Мультиагентное моделирование сетевой атаки типа DDoS

И.В. Георгица, С.А. Гончаров, В.А. Мохов

В последнее время наблюдается рост количества распределенных атак на глобальные компьютерные сети. Значительная часть этих атак направлена на нарушение доступности или «распределенный отказ в обслуживании» (Distributed Denial of Service, DDoS). Атака заключается в перегрузке хоста или сетевого ресурса посредством наполнения системы большим количеством сетевых пакетов [1]. Эти атаки реализуются множеством программных агентов («ботов» или «демонов»), размещенных на хостах, которые злоумышленник скомпрометировал ранее. Реализация этих атак может привести не только к выходу из строя отдельных хостов и служб, но и остановить работу корневых DNS-серверов и вызвать частичное или полное прекращение работы Интернета. В связи с критичностью и нетривиальностью атак данного класса, построение эффективных средств защиты от них представляет собой сложную научно-техническую проблему.

На практике достаточно проблематично осуществить проверку и оценку применения того или иного механизма защиты. Исследование на основе реальных сетей трудно реализуемо практически вследствие следующих причин: необходимо либо располагать большим выделенным фрагментом сети, где можно проводить эксперименты, либо использовать для экспериментов сети реальных Интернет-провайдеров [2]. Следует отметить, что атаки DDoS создают большую нагрузку на сеть, вплоть до полного отказа в обслуживании, что приводит к выходу эксперимента из-под контроля, и даже распространению атак в Интернет. При этом важные условия научного эксперимента, такие как повторяемость и контролируемость, не могут быть соблюдены. Вследствие описанных причин для исследования механизмов защиты от атак DDoS, а также для получения возможности одновременного рассмотрения и оценки нескольких

альтернативных вариантов решений [3] авторами используются инструменты имитационного моделирования, а именно многоагентное (мультагентное) моделирование, позволяющее представить изучаемые процессы в виде совокупности автономных агентов и взаимодействий между ними [4, 5].

В общем виде представим модель DDoS-атаки следующим образом:

$$S = (A, ACT, E, H, R),$$

где A – множество агентов с заданным набором действий ACT, активных на множестве запущенных платформ E на хостах H, имеющих доступ к целевому серверу R.

Множество агентов на хостах, имеющих доступ к сервису, представим так:

$$A = \{CA, VF\},$$

где *CA* (*Control Agent*) – агент, который дает команду для атаки, *MA* (*Mobile Agent)* – агент (непосредственно атакующий), который посылает запрос заданного вида на сервер, по команде *CA*. При этом агента *MA* представим следующим образом:

$$MA = (ID, ST, Mact),$$

где *ID* – уникальное имя агента, *ST* – множество состояний, *Mact* – множество действий агента.

Для программной реализации модели была определена платформа мультиагентного программирования JADE (Java Agent Development Framework) [6]. Платформа написана на языке программирования Java с использованием Java RMI, Java CORBA IDL, Java Serialization и Java Reflection API и упрощает разработку мультиагентных систем благодаря использованию FIPA-спецификаций и инструментов, которые поддерживают большинство фаз их отладки и развертывания [7].

Следует отметить, что для разгрузки управляющего агента *CA* от «рутинной работы» в модели был дополнительно реализован агент *Dispatcher*, предназначенный для выполнения задач подсчета времени и записи необходимых параметров в файл.

Реализация обмена сообщениями выполнена путём формирования диалога между агентами на языке ACL (для чего был определён перечень допустимых сообщений для каждого агента системы [8,9]). Схема коммуникации агентов представлена на рис.1.

