УДК 004.056

К. А. Борисенко, Я. А. Бекенева, Н. Н. Шипилов, А. В. Шоров
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Система имитационного моделирования для разработки и тестирования методов защиты от DDoS-атак с возможностью подключения реальных узлов

Представлена гибридная система моделирования DDoS-атак и методов защиты компьютерных сетей. Разработанная система позволяет с минимальными трудозатратами создавать различные сетевые топологии, выполнять эксперименты по моделированию DDoS-атак, разработки новых и тестирования известных методов защиты. Система позволяет не только строить виртуальные сети, но и подключать реальные сетевые узлы для повышения точности выполняемых экспериментов. Описаны формальные модели компонентов, представлена архитектура системы, описан этап верификации виртуальной сети относительно реальной. Также описаны эксперименты выполнения DDoS-атак и работы методов защиты.

DDoS-атака, виртуальная сеть, реальная сеть, имитационное моделирование, сценарий поведения клиентов, методы защиты, OMNeT++, INET, ReaSE, PlanetLab, SYN Flooding, HTTP Flooding, Egress Filtering

DDoS-атака – распределенная атака типа «отказ в обслуживании», направленная на выход из строя различных интернет-сервисов. Часто крупным DDoS-атакам подвергаются веб-сайты правительств и органов власти разных стран, веб-сайты ведущих IT-корпораций Amazon, Yahoo, Microsoft и т. д. Согласно отчету ARBOR Networks [1] с 2005 по 2014 г. максимальная мощность DDoS-атак выросла с 8 Гбит/с до 400 Гбит/с, при этом к 2014 г. DDoS-атаки вышли на первое место среди максимальных угроз для сервисов. Более того, с 2012 г. мощность атак увеличилась в 6.5 раз. Как отмечает ARBOR Networks, 55 % опрошенных считают, что они практически не подготовлены к отражению DDoS-атак. Свыше 60 % опрошенных обнаружили более 10 атак в месяц в 2014 г. К 2015 г. основными жертвами подобных атак становятся не только серверы компаний и интернет-провайдеры, но и их клиенты.

Мировые лидеры по информационной безопасности [1] считают необходимость обнаружения и противостояния DDoS-атакам первостепенной задачей в своих исследованиях и разработках. Это свидетельствует о том, что разработка и внедрение методов защиты от DDoS-атак – актуальная задача.

В прошлых работах авторами была представлена система имитационного моделирования компьютерных сетей и различных типов DDoS-атак. Была поставлена задача разработки механизма включения реальных компьютеров в моделируемую сеть с целью повышения точности проведения экспериментов [2].

В данной статье описана реализация этой задачи, а также представлены эксперименты, произведенные с участием реальных узлов.

Рассмотрим общую архитектуру разработанной системы моделирования. Она может быть представлена в виде совокупности компонентов, иерархия которых показана на рис. 1.

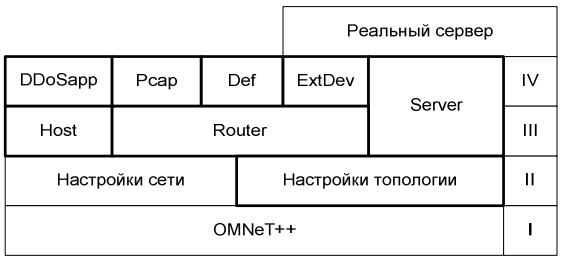

Рис. 1

На первом уровне иерархии (I) выделена дискретно-событийная система моделирования OMNeT++ [3]. Она была выбрана в качестве ядра системы, так как находится в открытом доступе, поддерживается большой командой разработчиков. Кроме того, существует большое количество библиотек, которые позволяют создавать виртуальные сети с характеристиками, максимально схожими с характеристиками реальной сети.

Второй уровень иерархии представлен моделью компьютерной сети (II). В качестве настроек сети, коммутации пакетов используется библиотека INET [4]. В качестве настроек топологии была доработана библиотека ReaSE [5]. Ее поддержку авторы прекратили в 2011 г., поэтому она была обновлена для работы с версией OMNeT++ 4.5.

