RESEARCHERID
THOMSON REUTERS

[Celiktas* *et al.*, 7(8): August, 2018]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

MAN IN THE MIDDLE (MITM) ATTACK DETECTION TOOL DESIGN

Baris Celiktas[*1], Mevlut Serkan TOK [2], Nafiz UNLU [3]

[1]Cyber Security Engineering and Cryptography Department, Institute of Informatics, ITU, Istanbul, Turkey

[2]Computer Engineering Department, Institute of Engineering and Science, TOBB ETU, Ankara, Turkey

[3]Cyber Security Engineering and Cryptography Department, Institute of Informatics, ITU, Istanbul, Turkey

DOI: 10.5281/zenodo.1336698

ABSTRACT

Man in the Middle (MITM) attack is aimed at seizing data between two nodes. The ARP Spoofing/Poisoning technique is a technique frequently used by attackers which allows MITM attacks to be carried out on local area networks (LANs). The attack exploits the vulnerabilities of ARP, which is a stateless protocol. This attack, which is carried out quite simply and quickly, can induce a high level of threats on the targets. Therefore, it is crucial to be protected against this type of attack. Many medium-sized and large-sized enterprises are generally not exposed to this attack because of reliable network infrastructures and commercial software they utilized against MITM attacks. However, small-sized enterprises and individuals are frequently exposed to this threat. The purpose of this study is to design a simple, fast and reliable MITM attack detection tool for LAN users who are often exposed to the threat.

In our study, we present the basic characteristics of the MITM Detection Tool, which detects ARP Spoofing/Poisoning attacks on clients. Based on the findings from the comprehensive review of related literature and tests performed on Kali Linux and Windows 7 OS machines, how MITM attacks can be performed and detected are described.

KEYWORDS: MITM Attack, ARP Spoofing, ARP Poisoning, MITM Attack Detection.

I. INTRODUCTION

MITM attacks are one of the well-known cyber-attack methods which are frequently used. In MITM, the attacker seizes the data transmitted in the process of data transfer [1]. It is named after a scenario, which is trained by the players in basketball practice. According to this scenario, while the two players are passing to each other, the opponent player tries to intercept the ball from the other players without being noticed. MITM attack is also known as “Bucket Brigade Attack”, “Fire Brigade Attack”, “Monkey in the Middle Attack”, “TCP Hijacking” and “TCP session hijacking” [2]. In addition, MITM attacks are basically carried out in four ways as Deceptive Base Station Attacks, SSL / TLS based attacks, BGP based attacks and False Base Station [3].

II. DEFINITIONS

What is ARP?

As a simple definition, Address Resolution Protocol (ARP) is a protocol used to match IP addresses and Media Access Control (MAC) addresses within a LAN [4]. A machine within a LAN communicates with another machine through MAC addresses. According to the OSI Reference model, all devices of Layer 2 and higher ones write MAC addresses to their ARP Tables which they have learned in the local network they are on. Sample ARP Tables for Computers, Routers and Switches are shown in Figure 1, Figure 2 and Figure 3.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
C:\Users\IEUser>arp -a

Interface: 10.0.2.7 --- 0xe
  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     dynamic
  10.0.2.3              08-00-27-4a-51-0f     dynamic
  10.0.2.15             08-00-27-59-1b-51     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 1: A Sample ARP Table of A Computer.

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0001.427b.ead2    DYNAMIC     Fa0/1
   1    0001.96a5.a493    DYNAMIC     Fa0/3
   1    0007.ec50.2490    DYNAMIC     Fa0/4
   1    0060.3e83.9819    DYNAMIC     Fa0/2
```

Figure 2: A Sample ARP Table of a Switch

