Математические структуры и моделирование 2017. № 3(43). С. 139–149

УДК УДК 004.7 : 004.056
DOI: 10.25513/2222-8772.2017.3.139-149

ПРОГРАММА, МОДЕЛИРУЮЩАЯ КОМПЬЮТЕРНУЮ СЕТЬ И СЕТЕВЫЕ АТАКИ

А.К. Гуц
профессор, д.ф.-м.н., кафедра кибернентики, e-mail: guts@omsu.ru
Е.П. Эннс
студент, e-mail: jecka-gtx@mail.ru

Омский государственный университет им. Ф.М. Достоевского

Аннотация. Представляется компьютерная программа, которая имитирует работу компьютерной сети, а также позволяет продемонстрировать результат атак злоумышленников на сеть, использующих некоторые имеющиеся в сети уязвимости.

Ключевые слова: компьютерная сеть, сетевые атаки, имитация атак, компьютерная программа.

Введение

Анализу моделей атак на компьютерные сети и проверке эффективности этих атак посвящается много работ. Очень часто для этого используются тестовые компьютерные сети [1], как правило, являющиеся реальными компьютерными сетями. Однако это же можно делать, создавая компьютерные программы, имитирующие реальные сети, атаки на них с демонстрацией последствий [2–4].

В данной статье представляется простая учебная компьютерная программа [5], которая моделирует работу компьютерной сети, а также позволяет продемонстрировать результат атак злоумышленников на сеть, использующих некоторые имеющиеся в сети уязвимости. Компьютер хакера и компьютеры сети представляются окнами на мониторе программы.

Основные задачи, которые необходимо было решить:

1) создание окна злоумышленника;

2) создание окон коммутатора и маршрутизатора, работающих по схожим принципам, что и реальные устройства;

3) создание окна сервера;

4) создание базового функционала окон хостов и описание способ конфигурирования моделируемой сети;

5) представление функционал окон хостов, в особенности – злоумышленника (хакера);

6) обеспечение возможности передачи сообщений между окнами программы;

7) демонстрация, что выбранные уязвимости в моделируемой сети присутствуют, а атаки с их использованием — действенны.

Уязвимость компьютерной системы — это недостаток в системе, её слабое место, используя которое можно намеренно нарушить целостность, т. е. отдельные элементы системы перестают функционировать и это приводит к неправильной её работе. Уязвимость может быть результатом ошибок программирования, недостатков, допущенных при проектировании системы, ненадёжных паролей, вирусов и т. д.

1. Моделирование

Для того чтобы наглядно показать действие атак, использующих уязвимости компьютерных сетей, требуется создать соответствующую компьютерную программу, с помощью которой можно имитировать нужную атаку.

Опишем возможный принцип работы такой программы.

1.1. Окна

Компьютер злоумышленника (хакера) и компьютеры сети в программе представляются окнами на мониторе запущенной программы.

В каждом окне (кроме окна коммутатора) имеется возможность назначить адрес, а в окна компьютеров (и злоумышленника также) — ещё и назначить шлюз.

Внешние адреса машин в программе — 100 и больше.

Это сделано для моделирования локальной (доменной) сети и внешней сети (интернет). В окно злоумышленника добавлена возможность отправить пакет, собранный из отдельных «кусочков», каждый из которых задаётся в отдельности.

Программа использует свою структуру сетевого пакета в отличие от реальных сетей. Это допущение позволяет увеличить наглядность и упростить моделирование.

В реальности для атаки часто требуется большое количество хостов, мы ограничимся двумя-тремя.

1.2. Отсылка сообщения хакером

Злоумышленник направляет сообщение в сеть, сеть его принимает: имеем передачу сообщений между двумя окнами, представляющими два компьютера.

Чтобы реализовать возможность передачи сообщений между окнами программы, было решено использовать несколько встроенных классов, обеспечивающих клиентские подключения для сетевых служб, работающих с потоками данных.

