С. 711–720.
11. *Perumalla K. S., Sundaragopalan S.* High-Fidelity Modeling of Computer Network Worms // Technical Report GIT-CERCS-04-23. Center for Experimental Research in Computer Science. Georgia Institute of Technology, 2004. 40 p.
12. *Cohen P., Levesque H.J.* Teamwork. Nous, 1991. 67 p.
13. *Grosz B., Kraus S.* Collaborative Plans for Complex Group Actions // Artificial Intelligence. 1996. Vol. 86. P. 33–50.
14. *Tambe M.* Towards flexible teamwork // Journal of AI Research. 1997. Vol. 7. P. 50–75.
15. *Jennings N. R.* Controlling cooperative problem solving in industrial multi-agent systems using joint intentions // Artificial Intelligence. 1995. Vol. 75, no. 2. P. 120–134.
16. *Martin D., Cheyer A., Moran D.* The open agent architecture: A framework for building distributed software systems // Applied Artificial Intelligence. 1999. Vol. 13, no. 2. P. 141–154.
17. *Yen J., Fan X., Sun S., Wang R., Chen C., Kamali K., Miller M., Volz R. A.* On Modeling and Simulating Agent Teamwork in CAST // Proceedings of the Second International Conference on Active Media Technology. Chongqing, 2003. P. 56–78.
18. *Giampapa J. A., Sycara K.* Team-Oriented Agent Coordination in the RETSINA Multi-Agent System // Technical Report CMU-RI-TR-02-34. Robotics Institute, Carnegie Mellon University, 2002. 75 p.
19. *Zachary W. W., Mentec J. L.* Modeling and simulating cooperation and teamwork // Military, government, and aerospace simulation. 2000. Vol. 32. P. 1216–1235.
20. *Котенко И. В., Станкевич Л. А.* Командная работа агентов в реальном времени // Новости искусственного интеллекта. 2003. № 3. С. 21–34.
21. *Charniak E., Goldman R. P.* A Bayesian Model of Plan recognition // Artificial Intelligence. 1993. Vol. 64, no. 1. P. 354–361.
22. *Kautz H., Allen J. F.* Generalized plan recognition // Proceedings of the Fifth National Conference on Artificial Intelligence. Philadelphia, 1986. P. 241–262.
23. *Vilain M.* Getting Serious about Parsing Plans: A Grammatical Analysis of Plan Recognition // Proceedings of the Eighth National Conference on Artificial Intelligence. Cambridge, MA, 1990. P. 116–126.
24. *Wellman M. P., Pynadath D. V.* Plan Recognition under Uncertainty // Proceedings of the Tenth National Conference on Artificial Intelligence. Cambridge, MA, 1997. P. 251–260.
25. *Goldman R. P., Geib C. W., Miller C. A.* A New Model of Plan Recognition // Proceedings of the 1999 Conference on Uncertainty in Artificial Intelligence. Washington, DC, 1999. P. 15–36.
26. *Geib C. W., Goldman R. P.* Plan recognition in intrusion detection systems // Proceedings of DARPA Information Survivability Conference and Exposition. Hilton Head, SC, 2001. P. 334–368.
27. *Котенко И. В.* Распознавание планов агентов-хакеров при обнаружении компьютерных атак // Труды Международных научно-технических конференций «Интеллектуальные системы (IEEE AIS'04)» и «Интеллектуальные САПР (CAD-2004)». М.: Физматлит, 2004. С. 47–58.
28. *Лефевр В. А.* О самоорганизующихся и саморефлексивных системах и их исследовании // Проблемы исследования систем и структур: Сб. науч. работ. М., 1965. С. 18–25.
29. *Лефевр В. А.* Рефлексия. М.: Когито-Центр, 2003. 512 с.
30. *Лепский В. Е., Рапуто А. Г.* Моделирование и поддержка сообществ в Интернет. М.: Институт психологии РАН, 1999. 230 с.
31. *Стогний А. А., Кондратьев А. И.* Теоретико-игровое информационное моделирование в системах принятия решений. Киев: Наукова думка, 1986. 150 с.
32. *Дружинин В. В., Конторов Д. С., Конторов М. Д.* Введение в теорию конфликта. М.: Радио и связь, 1989. 311 с.
33. *Gorodetski V., Kotenko I.* Attacks against Computer Network: Formal Grammar-based Framework and Simulation Tool // Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection. Innsbruck, 2002. Vol. 2516. P. 1456–1463.
34. *Csuhaj-Varjú E.* COLONIES: a multi-agent approach to language generation // Proceedings of ECAI'96 Workshop on Extended Finite State Models of Language / Ed. A. Kornai. Budapest, 1996. P. 31–44.
35. *Kelemen J.* Colonies: grammars of reactive systems // Proceedings of the Artificial Intelligence Computing Research Symposium. Singapore, 1997. P. 211–245.