```
Router#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  40.0.0.1               -   00E0.F7E0.3B88  ARPA   FastEthernet0/0
Internet  192.168.1.1            -   0001.427B.EAD2  ARPA   FastEthernet1/0
```

Figure 3: A Sample ARP Table of a Router

Machine A is firstly required to learn whether it is on the same network as the target machine (Machine B) before communicating with Machine B. Thus, Machine A subjects the IP address of the Machine B to the logical "and" process with the subnet mask and learns whether it is on the same LAN as Machine B.

If the Machine A wishes to communicate with Machine B on a different LAN, it forwards the package directly to gateway, which provides the egress address. Therefore, it uses the MAC address of the gateway as the target MAC address.

Machine A must recognize the MAC address of this device to communicate with Machine B on the same LAN. If it does not recognize the MAC address of Machine B, it broadcasts an ARP Request message to all devices on the network, including the IP address of the Machine B. Machine B detects its IP address registered in the target IP section of the ARP Request packet and responds to Machine A with a unicast ARP Reply packet containing its own MAC address. Machine A also saves the MAC address of Machine B in its own ARP table [5]. Figure 4 shows a sample of an ARP Request Packet and Figure 5 shows a sample of an ARP Reply Packet.

```
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: PcsCompu_59:1b:51 (08:00:27:59:1b:51)
    Sender IP address: 10.0.2.15
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.2.7
```

Figure 4: A sample of an ARP Request Packet Content.

* The target MAC address is unrecognized, while the target IP address is recognized.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: PcsCompu_90:88:cf (08:00:27:90:88:cf)
    Sender IP address: 10.0.2.7
    Target MAC address: PcsCompu_59:1b:51 (08:00:27:59:1b:51)
    Target IP address: 10.0.2.15
```

Figure 5: A Sample ARP Reply Packet Content.

ARP Spoofing/Poisoning

ARP is a protocol that relies on the good faith of all users by its nature [6]. ARP is a stateless protocol that handles every request or reply regardless of past communication. For this reason, any layer 2 or higher device has no mechanism to control the integrity and authentication of ARP Request or Reply packets [7]. It is called ARP Spoofing/Poisoning when an attacker manipulates ARP tables of target device or devices through fake ARP packets. By doing so, all data flow between two devices can be tracked and a successful MITM attack can be carried out. MITM attacks using ARP Spoofing / Poisoning technique are examined in two different categories as deception of users and deception of gateway [8].

Seizure of Communication by Deceiving Users

Figure 6 shows the MITM attack mechanism for the seizure of communication by deceiving users through the ARP Spoofing/Poisoning technique. The attacker manipulates ARP Tables of Machine A and B through ARP Reply packets which he/she has simultaneously sent. After the modification in the ARP Table of Machine A and B, it enables all traffic between devices with IP address 10.0.0.10 and 10.0.0.7 should be transmitted through the attacker's device. Thus, the attacker machine is able to perform many dangerous attacks, such as forwarding the packets transmitted between Machines A and B to other parties by modifying their content, as well as sniffing the packets.

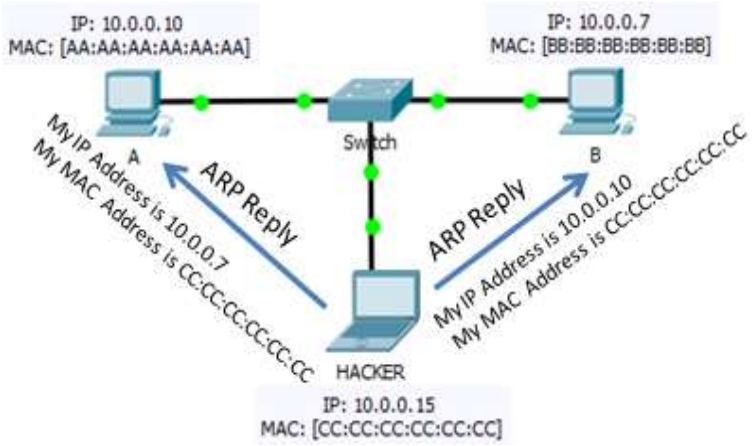

Figure 6: Sample MITM Attack by Deceiving Users.

