Математические структуры и моделирование 2019. № 1(49). С. 108–118

УДК 004.7:004.056
DOI: 10.25513/2222-8772.2019.1.108-118

ПРОГРАММА, ИМИТИРУЮЩАЯ СЕТЕВЫЕ АТАКИ НА КОМПЬЮТЕРЫ

А.К. Гуц
д.ф.-м.н., профессор, e-mail: aguts@mail.ru
О.В. Матюшина
студент, e-mail: vladimirova.o94@gmail.com

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация. Цель статьи — представить программу, которая моделирует работу компьютерной сети, а также демонстрирует результаты использования недобросовестным пользователем некоторых сетевых уязвимостей.

Ключевые слова: компьютерная сеть, сетевые атаки, сетевые уязвимости, имитация сетевых атак.

Введение

При обучении студентов по направлению «Информационная безопасность», осваивающих средства защиты от сетевых атак, полезно иметь программное приложение, имитирующее сетевые атаки типа Smurfing, SYN-Flood, подмена MAC-адрес, ARP-spoofing и другие. Такие программы уже создавались в ОмГУ [1, 2], но каждая из них, к сожалению, имеет свои недостатки, хотя и решает поставленные задачи.

В статье представлена новая компьютерная программа, имитирующая сетевые атаки.

В ходе работы программы открываются окна, представляющие компьютер злоумышленника и компьютеры, входящие в состав атакуемой сети.

1. Описание окон программы и их возможностей

Для разработки использовался язык программирования $C\sharp$.

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 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;
```

Приём сообщения осуществляется сложнее. Здесь уже вступают в роль потоки. Чтобы прослушивать заданный порт беспрерывно, а не по нажатию кнопки, создаётся отдельный поток, который этим занимается, периодически «засыпая», чтобы постоянно не грузить систему. На каждый прослушиваемый порт создаётся свой поток, благодаря чему можно принимать сообщение одновременно с них всех. Если в поток приёма данных попадает байтовый массив, переданный с другого окна, то этот массив переводится в строковую переменную.

Окна злоумышленника и компьютеров. На окна злоумышленника и компьютеров была добавлена 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 — 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;
}
```

Окно сервера. Это окно 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.

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

Окно коммутатора. Принимая на один из 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 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 о том, какие «соседские» порты подключены к каким локальным портам (имитация соединения проводом). Иначе окна знали бы только на какой порт пришёл пакет, но не с какого на соседнем «устройстве».

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.

Во все окна (кроме коммутатора) добавлена возможность назначить адрес, а в компьютеры и злоумышленника — ещё возможность назначить шлюз. Это сделано для имитации локальной (доменной) сети и внешней (интернет). В окно злоумышленника добавлена также возможность отправить пакет, собранный из отдельных «кусочков», каждый из которых задаётся в отдельности.

Атаки были протестированы.

2. Основная структура

Там, где в реальности для атаки необходимо большое количество хостов, мы ограничимся двумя-тремя. Программа использует свою структуру сетевого пакета в отличие от реальных сетей. Это допущение позволяет увеличить наглядность и упростить моделирование. При запуске программы открывается окно, выступающее в роли компьютера взломщика (см. рис. 1).