В функции отправки создаётся экземпляр класса «TcpClient», сообщение переводится из строкового типа в байтовый массив, который уже с помощью

открывшегося потока передачи данных посылается на указанный порт. Затем поток закрывается, а экземпляр класса удаляется:

```
                    //создание нового TCP клиента
    TcpClient client = new TcpClient("localhost Convert.ToInt32(port));
    byte[] data = new byte[256];
                    //перевод строки в байтовый массив
    byte[] msg = Encoding.UTF8.GetBytes(str);
    int count = msg.Length <= 256 ? msg.Length : 256;
    Array.Copy(msg, data, count);
                    //создание потока передачи данных
    NetworkStream stream = client.GetStream();
                    //запись строки в поток
    stream.Write(data, 0, 256);
                    //закрытие потока и удаление TCP клиента
    stream.Close();
    client.Close();
    textBox3.Text += "Отправлено "+ oppPort + »» "+ str;
    textBox3.Text += Environment.NewLine;
```

1.3. Приём сообщения системой

Приём сообщения осуществляется посредством создания потоков. Чтобы прослушивать заданный порт беспрерывно, а не по нажатию кнопки, создаётся отдельный поток, который этим занимается, периодически «засыпающий», чтобы постоянно не грузить систему. На каждый прослушиваемый порт создаётся свой поток, благодаря чему можно принимать сообщение одновременно со всех портов. Если в поток приёма данных попадает байтовый массив, переданный из одного окна другому, то этот массив переводится в строковую переменную.

1.4. Передача сообщений между окнами

В окнах злоумышленника и компьютеров сети предусмотрена возможность отправить сообщение на выбранный адрес. При этом если отправитель не знает входящий порт адресата (имитация MAC-адреса), то будет послан широковещательный пакет ARP-Request. Также был добавлен обработчик, проверяющий, что за сообщение нам пришло, и либо отбрасывающий его, либо инициализирующий какие-то определённые действия.

Например, если пришёл ARP-Request, то в ответ отправляется ARP-Reply:

```
                    //Если ARP-Request
if (instr.Substring(13, 1) == "1")
    {
    textBox.Text += ««ARP REQUEST от "+ instr.Substring(10, 3);
    textBox.Text += Environment.NewLine;
                    //Обычное + Mac-адрес назначения + наш Mac-адрес +
адрес назначения + наш адрес + ARP-Reply
```

```
str += "0"+ instr.Substring(4, 3) + localPort + instr.Substring(10, 3) +
adres + "2";
Thread.Sleep(1000);
                    //Отправка
Send(str);
return;
}
```

1.5. Окно сервера

Создано окно сервера. Оно принимает не ARP сообщения только после процедуры «тройного рукопожатия».

После получения пакета SYN сервер записывает исходящий MAC-адрес из пакета в буфер и отправляет ответ SYN-ACK. Если затем придёт пакет ACK от той же машины, то сервер «запомнит» подключение и уберёт соответствующую запись из буфера.

Чистку буфера через некоторое время, как на настоящем оборудовании, решено не добавлять, так как программа работает значительно медленнее реальных сетей, и для демонстрации это не понадобится.

1.6. Конфигурирование сети

В программе конфигурирование сети производится посредством кнопок «Добавить маршрутизатор», «Добавить коммутатор», «Добавить сервер», «Добавить компьютер».

1.7. Окно коммутатора

Создано окно коммутатора. Принимая на один их портов пакет, коммутатор сначала проверяет MAC-адрес отправителя и ищет его в своей таблице MAC-адресов.

Если не находит, то делает в ней запись, что этот MAC доступен по порту, с которого пришло сообщение. Затем проверяется MAC-адрес назначения и также ищется в таблице. Если находится, то пакет посылается на нужный порт, если не находится — происходит широковещательная рассылка.

Для того чтобы программа нормально функционировала, на все «устройства» добавлены переменные или их массивы (в зависимости от типа), хранящие информацию о том, какие «соседские» порты к каким локальным портам подключены (имитация соединения проводом). Иначе окна знали бы только на какой порт пришёл пакет, но не с какого на соседнем «устройстве».

1.8. Окно маршрутизатора

Создано окно маршрутизатора. Его отличает от коммутатора то, что он работает с обычными адресами, а не с MAC-адресами. А также то, что маршрутизаторы обмениваются между собой таблицами маршрутизации.

В программе это происходит при изменении топологии сети, инициализированном с одного из них.