На третьем уровне выделены разработанные модели хостов и маршрутизаторов (III), входящих в сеть. Модель сервера относится как к этому уровню, так и к следующему.

На четвертом уровне представлены следующие модели (IV), разработанные авторами:

• DDoSApp – приложение, которое будет выполнять атаку по заданному сценарию.

• Pcap – модули учета потока трафика.

• Def – алгоритм защиты, запущенный на маршрутизаторе. Алгоритм пишется программно, в маршрутизаторе может присутствовать как любое количество алгоритмов, так и ни одного.

• ExtDev – количество интерфейсов, подключенных к внешней сети. Необходимо для маршрутизаторов, в локальной сети которых находится внешний узел.

Созданные модели и архитектура были использованы для разработки системы гибридного моделирования. Для виртуализации операционных систем и вычислительных ресурсов использовались программные среды VirtualBox и Vmware. Разработанная система позволяет удобно и быстро строить различные топологии виртуальных сетей и настроить подключение к реальному серверу. Система позволяет создавать сценарии поведения клиентов, при этом модель сценариев атак DDoSApp можно использовать как сценарий поведения легитимного клиента. Все модели могут быть легко изменены. В экспериментах можно применить любой алгоритм защиты – как уже существующий, так и разработанный пользователем. Использование системы позволит максимально сократить время на подготовку экспериментов, тем самым предоставляя большее количество времени на подготовку и тестирование новых методов защиты.

Далее приведены примеры настроек созданных компонентов.

В качестве примера реализации модели Router представлена структура перенаправляющего маршрутизатора (рис. 2).

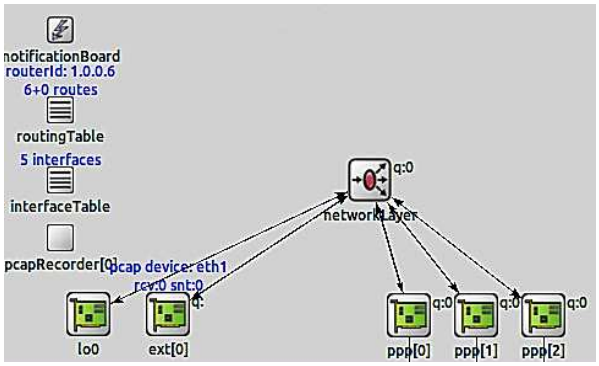

Рис. 2

Настройки моделей Pcap, ExtDev на примере того же маршрутизатора:

**.edge2.numPcapRecorders = 1

#pcaprecorder for router

**.edge2.pcapRecorder[0].pcapFile = “inetVerific1.pcap”

*.edge2.numExtInterfaces = 1

*.edge2.routingFile = “reaseserv.mrt”

*.edge2.ext[*].filterString = “ip and dst host 1.0”

*.edge2.ext[*].device = “eth1”

*.edge2.ext[*].mtu = 1500B

Программное обеспечение tcpdump использовалось в процессе отладки экспериментов для мониторинга сетевой активности в реальном времени. Для записи всех сетевых пакетов на реальном сервере и реальных узлах во время проведения экспериментов использовалось программное обеспечение tshark.

Созданная структура клиента виртуальной сети выглядит следующим образом (рис. 3).

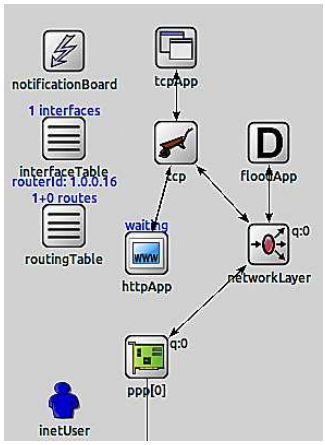

Рис. 3

Из-за специфики атаки TCP Flooding необходимо обойти стандартный сценарий работы TCP-протокола, а именно трехэтапное рукопожатие. Но для выполнения HTTP Flooding TCP-протокол дол-

жен функционировать корректно, поэтому модули httpApp и floodApp разделены, причем все их настройки заданы одинаковыми, кроме SpecialP.