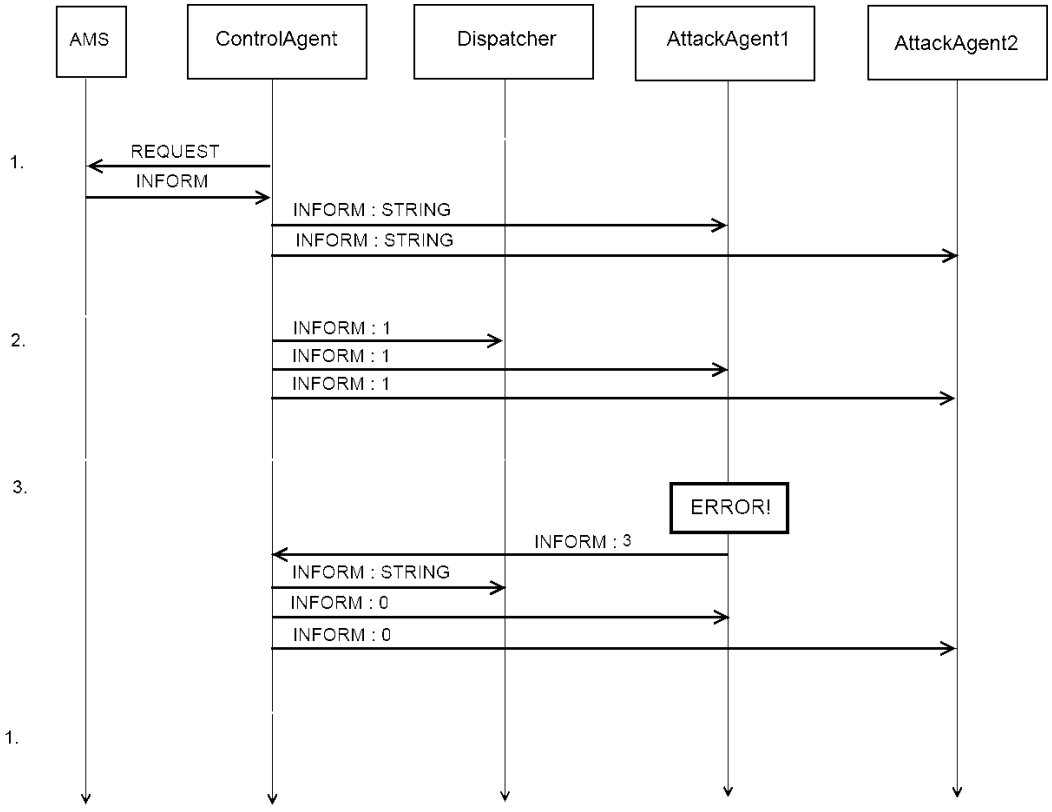

Рис. 1. – Коммуникация агентов посредством обмена сообщениями

Один цикл работы системы представляет собой три этапа. На первом этапе управляющей агент обращается с запросом REQUEST к системному агенту AMS платформы, чтобы получить список всех агентов и в последующем проинформировать их, отправив сообщения типа INFORM, содержащие адрес целевого хоста. На втором этапе управляющей агент посылает сообщения типа INFORM со значением «1» атакующим агентам (на рис.1 они обозначены, как агенты AttackAgent1 и AttackAgent2). На третьем этапе управляющий агент ждет, пока не будет принято входящее сообщение типа INFORM со значением «3», которое означает отказ сервера в доступе, после чего отправляет строку типа INFORM диспетчеру и символ «0» всем атакующим агентам. После этого этапы повторяются в зависимости от

количества циклов работы системы. Результаты моделирования представлены на рис.2, откуда видно, как ведет себя сервер при объеме буфера – 15 байт.