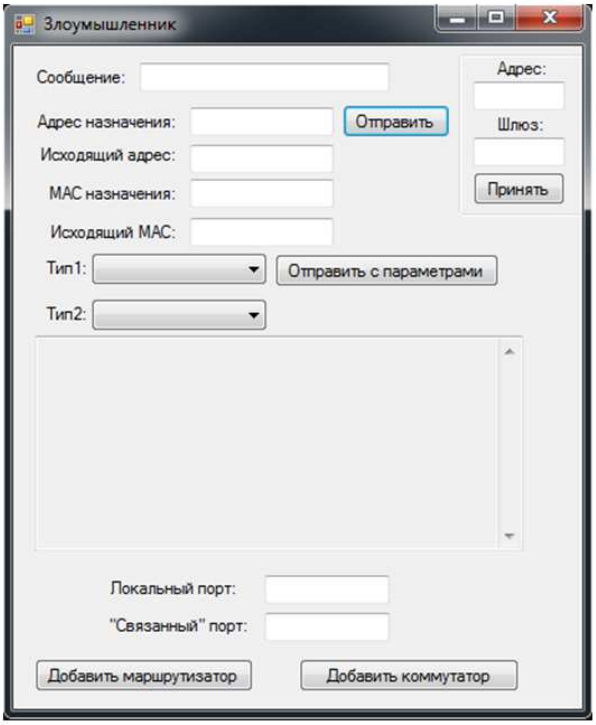

Рис. 1. Стартовое окно (оно же – компьютер злоумышленника)

Решение начинать с него обусловлено тем, что в сети такое будет одно. В окне «Злоумышленник» имеется 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» — наличие кнопки «Отправить с параметрами», позволяющей, используя введённые пользователем данные, передать сообщение с произвольными адресами назначения и отправки, а также выбранного типа. Чтобы добавить в сеть новое устройство (оно будет сразу подключено к компьютеру «проводом»), надо в два поля ввести трёхзначные адреса, первый — порта компьютера с которого идёт подключение, второй – подключённого «проводом» порта второго устройства, затем нажать «Добавить маршрутизатор» или «Добавить коммутатор». Функция открытия нового окна «Маршрутизатор» с нужными параметрами:

```
string inPort = ;
string outPort = ;
if (inPortBox.Text.Length != 3 || outPortBox.Text.Length != 3)
{
MessageBox.Show("Порты должны состоять из трех цифр");
return;
}
        //порт в родительском окне
inPort = inPortBox.Text;
```

```
        //порт в подключаемом окне
outPort = outPortBox.Text;
        //открытие порта
Open(inPort);
oppPort = outPort;
localPort =  + inPort;
this.Text +=  + inPort;
CreateRouter.Visible = false;
inPortBox.Visible = false;
outPortBox.Visible = false;
label2.Visible = false;
label3.Visible = false;
CreateSwitch.Visible = false;
        //добавление нового окна и передача ему номеров портов
        //(имитация подключения проводом)
Form Form2 = new Router(outPort, inPort);
Form2.Show();
```

В открывшемся окне (и в маршрутизаторе, и в коммутаторе) появляется новая возможность — добавить в сеть новый компьютер. Принцип её действия аналогичен только что рассмотренному.

Машина, добавленная с её помощью, сразу соединится с родительским устройством (с тем, с которого она была открыта), а также не будет иметь возможность добавлять новые узлы в сеть, так как будет являться по сути одной из конечных точек сети.

3. Smurfing

Для демонстрации взрывного увеличения сетевого трафика в программу на компьютер злоумышленника добавлена возможность отослать пакет ICMP ECHO REQUEST.

На компьютеры, в свою очередь, добавлена функция обработки этого условного пакета и отправки в ответ ICMP ECHO REPLY.

При отправке с компьютера злоумышленника кнопкой «Отправить с параметрами» можно выбрать данные как на рисунке (см. рис. 2), тогда после первой отправки инициализируется ARP–знакомство хостов, а после второй уже пойдут пакеты ICMP ECHO REPLY на машину жертвы (см. рис. 3).