Для проведения экспериментов использовался компьютер со следующими характеристиками:

Процессор: Intel Core i7-3770 3,4 Ghz * 8 ядер.
RAM: 15,6 GB DDR3 1600 Mhz.
ОС: Ubuntu 14.04 64-bit.

Разработанная система может быть развернута и на семействе операционных систем Windows. На данный момент система моделирования работает в однопоточном режиме. Используется более 200 виртуальных ПК, атакующих с задержкой в 10 мс. При этом загрузка процессора во время экспериментов составляет 50 % мощности одного ядра.

В процессе экспериментов система может записывать логи трафика, которые в дальнейшем можно использовать для анализа работы механизмов атаки и защиты.

Верификация системы. Для проверки корректности работы созданных моделей и алгоритмов была проведена верификация разработанной системы моделирования.

Были проведены серии экспериментов с помощью динамических методов верификации, таких, как мониторинг и системное тестирование. С помощью мониторинга оценивалось поведение системы в ходе ее обычной работы. Системное тестирование использовалось для проверки взаимодействия модулей в рамках системы в целом, при этом оценивалась работа моделируемой сети относительно реальной.

Реальная сеть создавалась с помощью проекта PlanetLab [6]. В разработанной системе моделирования была создана виртуальная сеть, соответствующая сети, построенной в PlanetLab. После проведения серии тестов были сделаны выводы о корректности работы системы моделирования в различных условиях.

В ходе верификации платформа PlanetLab использовалась в экспериментах по проведению атак SYN-flooding и HTTP-flooding с участием реальных узлов, расположенных в разных странах Европы. Инструменты Mtr, traceroute, tracepath использовались для определения топологии сети, построенной с помощью платформы PlanetLab, и определения времени ответа на запрос между этими узлами. Полученные данные служили для построения адекватной модели соответствующей сети.

Для проведения атак SYN-floofing на узлах PlanetLab применялось программное обеспечение Mausezahn, которое запускалось одновременно на всех атакующих узлах сети для многократной генерации SYN-запросов атакуемому серверу.

Для проведения HTTP-flooding на узлах PlanetLab использовалось программное обеспечение cUrl. Оно одновременно запускалось на всех атакующих узлах для многократной генерации запросов index.html, отправляемых на атакуемый сервер.

Сравнение логов трафика. Для примера рассмотрим логи HTTP-запроса, выполненного в виртуальной и в реальной сетях (рис. 4 и 5).

1.0.0.102	2.0.0.1	TCP	1002→80 [SYN] Seq=0 Win=7504
2.0.0.1	1.0.0.102	TCP	80→1002 [SYN, ACK] Seq=0 Ack=
1.0.0.102	2.0.0.1	TCP	1002→80 [ACK] Seq=1 Ack=1 Wir
1.0.0.102	2.0.0.1	HTTP	GET / HTTP/1.1
2.0.0.1	1.0.0.102	TCP	80→1002 [ACK] Seq=1 Ack=31 Wi
2.0.0.1	1.0.0.102	HTTP	HTTP/1.1 200 OK (text/html)
1.0.0.102	2.0.0.1	TCP	1002→80 [ACK] Seq=31 Ack=581
1.0.0.102	2.0.0.1	TCP	1002→80 [FIN, ACK] Seq=31 Ack
2.0.0.1	1.0.0.102	TCP	80→1002 [FIN, ACK] Seq=581 Ac
1.0.0.102	2.0.0.1	TCP	1002→80 [ACK] Seq=32 Ack=582

Рис. 4

2.0.0.2	2.0.0.1	TCP	43359→80 [SYN] Seq=0 Win=292
2.0.0.1	2.0.0.2	TCP	80→43359 [SYN, ACK] Seq=0 Ac
2.0.0.2	2.0.0.1	TCP	43359→80 [ACK] Seq=1 Ack=1 W
2.0.0.2	2.0.0.1	HTTP	GET / HTTP/1.1
2.0.0.1	2.0.0.2	TCP	80→43359 [ACK] Seq=1 Ack=312
2.0.0.1	2.0.0.2	HTTP	HTTP/1.1 200 OK (text/html)
2.0.0.2	2.0.0.1	TCP	43359→80 [ACK] Seq=312 Ack=5
2.0.0.2	2.0.0.1	TCP	43359→80 [FIN, ACK] Seq=312
2.0.0.1	2.0.0.2	TCP	80→43359 [FIN, ACK] Seq=574
2.0.0.2	2.0.0.1	TCP	43359→80 [ACK] Seq=313 Ack=5

Рис. 5

Результаты приведены в хронологическом порядке. Рис. 4 и 5 показывают, что работа виртуального и реального клиентов совпадает. Повторные запросы в случае неполучения пакета сервером осуществляются в OMNeT++ с помощью библиотеки INET.