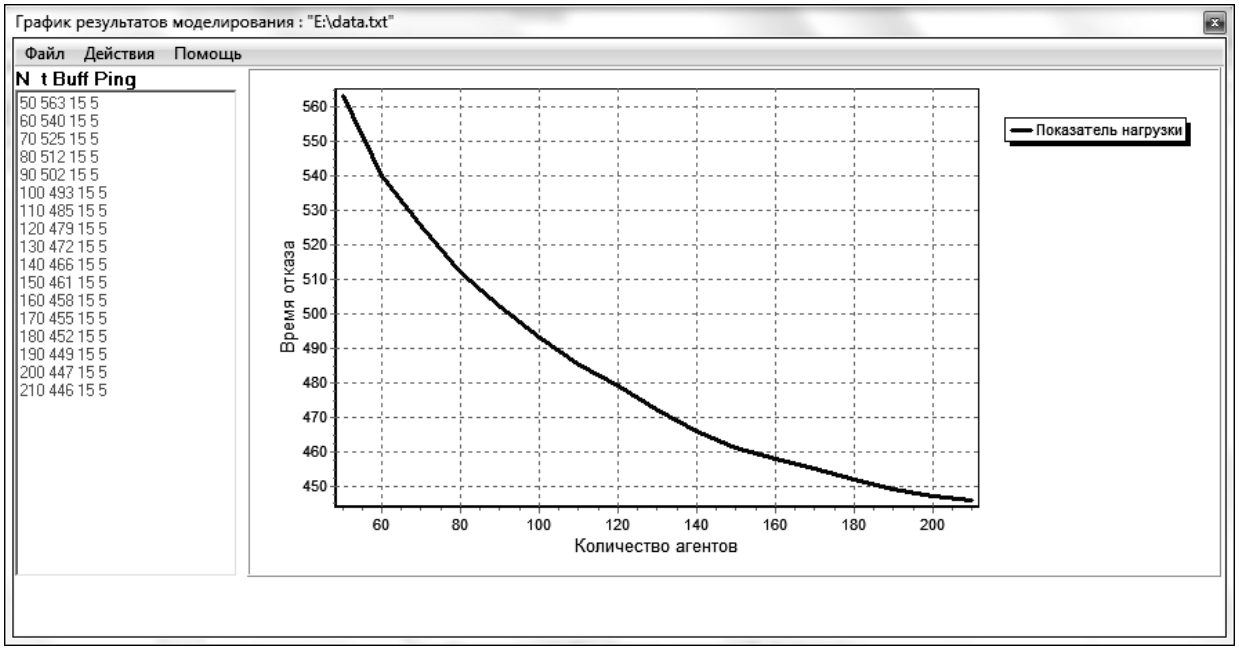

Рис. 2. – Графическое представление результатов моделирования

Варьируя размером буфера и временем отклика агента-сервера, разработанная модель позволяет для конкретного информационного сервера экспериментальным путем получить зависимость [10]

$$t = f(N, Buff, ping),$$

определяющую время от начала атаки до отказа в доступе к серверу (t), в зависимости от количества атакующих агентов (N), размера входного буфера ($Buff$) и времени отклика ($ping$).

Анализ полученных графиков позволяет сделать очевидный вывод о том, что любое увеличение объёма буфера лишь отдалит отказ в доступе к серверу, но не поможет его избежать. В этом случае для предотвращения перегрузки сервера авторами предлагается использовать метод «flood-gate». Основную функцию этого метода должен выполнять скрипт (служба), который дополнительно располагается на сервере. Предполагается, что скрипт контролирует нагрузку сервера и, если она превосходит допустимую,

принимает меры по её ограничению. В данном случае возникает задача определения уровня нагрузки, при превышении которого имеет смысл активировать скрипт.

Отказ сервера есть не что иное, как переполнение буфера входящих запросов на серверной части. Следовательно, необходимо подобрать такой допустимый уровень заполнения буфера, при котором сервер будет близок к перегрузке, но всё ещё будет продолжать функционировать.

Далее рассмотрим сущность методики для решения указанной задачи.

Результатом предлагаемой методики должна быть безотказная работа сервера в отношении атак типа DDoS. Вариант графика с демонстрацией ограничений нагрузки представлен на рис.3.

На этом рисунке E_{max} показывает максимальную нагрузку, при которой сервер перестает отвечать на запросы, а E' – нагрузку, которую допускает скрипт и при которой сервер продолжает обрабатывать поток трафика.

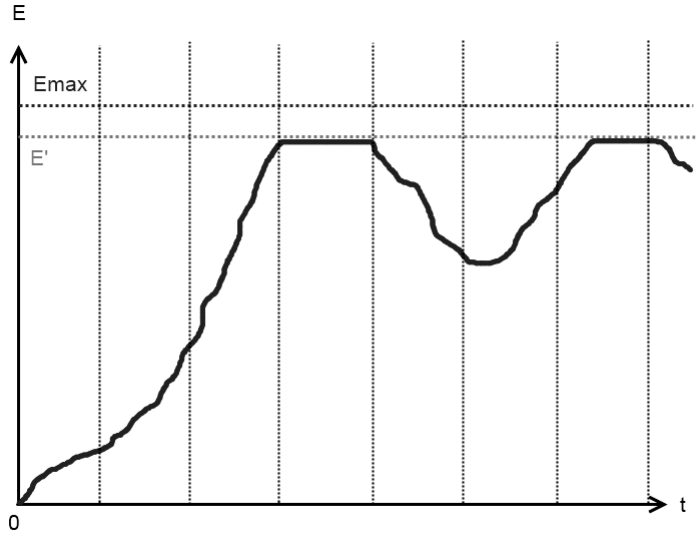

Рис. 3. – Ограничение нагрузки скриптом «flood-gate».

Получение подобного результата основывается на обработке результатов экспериментов с мультиагентной моделью атаки, описанной выше. Для этого сначала необходимо отметить все точки A_i (моменты отказа сервера, в которых нагрузка достигает максимального значения) для

соответствующих точек n_i (количество агентов) и t_i, (длительность временного интервала работы сервера), как показано на рис.4.