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

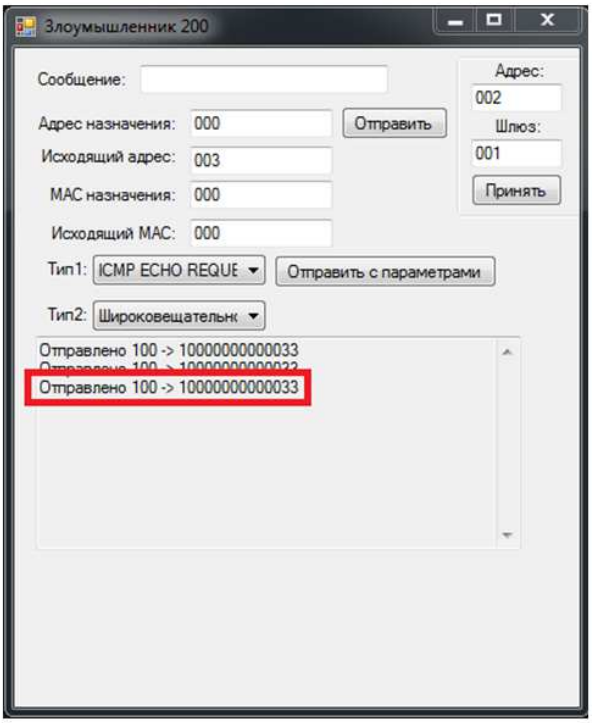

Рис. 2. Отправка широковещательного запроса ECHO REQUEST

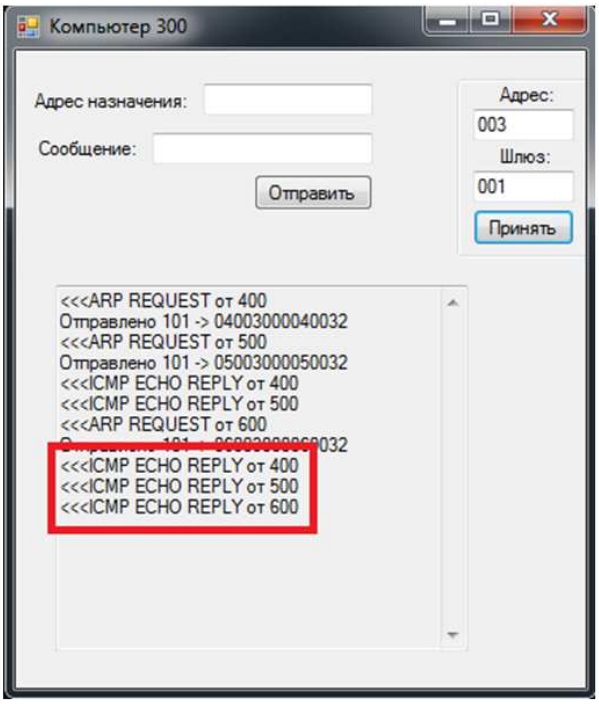

Рис. 3. Сообщения ECHO REPLY на машине жертвы

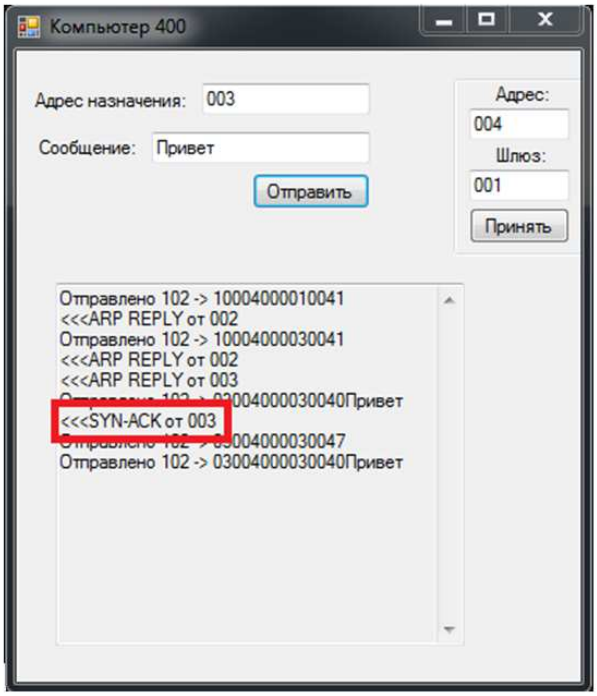

Рис. 4. Ответ SYN-ACK от сервера

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);
      }
}
```

4. SYN-Flood

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».

В обычной ситуации сервер отвечает пакетом SYN-ACK на все новые подключения (см. рис. 4). Но если злоумышленник начнёт непрерывно посылать с 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 серьезными.

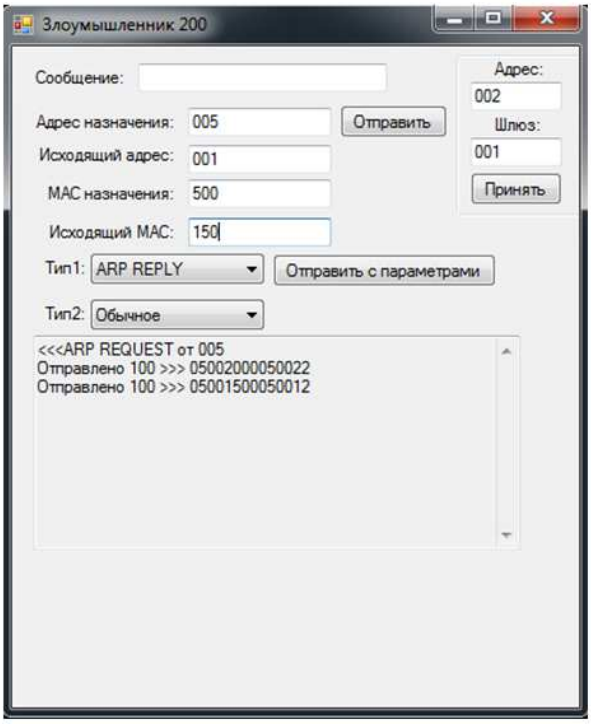

Рис. 5. Отправка поддельного ARP-Reply

5. Подмена MAC-адреса

Чтобы продемонстрировать возможности 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 имитирующий интернет. Послание дойдёт до адресата. Затем отправим от злоумышленника ARP-Reply с нужными параметрами (см. рис. 5), 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. 6) не увенчается успехом (см. рис. 7). Теперь жертва отрезана от интернета.