Сравнение характеристик сети. Характеристики виртуальной сети сравнивались с характеристиками реальной сети, а именно время, затрачиваемое на получение сервером SYN-пакета и на получение SYN-ACK-пакета от сервера клиентом. В реальной сети в одном из экспериментов было использовано 6 ПК и 1 сервер (рис. 6). Сервер и 1 ПК располагались в Хельсинки (Финляндия), 2 ПК находились в Оулу (Финляндия) и 4 ПК в Париже (Франция). В разработанной среде была создана топология виртуальной сети, повторяющая характеристики и свойства реальной сети.

На рис. 7 отображено время ответов сервера в реальной и виртуальной сетях. Точечной линией показан трафик реальной сети, сплошной – виртуальной. В реальной сети трафик проходит между сервером в Хельсинки и Париже. Времена ответа в

реальной и виртуальных сетях схожи. Стоит отметить, что, зная задержку ответа сервера компьютеру в любой точке мира, а также ее разброс и скорость передачи данных, можно воссоздать характеристики данного клиента в виртуальной сети. Это позволяет сделать модель Router.

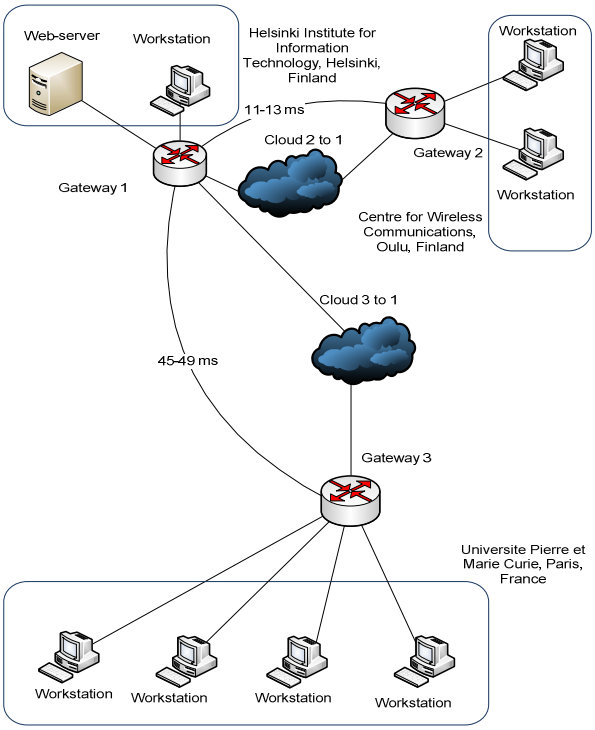

Рис. 6

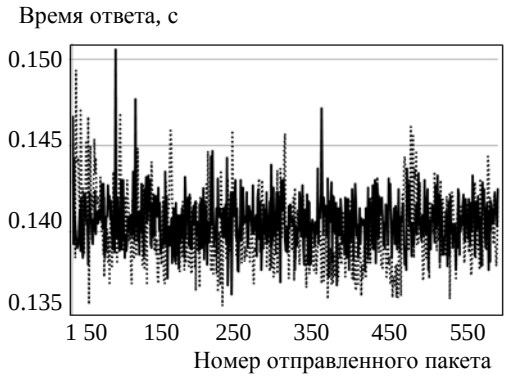

Рис. 7

Сравнение характеристик сети под воздействием атак. Характеристики виртуальной сети сравнивались с характеристиками реальной сети в случае выполнения атаки SYN-Flooding и атаки HTTP-Flooding. Для проведения эксперимента в реальной сети было использовано 36 ПК и 1 сервер, расположенные в различных странах Европы. В виртуальной сети была создана аналогичная топология, полностью повторяющая характеристики реальной.

