МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Ф.М. ДОСТОЕВСКОГО

ОМСКИЕ НАУЧНЫЕ ЧТЕНИЯ – 2018

Материалы Второй Всероссийской научной конференции

(Омск, 10–15 декабря 2018 г.)

ISBN 978-5-7779-2339-4

Омск

ИЗДАТЕЛЬСТВО ОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2018

МОДЕЛИРОВАНИЕ СЕТЕВЫХ АТАК

О.В. Матюшина, А.К. Гуц

Омский государственный университет им. Ф.М. Достоевского, г. Омск, Россия
E-mail: vladimirova.o94@gmail.com; guts@omsu.ru

SIMULATION OF COMPUTER NETWORK ATTACKS

O.V. Matyushina, A.K. Guts

Dostoevsky Omsk State University, Omsk, Russia

В докладе представлена компьютерная программа, позволяющая имитировать сетевые атаки на компьютерные сети.

In the report we present a computer program that allows to simulate network attacks on computer networks.

Ключевые слова: сетевые атаки, имитация атак, моделирование компьютерной сети.

Keywords: network attacks, simulation of attacks, modeling of computer networks.

В докладе представлена компьютерная программа, имитирующая сетевые атаки. В ходе работы программы открываются окна, представляющие компьютеры, входящие в состав сети.

Для разработки использовался язык программирования C#. Чтобы реализовать возможность передачи сообщений между окнами программы было решено использовать несколько встроенных классов: обеспечивающих клиентские подключения для сетевых служб, работающих с потоками и потоками данных.

В функции отправки создается 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;
```

Прием сообщения осуществляется сложнее. Здесь уже вступают в роль потоки. Чтобы прослушивать заданный порт беспрерывно, а не по нажатию кнопки, создается отдельный

поток, который выполняет востребованную функцию, периодически «засыпая», чтобы постоянно не грузить систему. На каждый прослушиваемый порт создается свой поток, благодаря чему можно принимать сообщение одновременно с них всех. Если в поток приема данных попадает байтовый массив, переданный с другого окна, то этот массив переводится в строковую переменную.

2. Для окон злоумышленника и компьютеров была предоставлена 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 определенные действия. Например, если пришел ARP-Request – отправить в ответ 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;  }
```

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 придет пакет ACK от той же машины, то сервер «запомнит» подключение и уберет соответствующую запись из буфера. Чистку буфера через некоторое время, как на настоящем оборудовании, решено не предоставлять, так как программа работает значительно медленнее реальных сетей и для демонстрации это не понадобится.

Был принят в работу способ конфигурирования сети посредством кнопок «Добавить …».

3. 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 так же ищется в таблице. Если находит, пакет посылается на нужный порт, если нет – происходит широковещательная рассылка.

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

4. Создано окно маршрутизатора. Его отличает от коммутатора то, что он работает с обычными адресами, а не с MAC. А также то, что маршрутизаторы обмениваются между собой таблицами маршрутизации. В программе это происходит при изменении топологии сети, инициализированном с одного из них.

5. Во всех окнах (кроме коммутатора) имеется возможность назначить адрес, а в компьютеры (злоумышленника тоже) – ещё и назначить шлюз. Это сделано для имитации локальной (доменной) сети и внешней (интернет). В окне злоумышленника добавлена возмож-

ность отправить пакет, собранный из отдельных «кусочков», каждый из которых задается в отдельности.

6. Атаки были протестированы.

Разработанное программное приложение полезно в первую очередь для обучения студентов и администраторов, которые по долгу службы должны обеспечивать безопасность информационных ресурсов. Данная работа продолжает разработки по моделированию атак на компьютерные сети посредством создания специализированного программного обеспечения, начатые в [1]. Приложение не предусматривает демонстрацию защиты от сетевых атак.

---

1. *Гуц А.К., Эннс Е.П.* Программа, моделирующая компьютерную сеть и сетевые атаки // Математические структуры и моделирование. 2017. № 3 (43). С. 139–149.