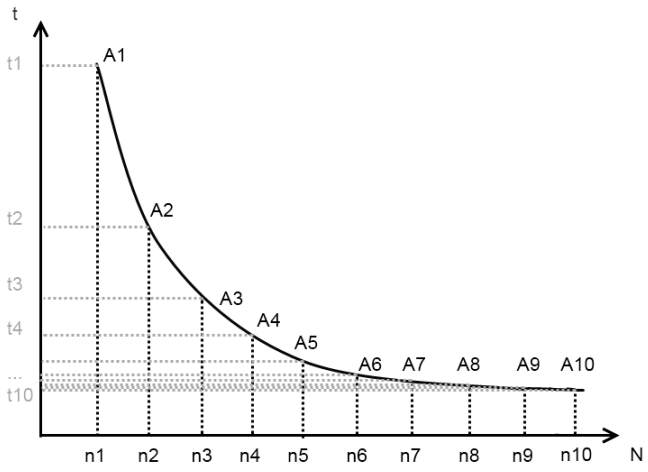

Рис. 4. – График зависимости времени отказа от количества агентов

Чтобы график на рис.4 использовать для получения графика на рис.3. необходимо выполнить несколько преобразований, с помощью которых будет подобрано оптимальное значение E' для буфера сервера.

Сначала все точки, характеризующие время отказа сервера с графика рис. 4 переносятся на график рис. 3 на ось времени, как показано на рис. 5.

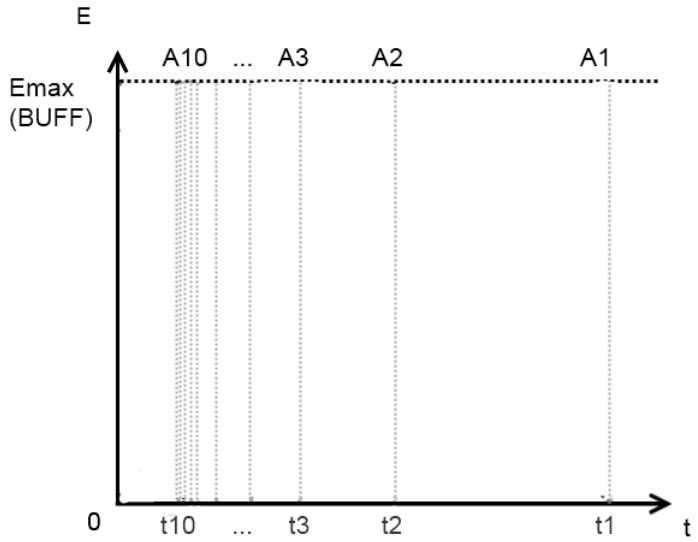

Рис. 5. – Перенос значений времени и проекция точек отказа сервера

Далее обозначается прямая E_{max}. Проецируются точки t_i на прямую E_{max} и получаются необходимые точки A_i. Затем совершается перенос графика

нагрузки с рис. 4 на каждый промежуток времени. Производится корректировка и масштабирование по оси абсцисс так, чтобы его начало было в точке *0*, а конец в точке A_i. Каждый из этих графиков будет характеризовать определенный характер роста нагрузки в зависимости от массовости распределенной атаки. Чтобы определить *E'*, проводится прямая *BC* из точки $t = t_1$, при $E = 0$ в точку $t = 0$, при E_{max}, согласно рис. 6. Точки пересечения x_i прямой *BC* с кривой графика, заключенного в промежутке $t = (0;t_{10})$, будут служить правилом задания для *E'*.