Опыт 1. При выполнении атаки SYN-Flooding атакующие узлы начинали отправку вредоносных пакетов в произвольный момент времени, который фиксировался программными средствами. Узлы включались в атаку последовательно, с интервалом в 1 с, подмена адреса не использовалась.

На рис. 8 приведено сравнение времен ответа в реальной и виртуальной сетях для нескольких случаев. Результаты были зафиксированы на узле, расположенном в Румынии, для реальной сети и на его аналоге в виртуальной. Линией *1* показано время ответа сервера в реальной сети, 3 другие линии относятся к виртуальной сети. В первом случае (линия *2*) моделировалась ситуация, когда в атаке участвовало только 3 хоста, которые генерировали вредоносные пакеты с периодичностью 100 мс. Для этого случая времена ответа сервера в реальной и виртуальной сетях практически идентичны, разброс результатов не превышал 2 мс. Во втором случае (линия *3*) показана ситуация, когда в атаке принимали участие все узлы виртуальной сети, каждый из которых генерировал вредоносный пакет с периодичностью 100 мс. Из-за одновременного поступления большого числа пакетов на перенаправляющий маршрутизатор (см. рис. 2) задержка возросла на 5 мс. Это свидетельствует о том, что перенаправляющий маршрутизатор не способен обрабатывать большой объем входящего трафика с той же скоростью, что и реальный маршрутизатор. В третьем случае (линия *4*) показана ситуация, когда все узлы виртуальной сети участвуют в атаке с периодичностью 1 с, при этом для трех из них время начала атаки смещено на 0.5 с относительно остальных узлов. Из графика видно, что для этой ситуации времена ответа сервера практически идентичны для реальной и виртуальной сетей и разброс значений времени ответа не превышает 2 мс.

Рис. 8

Опыт 2. При выполнении атаки HTTP-Flooding атакующие узлы начинали отправку вредоносных пакетов в произвольный момент времени, который фиксировался программными средствами. Узлы включались в атаку распределенно по времени с разницей в 1 с. Время между завершением очередной TCP-сессии и началом новой составляло 100 мс для каждого узла.

На рис. 9 сравниваются времена ответа для реальной (линия *1*) и виртуальной сетей. Для виртуальной сети выделены 2 случая. В первом случае (линия *2*) в атаке участвуют 3 узла, график идентичнен графику, полученному для реальной сети, при этом разница не превышает 0.5 мс. Во втором случае (линия *3*) в атаке участвуют все узлы. Видно, что при увеличении нагрузки задержка ответа сервера возрастает, разброс результатов находится в пределах 2 мс.

Результаты экспериментов показали, что с увеличением количества атакующих узлов возрастает задержка ответа сервера. Для выявления зависимости времени задержки от количества атакующих серверов были проведены эксперименты с использованием сети, состоящей из 204 хостов. На рис. 10 показаны 3 различные ситуации, аналогичные описанным ранее.