1.9. Тестирование программы

Программа написана на языке C#.

Программа должна имитировать разные типы атак на компьютерную сеть. Атаки тестируются. Если что-то будет работать ненадлежащим образом, то следует найти причины ошибки и исправить её.

2. Запуск программы и работа с ней

2.1. Запуск программы

При запуске программы появляется стартовое окно — окно злоумышленника (рис. 1).

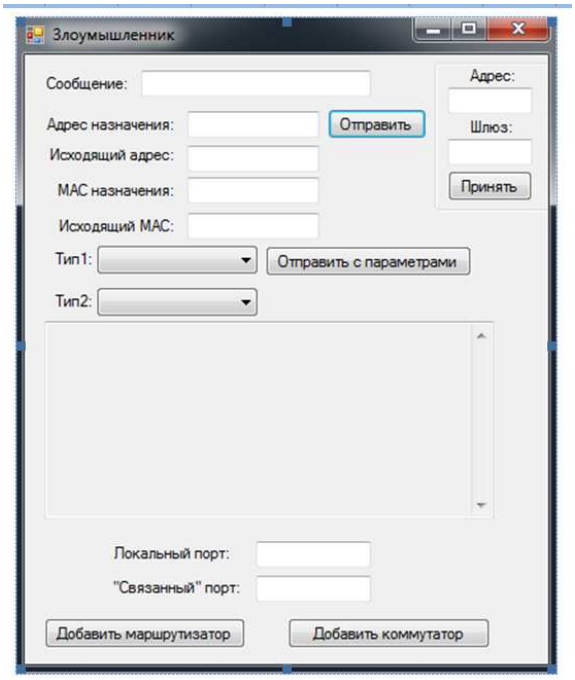

Рис. 1. Окно злоумыленника

Мы видим, что злоумышленник имеет возможность отправить произвольное сообщение на указанный адрес внутри моделируемой сети, подключить новый маршрутизатор или коммутатор.

Имеется поле, куда будут выводиться полученные и служебные сообщения.

Основное отличие этого окна от других «хостов» — это наличие кнопки «**Отправить с параметрами**», позволяющей, используя введённые пользователем данные, передать сообщение с произвольными адресами назначения и отправки, а также выбранного типа.

2.2. Конфигурирование сети

Чтобы создать компьютерную сеть, которую злоумышленнику предстоит атаковать, нужно из окна злоумышленника «найти», породить маршрутизатор или коммутатор, принадлежащий этой сети.

Это новое устройство будет сразу подключено к компьютеру злоумышленника «проводом». Для это следует в два поля ввести **трёхзначные адреса**, первый в поле «Локальный порт» — это порт компьютера злоумышленника, с которого идёт подключение, второй — в поле «Связанный порт» — подключённого «проводом» порта второго создаваемого устройства, затем нажать «Добавить маршрутизатор» или «Добавить коммутатор».

2.3. Порождение хостов сети

В открывшемся окне порождённого маршрутизатора или коммутатора можно добавлять в сеть новый компьютер. Добавленная машина сразу соединится с родительским устройством (с тем, с которого она была открыта).

В созданном хосте нет возможности добавлять новые узлы в сеть, так как в данной программе он является одной из конечных точек сети.

2.4. Добавление сервера

Добавление сервера производится из окна коммутатора. Его отличие от компьютера в том, что он не принимает входящие сообщения, пока не будет установлено соединение процессом «тройного рукопожатия».

3. Имитация атак

Программа активно использует многопоточность для симуляции сети, в особенности это важно для коммутаторов и маршрутизаторов, т. к. им приходится прослушивать сразу несколько портов. Сообщения между окнами передаются через TCP-клиенты потоковой передачей.

Программа позволяет имитировать атаки типа Smurfing (ICMP-Flood), SYN-Flood, подмена MAC-адреса, ARP-spoofing.

3.1. Smurfing (ICMP-Flood)

Для демонстрации взрывного увеличения сетевого трафика в окне злоумышленника имеется возможность отослать пакет ICMP ECHO REQUEST (поле «Тип 1»). В окнах компьютеров, в свою очередь, добавлена функция обработки этого условного пакета и отправки в ответ ICMP ECHO REPLY.