6. ARP-spoofing

Демонстрация 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.

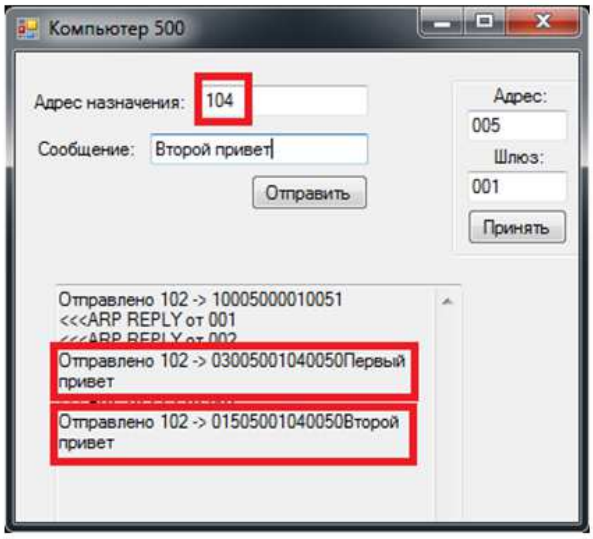

Рис. 6. Отправка двух сообщений

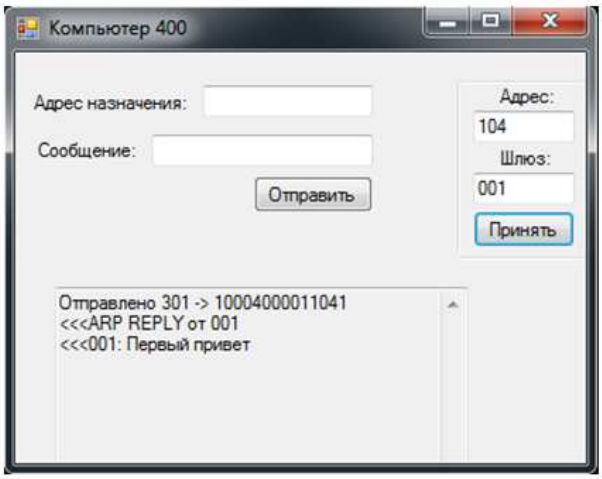

Рис. 7. Получение первого сообщения

Затем посылается поддельный пакет ARP–Reply жертве, которая, сопоставив IP из пакета с IP роутера, заменяет его MAC-адрес на MAC-адрес злоумышленника.

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. 8), но не напрямую, а предварительно пройдя через машину злоумышленника (см. рис. 9).

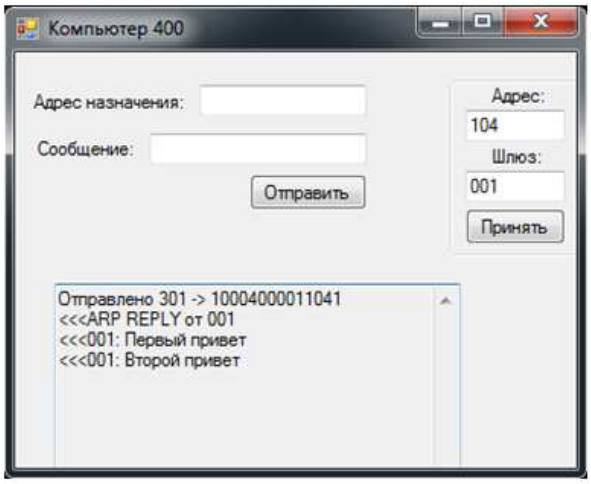

Рис. 8. Получение обоих сообщений

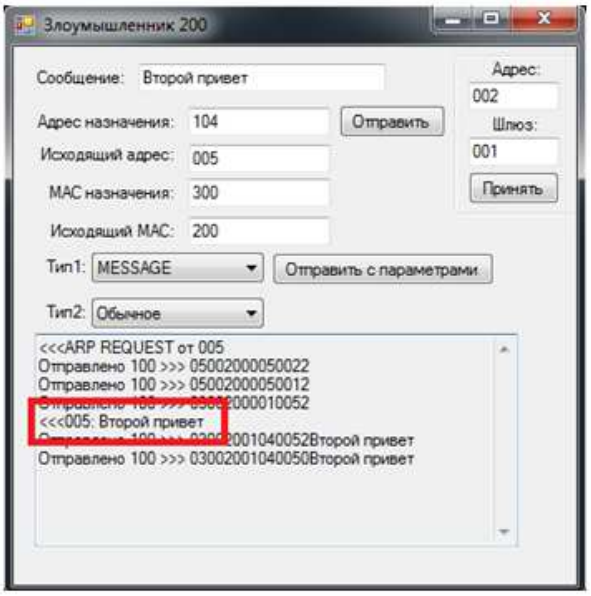

Рис. 9. Получение сообщения злоумышленником

Заключение

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, 2]. Приложение не предусматривает демонстрацию защиты от сетевых атак.

Литература

1. Гуц А.К., Эннс Е.П. Программа, моделирующая компьютерную сеть и сетевые атаки // Математические структуры и моделирование. 2017. № 3(43). С. 139–149.