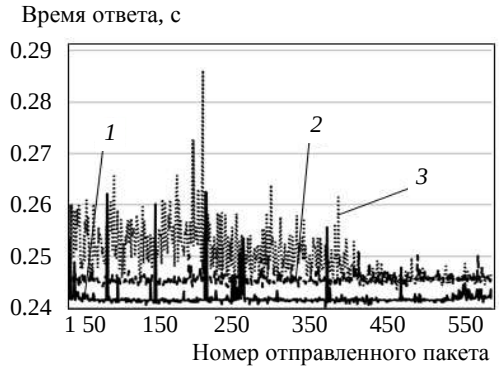

Рис. 9

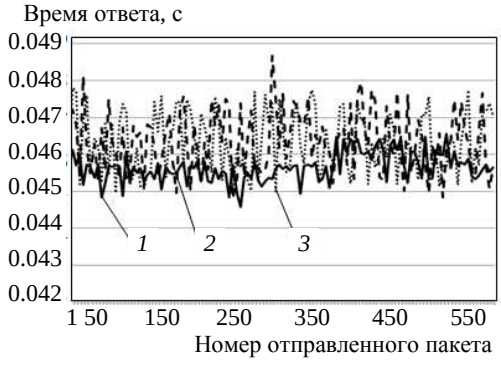

Рис. 10

Опыт 3. Далее число атакующих узлов было увеличено. На рис. 11 показаны 2 случая. В первом случае в атаке участвует 102 узла (линия *1*), во втором – 204 (линия *2*). Из графика видно, что по сравнению с результатами, представленных ранее, задержка ответа сервера значительно возросла. При этом ее среднее значение осталось неизменным для обеих ситуаций. При этом в реальной сети время ответа сервера для всех рассмотренных ситуаций оставалось неизменным.

Таким образом, при увеличении числа атакующих узлов время ответа сервера в имитационной модели превосходит время ответа в реальной сети. Однако при 100 и более атакующих узлах время задержки не изменяется. Эту особенность, а также вероятность потери отдельных пакетов следует учитывать при проектировании сетей с большим количеством узлов и атак в таких сетях.

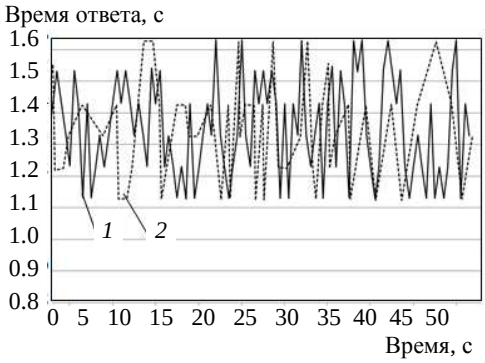

Рис. 11

Эксперименты. Перед тем как описывать проведенные эксперименты, необходимо указать характеристики и настройки реального сервера, который использовался при проведении экспериментов. Характеристики сервера:

Процессор: Intel Core i5-4440S 2.8 Ghz * 4 ядра.
RAM: 8 GB DDR3.
ОС: Ubuntu 14.04.1 LTS.
Веб-сервер: Apache/2.4.7.
Версия PHP: PHP 5.5.9-1

Для успешной атаки SYN Flooding на сервере были отключены SYN cookies. Для HTTP Flooding был написан скрипт на языке PHP, моделирующий уязвимость в настройке сервера, так что процессорное время, затраченное на обработку GET-запроса, достигало 120 мс.

```
<?php
$c=10000000/2;
For ($i = 0; $i<$c; $i++) {
```

```
$f = 5 + 5;
}
echo “DONE<br/>”;
```

Топология сети для выполнения экспериментов состояла из семи маршрутизаторов, 204 клиентов и одного реального сервера. Задержки между узлами сети равны 1 мкс. Библиотека ReaSE использовалась для моделирования легитимного трафика.

Перейдем к анализу экспериментов сценариев атак SYN Flooding и HTTP Flooding без использования методов защиты и с использованием метода Egress Filtering.

Эксперимент со сценарием атаки SYN-Flooding. Для этого эксперимента настройка сценария атаки выглядит следующим образом:

```
**.victimAddress =
“InetServTest.host_WebServer17”
**.activation_level = 1
**.attackType = “2” # http - 1
**.connectPort = 80
**.timeBetweenPackets = 250ms
**.attackStartTime = uniform(10s, 40s)
**.stopTime = uniform(45s, 60s)
**.maxPacket = “-1”
**.addressSpoofingOn = false
```

В атаке участвовало 100 % от общего количества клиентов в сети (204 компьютера). При этом 3 ПК атаковали на протяжении всего времени моделирования.

На рис. 12 отображены пакеты SYN (линия *1*), отправленные на сервер из виртуальной сети, а также показаны ответы сервера без повторных отправок SYN-ACK (линия *2*) для уже совершенных полуоткрытых соединений. До 10-й секунды к серверу обращаются всего 3 клиента. В начале атаки, на 10-й секунде, количество обращений к серверу увеличивается, так как все больше и больше виртуальных клиентов начинают принимать участие в атаке. До 15-й секунды сервер успевает обрабатывать все обращения, после чего TCP-стек переполняется и сервер не может обработать увеличивающийся поток обращений. Далее, начиная с 45-й секунды по 60-ю, виртуальные клиенты выходят из атаки. С 60-й по 90-ю секунду к серверу обращаются снова 3 клиента. На 90-й секунде моделирование было остановлено. В период с 15-й по 57-ю секунду сервер отвечал не на все приходящие SYN-пакеты.