Для отправки такого сообщения с компьютера 200, принадлежащего злоумышленнику, с помощью кнопки «Отправить с параметрами» надо выбрать данные, как на рисунке (см. рис. 2, слева). Тогда после первой отправки инициализируется ARP знакомство хостов, а после второй уже пойдут пакеты ICMP ECHO REPLY на машину жертвы (компьютер 300, см. рис. 2, справа). Как

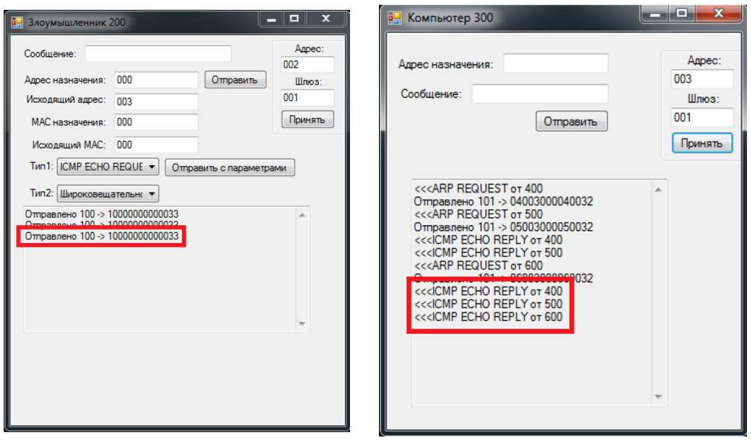

Рис. 2. Атака Smurfing. Окно злоумышленника слева, окно жертвы – справа

видно из рисунка, количество сообщений на канале передачи данных жертвы равно количеству хостов в сети (не считая двух «главных героев») при всего лишь одном сообщении у злоумышленника.

Цикл широковещательной отправки:

```
                    //отправка по всем каналам, кроме того, откуда пришло
for (int i = 0; i < llen; i++)
{
            if (localPorts[i] != port)
            {
            Thread.Sleep(200);
            Send(outPorts[i], str);
            }
}
```

3.2. SYN-Flood

SYN-Flood использует механизм установления TCP-соединения. Есть три состояния: посылка SYN-пакета, получение пакета SYN-ACK и посылка ACK-пакета.

Идея атаки состоит в создании большого количества не до конца установленных (полуоткрытых) TCP-соединений. При этом злоумышленник создаёт и от имени любых, даже никому не принадлежащих, IP-адресов направляет на атакуемый сервер большое количество запросов на установление соединения (пакеты с флагом SYN). Атакуемая машина отвечает пакетами-подтверждениями SYN-ACK на ложные адреса отправителей в интернете, которые в результате или не найдут адресата, или «озадачат» операционную систему на действующем IP-адресе (поскольку никаких запросов эта система на данный сервер не отправляла) и будут проигнорированы.

Для каждого полученного SYN-пакета сервер-жертва выделяет место в буфере, в результате исчерпания ресурсов которого новые соединения (даже настоящие) не могут быть открыты, поскольку протокол сеансового уровня не сможет вычислить фальшивые запросы с тем, чтобы установить больший приоритет настоящим.

Для начала демонстрации существенных задержек при установке связи нужно добавить в сеть сервер.

Его отличие от компьютера в том, что он не принимает входящие сообщения, пока не будет установлено соединение процессом «тройного рукопожатия».