2. Гуц А.К., Баженов А.В. Программное обеспечение для моделирования сетей и имитации атак на компьютерные сети // Математические структуры и моделирование. 2018. № 4(48). С. 99–112.

THE PROGRAM THAT SIMULATES NETWORK ATTACKS ON COMPUTERS

A.K. Guts
Dr.Sc. (Phus.-Math.), Professor, e-mail: aguts@mail.ru
O.V. Matushina
Student, e-mail: vladimirova.o94@gmail.com

Dostoevsky Omsk State University, Omsk, Russia

Abstract. The purpose of the article is to present a computer program that simulates the operation of a computer network, and also demonstrates the results of using some network vulnerabilities by an unfair user.

Keywords: computer network, network attacks, network vulnerabilities, network attack imitation.

References

1. Guts A.K. and Enns E.P. Programma, modeliruyushchaya komp'yuternuyu set' i setevye ataki. Matematicheskie struktury i modelirovanie, 2017, no. 3(43), pp. 139–149. (in Russian)
2. Guts A.K. and Bazhenov A.V. Programmnoe obespechenie dlya modelirovaniya setei i imitatsii atak na komp'yuternye seti. Matematicheskie struktury i modelirovanie, 2018, no. 4(48), pp. 99–112. (in Russian)

Дата поступления в редакцию: 03.03.2019