Рис. 12

Эксперимент со сценарием HTTP Flooding. Для этого эксперимента настройка сценария атаки выглядит следующим образом:

```
**.victimAddress =
“InetServTest.host_WebServer17”
**.activation_level = 0.5
**.attackType = “1” # http - 1
**.connectPort = 80
**.timeBetweenPackets = 10ms
**.attackStartTime = uniform(10s, 50s)
**.stopTime = uniform(70s, 100s)
**.maxPacket = “–1”
```

В атаке участвовало около 100 клиентов, посылающих HTTP-запросы с интервалом между сессиями 10 мс. При этом один клиент выполнял сессию на протяжении всего времени моделирования. График на рис. 13 отображает длительность сессий относительно времени выполнения эксперимента. Начиная с 10-й секунды, при постепенном увеличении количества атакующих клиентов время сессии растет, при максимальной нагрузке атаки (50–70-я секунды) сервер отвечает более 3 с на запрос страницы, что превышает время ответа более чем в 25 раз. Даже при таком небольшом количестве атакующих клиентов сервер начинает отвечать на легитимные запросы более 3 с, не включая время задержки на путь пакетов от клиента до сервера и обратно (замеры проводились в локальной сети сервера).

Рис. 13

Эксперименты со сценарием SYN Flooding с использованием Egress Filtering. Egress Filtering – это один из наиболее популярных методов защиты от DDoS-атак. Он устанавливается на исходящие маршрутизаторы, например провайдером сети, для того чтобы за пределы сети не выходили пакеты с IP-адресами, которые не совпадают ни с одним существующим IP-адресом в этой сети.

Была проведена серия экспериментов с использованием фильтра на двух, трех и четырех маршрутизаторах в виртуальной сети. Был воссоздан сценарий атаки SYN Flooding с подменой IP-адреса отправителей. Три клиента также атаковали с начала и до конца эксперимента (рис. 14).

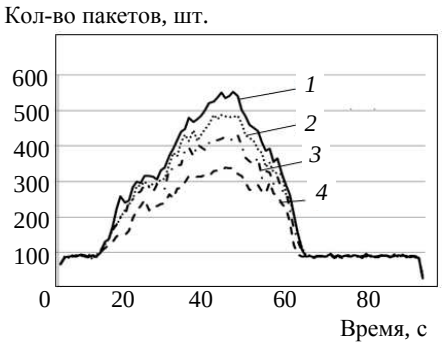

Рис. 14

Для подведения итогов экспериментов с фильтрацией необходимо знать, что в топологии всего было 6 маршрутизаторов, в локальных сетях которых были ПК. Соответственно эксперименты проводились как без использования фильтров (линия *1*), так и с использованием фильтров на двух (линия *2*), трех (линия *3*), четырех (линия *4*) маршрутизаторах. При этом клиенты, атакующие сервер постоянно, находились в локальной сети маршрутизатора, не использующего Egress Filtering. С увеличением количества фильтров наблюдалось уменьшение мощности атаки на сервер.

На основании приведенных экспериментов можно заключить, что система при правильной настройке показывает реалистичные результаты работы различных типов атак и защиты, не требует большого количества времени для настройки топологий и сценариев атаки.

Представленные эксперименты доказывают, что разработанную систему можно использовать в качестве лаборатории по созданию и тестированию механизмов защиты от DDoS-атак. При этом настройка топологий, клиентов, маршрутизаторов, сценариев атак занимает немного времени.

Разработанная система может быть использована для исследования DDoS-атак, а также методов защиты от них. Также администраторы сетей могут быстро и точно воспроизвести обслуживаемую ими сеть, выполнить нагрузочное тестирование, оценить устойчивость сервера к атакам, пропускную способность сети, а также качество работы механизмов защиты.