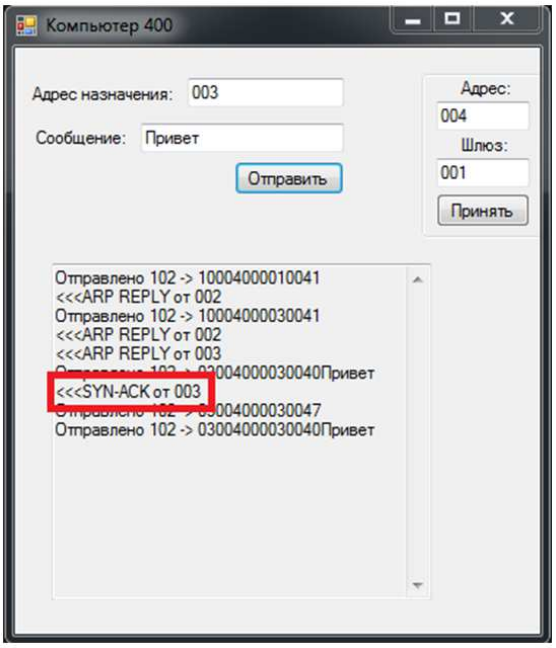

Рис. 3. Ответ SYN-ACK от сервера

В обычной ситуации сервер отвечает пакетом SYN-ACK на все новые подключения (см. рис. 3). Но если злоумышленник начнёт непрерывно посылать от чужих адресов пакеты SYN на сервер, последний будет добавлять эти запросы в очередь. А поскольку злоумышленник отвечать на SYN-ACK сервера не будет, очередь переполнится, и не останется места новым подключениям. Даже при условии чистки очереди через некоторое время (тайм-аут подключений) задержки будут серьёзными.

3.3. Подмена MAC-адреса

Хакер может легко ассоциировать рабочий IP-адрес с фальшивым MAC-адресом. Например, он посылает ARP-ответ, который связывает IP вашего роутера с несуществующим MAC-адресом. Ваши компьютеры уверены в том, что они знают шлюз по умолчанию, но в реальности они шлют пакеты неизвестно

куда и неизвестно кому. Одним легким движением злоумышленник отрезал вас от интернета.

Для демонстрации изменения MAC-адреса в ARP-таблице для начала отправим сообщение «Первый привет» с машины жертвы с внешним адресом назначения (в программе это 100 и больше) на заранее созданный компьютер, находящийся «за» маршрутизатором и имитирующим интернет. Послание дойдёт до адресата.

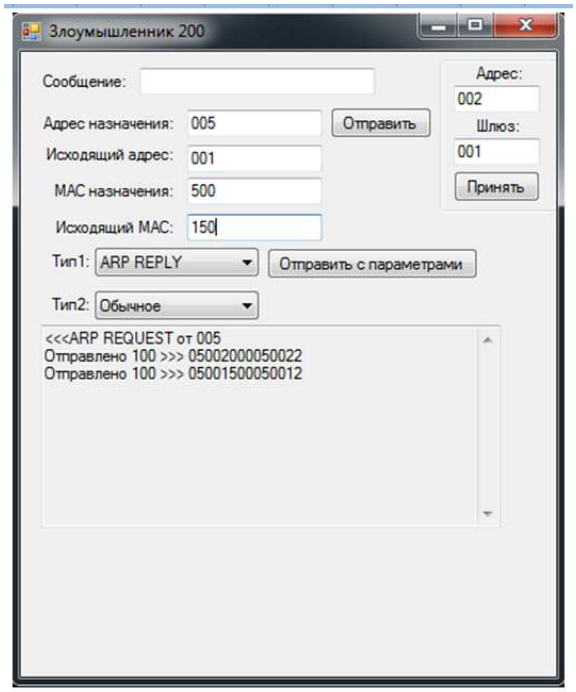

Рис. 4. Отправка поддельного ARP-Reply от хакера

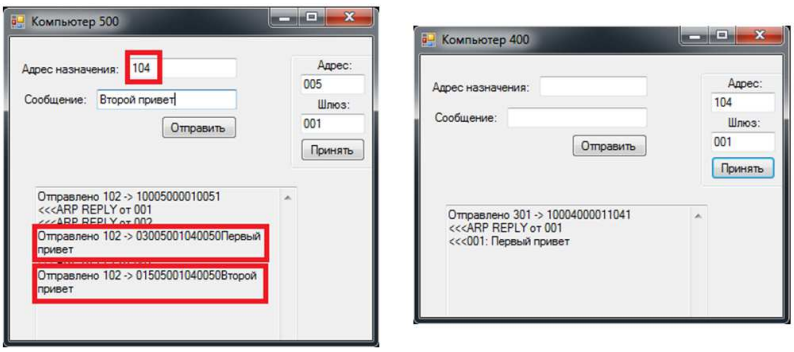

Рис. 5. Отправка двух сообщений с машины (слева). Получение первого сообщения (справа)

Затем отправим от злоумышленника ARP-Reply с нужными параметрами (см. рис. 4), чтобы машина жертвы, сопоставив IP из пакета с IP роутера,

заменила его MAC-адрес на новый. После этого попытка снова отправить сообщение «Второй привет» (см.рис. 5, слева) не увенчается успехом (см. рис. 5, справа). Теперь жертва отрезана от интернета.