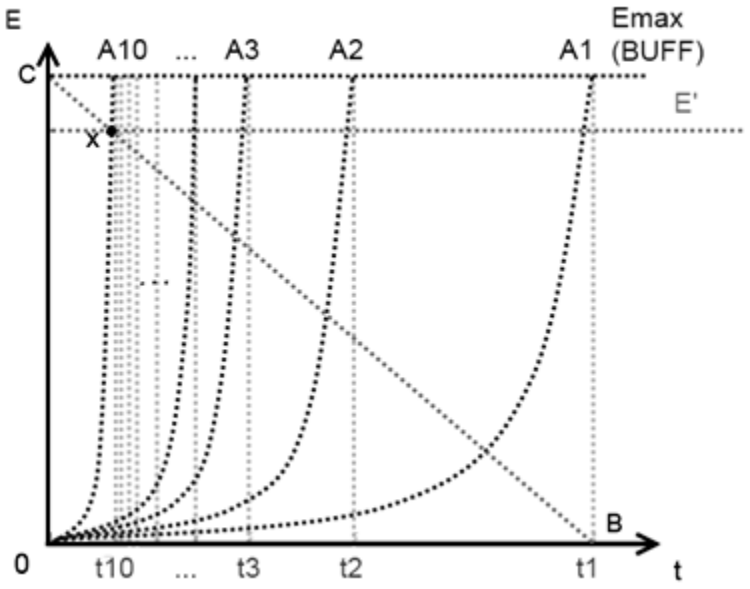

Рис. 6. Определение значения *E'*

Опуская перпендикуляр из точки x_i на ось ординат, определяется конкретное значение *E'* для настраиваемого сервера. Полученное таким образом значение нагрузки также определяет оптимальный момент активации скрипта Таким образом удерживается уровень нагрузки в пределах функционирования сервиса с максимальной нагрузкой наименьшей вероятностью краха системы от DDoS-атаки.

Таким образом, полученная методика позволяет выполнять настройку параметров защиты сетевых серверов на основе метода «flood-gate» в зависимости от технических характеристик аппаратуры по результатам

проведения экспериментов с разработанной мультиагентной моделью сетевой атаки типа DDoS.

Литература:

1. Касперски, К. Компьютерные вирусы изнутри и снаружи [Текст] / К. Касперски. – СПб.: Питер, 2006. – 527 с.

2. Колыхан, Н.В. Динамическое управление информационными потоками в телекоммуникационных сетях [Электронный ресурс] / Н.В. Колыхан, А.П. Самойленко // «Инженерный вестник Дона», 2008, № 2. – Режим доступа: http://www.ivdon.ru/magazine/archive/n2y2008/64 (доступ свободный) – Загл. с экрана. – Яз. Рус.

3. Синельщиков, А.В. Применение имитационного моделирования при анализе эффективности процессов перегрузки зерна [Текст] / А.В. Синельщиков, М.А. Гассельберг // Вестник Астраханского государственного технического университета. - 2011. - № 2. С. 31-36.

4. Лещев, В.А. Агентно-ориентированная технология проектирования [Текст] / В.А. Лещев, А.Ф Семенов, И.А. Кеменов, И.А. Конюхов // Программные продукты и системы. – 2006. - №1, С. 23-29.

5. Борщев, А.В. Практическое агентное моделирование и его место в арсенале аналитика [Текст] / А.В. Борщев // Exponenta Pro. – 2004. - № 3, С. 38-47.

6. Java Agent DEvelopment Framework [Электронный ресурс]: / Telecom Italia SpA – Режим доступа: http://jade.tilab.com, свободный.

7. The Foundation for Intelligent Phisical Agents [Электронный ресурс]: / IEEE FIPA – Режим доступа: http://www.fipa.org, свободный.

8. Мохов, В.А. Тестирование сервисов на устойчивость к DDOS атакам на основе мультиагентного моделирования [Текст] / В.А. Мохов, С.А. Гончаров // Інформаційні системи та технології управління : матеріали III Міжнар. інтернет-конф., 25 жовт. 2012 р. / Донец. нац. ун-т економіки і торгівлі ім. М. Туган-Барановського - Донецьк: [ДонНУЕТ], - 2012. - С. 48-51.

9. Мохов, В.А. Интегрированный алгоритм когнитивной оценки и выбора оптимального варианта онтологической модели [Электронный ресурс] / В.А. Мохов, Н.Н. Сильнягин // «Инженерный вестник Дона», 2011, № 4. – Режим доступа: http://www.ivdon.ru/magazine/archive/n4y2011/600 (доступ свободный) – Загл. с экрана. – Яз. Рус.

10.Мохов, В.А. Мультиагентная модель для тестирования Интернет-сервисов на устойчивость к сетевым атакам типа DDOS [Текст] / В.А. Мохов, С.А. Гончаров // Академические фундаментальные исследования молодых ученых России и Германии в условиях глобального мира и новой культуры научных публикаций: материалы Междунар. молодеж. конф., г. Новочеркасск, 4-5 окт. 2012 г. / Юж.-Рос. гос. техн. ун-т (НПИ) - Новочеркасск : Лик, 2012. - С. 154-156.