Верификация разработанной системы моделирования показала, что большое количество входящих пакетов, одновременно поступающих на перенаправляющий маршрутизатор, ухудшает качество моделирования взаимодействия реального сервера с виртуальными узлами. Эту особенность необходимо учитывать при построении сетей с большим количеством узлов и задании параметров атаки. В данный момент авторами ведется работа по повышению производительности системы моделирования.

В дальнейшем авторы планируют использовать систему для разработки и тестирования методов защиты на основе методов интеллектуального анализа данных.

СПИСОК ЛИТЕРАТУРЫ

1. Ежегодный отчет Worldwide Infrastructure Security Report. ARBOR Networks reports, 2014. URL: http://www.arbornetworks.com/resources/infrastructure-security-report.

2. Simulation of DDoS-attacks and protection mechanisms against them / Ya. Bekeneva, N. Shipilov, K. Borisenko, A. Shorov // Proc. of the 2015 IEEE North West Russia Section Young Researches in Electrical and Electronic Engineering Conf., St.-Petersburg, Febr. 2–4, 2015. P. 50–56.

3. Сайт разработчика программы OMNeT++ Discrete Event System Simulator. URL: http://www.omnetpp.org/intro.

4. Сайт разработчика программы INET Framework. URL: http://inet.omnetpp.org/.

5. Сайт разработчика программы ReaSE. URL: https://i72projekte.tm.uka.de/trac/ReaSE.

6. Сайт разработчика программы PlanetLabEurope. URL: https://www.planet-lab.eu/about.

K. A. Borisenko, Ya. A. Bekeneva, N. N. Shipilov, A. V. Shorov
Saint-Petersburg state electrotechnical university «LETI»

HYBRID MODELING SYSTEM FOR DDoS ATTCKS PROTECTION METHODS DEVELOPMENT AND TESTING

Protection against distributed attacks «distributed denial of service» (DDoS) is a very important task nowadays. In the paper we introduce a hybrid system for simulating DDoS attacks and computer network protection techniques. The developed system makes it possible to create various network topologies, perform experiments with DDoS attack simulation, develop new protection methods and test the existing ones. The suggested system not only allows us to design virtual networks, but also makes it possible to connect real network nodes for improving the accuracy of the experiments. The paper contains information on component formal models and system architecture. We also describe virtual network verification relative to a real network. Furthermore, we provide the experiments devoted to DDoS attacks and operation of the protection techniques.

DDoS attack, virtual network, real network, simulation modeling, client behavior scenario, protection techniques, OMNeT++, INET, ReaSE, PlanetLab, SYN Flooding, HTTP Flooding, Egress Filtering

УДК 519.87

В. И. Анисимов, М. А. Шабани
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Повышение эффективности систем автоматизированного проектирования на основе компактных методов обработки разреженных матриц

Рассматриваются методы повышения эффективности систем автоматизированного проектирования на основе технологии разреженных матриц. Приводится описание реализации компактных методов обработки разреженных матриц при построении программного обеспечения системы автоматизированного проектирования

Системы автоматизированного проектирования, моделирование систем, разреженные матрицы, распределенные системы, интернет-технологии

Одной из актуальных проблем в области дальнейшего развития систем автоматизированного проектирования является внедрение в них компактных методов обработки разреженных матриц, что позволяет существенно повысить эффективность программного обеспечения систем. Такая задача становится особенно актуальной при построении систем с распределенной архитектурой, где информационные ресурсы предоставляются потребителям посредством сетевых сервисов [1]–[3]. Использование интернет-технологий при разработке систем автоматизированного проектирования дает возможность реализовать слабосвязанное программное обеспечение, вследствие чего взаимодействие между приложениями не нарушается каждый раз, когда меняется реализация какого-либо сервиса, а также обеспечить взаимодействие на любой платформе между различными приложениями, написанными на любом языке программирования. Вместе с тем при построении систем с распределенной архитектурой существенно повышаются требования к быстродействию программного обеспечения, поскольку требуется минимизировать время взаимодействия пользователя с системой.

В связи с этим при создании систем автоматизированного проектирования необходимо использовать методы, основанные на формировании компактного описания моделируемой системы на основе тех или иных способов сжатия данных,