3.4. ARP-spoofing

Программа демонстрирует возможности перехвата сетевого трафика. Злоумышленник посылает поддельный пакет ARP reply роутеру, который, сопоставив IP из пакета с IP жертвы, заменяет её MAC-адрес на MAC-адрес злоумышленника. Затем посылается поддельный пакет ARP reply жертве, которая, сопоставив IP из пакета с IP роутера, заменяет его MAC-адрес на MAC-адрес злоумышленника.

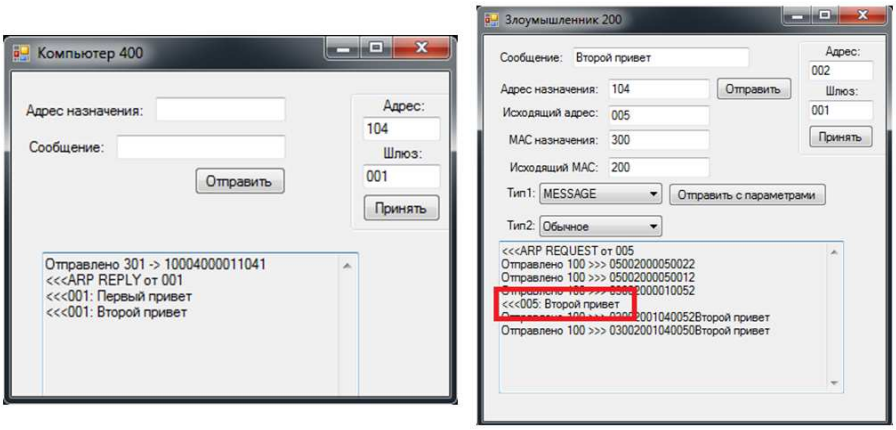

Рис. 6. ARP-spoofing. Окно злоумышленника слева, окно жертвы – справа.

Теперь злоумышленник может просматривать исходящий сетевой трафик жертвы. Данная атака весьма похожа на подмену MAC-адреса, отличие в том, что второе сообщение всё-таки доходит до компьютера-адресата (см. рис. 6, слева), но не напрямую, а предварительно пройдя через машину злоумышленника (см. рис. 6, справа).

Литература

1. Степашкин М.В., Котенко И.В., Богданов В.С. Моделирование атак для активного анализа уязвимостей компьютерных сетей // Вторая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» ИММОД–2005. С. 269–273.
2. Шоров А.В. Имитационное моделирование механизмов защиты компьютерных сетей от инфраструктурных атак на основе подхода «нервная система сети». Автореферат диссертации на соискание учёной степени кандидата технических наук. Санкт-Петербург, 2012.

3. Тумоян Е.П., Кавчук Д.А. Моделирование сетевых атак в задачах автоматического анализа защищённости информационных систем // Системи обробки інформації. 2012. Т. 2, вип.4. С. 74–78.
4. Мордвин Д.В., Абрамов Е.С. Разработка инструментальных средств для моделирования ЛВС // Известия ЮФУ. Технические науки. 2007. № 4. С. 113–116.
5. Эннс Е.П. Уязвимости сетевых протоколов и их моделирование: Выпускная квалификационная работа. ОмГУ, кафедра кибернетики, 2017. 29 с.

PROGRAM FOR SIMULATION OF COMPUTER NETWORK AND NETWORK ATTACKS

A.K. Guts
Dr.Sc. (Phys.-Math.), Professor, e-mail: guts@omsu.ru
E.P. Enns
Student, e-mail: jecka-gtx@mail.ru

Dostoevsky Omsk State University

Abstract. We present a computer program that simulates the work of computer network and allows us to demonstrate the result of malicious attacks on network using several existing network vulnerability.

Keywords: computer network, network attacks, attack simulation, computer program.

Дата поступления в редакцию: 02.07.2017