36. *Paun Gh., Salomaa A.* Grammatical models of multi-agent systems. Amsterdam: Gordon and Breach, 1999. 148 p.
37. *Anchorena S., Cases B.* Modeling chaotic series by simple eco-grammar systems with reproduction, death, and maturation. Grammars, 2003. 190 p.
38. *Городецкий В., Котенко И., Карсаев О.* Обучение и мета-обучение в многоагентных системах на примере задачи обнаружения вторжений в компьютерных сетях // 4-й международный семинар по прикладной семиотике, семиотическому и интеллектуальному управлению ASC/IC'99: Сборник трудов. М.: ПАИМС, 1999. С. 24–35.
39. *Редько В. Г.* Эволюционная кибернетика. Тезисы курса лекций [Электронный ресурс] // <http://www.keldysh.ru/pages/BioCyber/Lectures.html> (по состоянию на 24.03.2006).
40. *Back T., Fogel D. B., Michalewicz Z.* Evolutionary computation. Vol. 1. Basic algorithms and operators. Institute of Physics Publishing. 2000. 351 p.
41. *Back T., Fogel D. B., Michalewicz Z.* Evolutionary computation. Vol. 2. Advanced algorithms and operators. Institute of Physics Publishing. 2000. 312 p.
42. *Емельянов В. В., Курейчик В.В., Курейчик В. М.* Теория и практика эволюционного моделирования. М.: Физматлит, 2003. 154 с.
43. *Алгулиев Р. М.* Методы синтеза адаптивных систем обеспечения информационной безопасности корпоративных сетей. М.: УРСС, 2001. 252 с.
44. *Gu D., Yang E.* Multiagent Reinforcement Learning for Multi-Robot Systems: A Survey // Technical Report of the Department of Computer Science, University of Essex, CSM-404. 2004. 124 p.
45. *Котенко И. В.* Многоагентные технологии для анализа уязвимостей и обнаружения вторжений в компьютерных сетях // Новости искусственного интеллекта. 2004. № 1. С. 12–25.
46. *Kotenko I.* Agent-Based Modeling and Simulation of Cyber-Warfare between Malefactors and Security Agents in Internet // Proceedings of the 19th European Simulation Multiconference Simulation in wider Europe. Riga, 2005. P. 45–56.
47. *Kotenko I., Stankevitch L., Akhapkin S.* Time-constrained Teamwork // Proceedings of China-Russia Bilateral Conference on Intelligent Information Processing. Beijing, 2002. P. 29–37.
48. *Kotenko I., Ulanov A.* Multiagent modeling and simulation of agents' competition for network resources availability // Proceedings of the Second International Workshop on Safety and Security in Multiagent Systems. Utrecht, 2005. P. 47–61.
49. *Jin C., Wang H., Shin K. G.* Hop-count filtering: An effective defense against spoofed DDoS traffic // Proceedings of the 10th ACM Conference on Computer and Communications Security. Washington, DC, 2003. P. 76–89.
50. *Peng T., Leckie C., Kotagiri R.* Proactively Detecting DDoS Attack Using Source IP Address Monitoring // Networking. 2004. P. 21–29.
51. NS-2 homepage [Электронный ресурс] // <http://www.isi.edu/nsnam/ns/> (по состоянию на 24.03.2006).
52. OMNeT++ homepage [Электронный ресурс] // <http://www.omnetpp.org/> (по состоянию на 24.03.2006).
53. SSFNet homepage [Электронный ресурс] // <http://www.ssfnet.org> (по состоянию на 24.03.2006).
54. J-Sim homepage [Электронный ресурс] // <http://www.j-sim.org> (по состоянию на 24.03.2006).
55. *Kotenko I. V., Ulanov A. V.* Agent-based simulation of DDOS attacks and defense mechanisms // Journal of Computing. 2005. Vol. 4, no. 2. P. 16–37.
56. *Kotenko I. V., Ulanov A. V.* The Software Environment for multi-agent Simulation of Defense Mechanisms against DDoS Attacks // Proceedings of The International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Vienna, 2005. P. 67–81.
57. *Gorodetski V., Karsayev O., Kotenko I., Khabalov A.* Software Development Kit for Multi-agent Systems Design and Implementation // From Theory to Practice in Multi-Agent Systems. Second International Workshop of Central and Eastern Europe on Multi-Agent Systems, CEEMAS 2001 Cracow, Poland, September 26-29, 2001: Revised Papers / Series: Lecture Notes in Computer Science. Subseries: Lecture Notes in Artificial Intelligence. Vol. 2296 / Dunin-Keplicz, Barbara; Nawarecki, Edward (Eds.). Springer, 2002. P. 1556–1571.
58. FIPA [Электронный ресурс] // <http://www.fipa.org> (по состоянию на 24.03.2006).