Seizure of Communication by Deceiving Gateway

Figure 7 displays the attack mechanism for the seizure of communication by deceiving gateway through the ARP Spoofing/Poisoning technique. The attacker manipulates ARP tables of the target machine and gateway by introducing itself to the target machine as if it was a gateway and to the gateway as if it was the target, through the ARP Reply packages attacker has produced. Since all traffic from the target machine to the Internet or from the Internet to the target machine passing through the attacking device, all packets can be sniffed and modified by the attacker.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

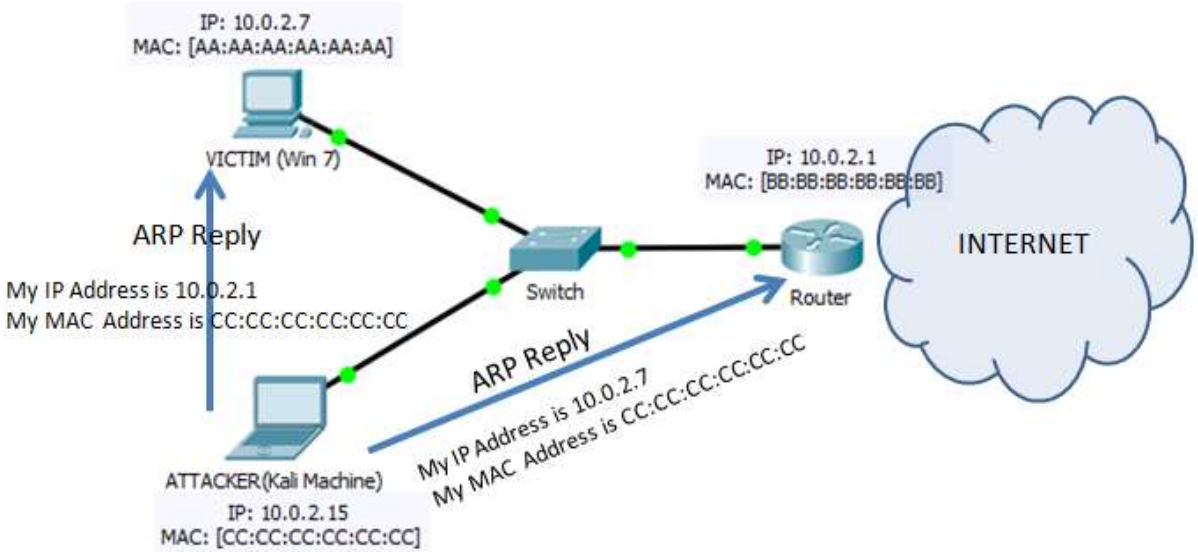

Figure 7: Sample MITM Attack by Deceiving Gateway.

III. THE SIMULATION OF AN ATTACK

Kali Linux and Windows 7 Virtual Machines were used to display a Sample Attack. Firstly, active information gathering methods have been used to determine the nodes in the network which we are on as an attacker, and the machines in our network have been determined as in Figure 8 and Figure 9.

```
root@kali:~# nbtscan -r 10.0.2.0/24
Doing NBT name scan for addresses from 10.0.2.0/24

IP address       NetBIOS Name     Server    User             MAC address
-------------------------------------------------------------------------------
10.0.2.0         Sendto failed: Permission denied
10.0.2.7         IE10WIN7         <server>  <unknown>        08:00:27:90:88:cf
10.0.2.15        <unknown>                  <unknown>
10.0.2.255       Sendto failed: Permission denied
```

Figure 8: Network Scanning with Nbtscan.

```
root@kali:~# netdiscover -r 10.0.2.0/24

 Currently scanning: Finished!   |   Screen View: Unique Hosts

 4 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 240

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
10.0.2.1        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.2        52:54:00:12:35:00      1      60  Unknown vendor
10.0.2.3        08:00:27:4a:51:0f      1      60  PCS Systemtechnik GmbH
10.0.2.7        08:00:27:90:88:cf      1      60  PCS Systemtechnik GmbH
```

Figure 9: Network Scanning with Netdiscover.

The machine with IP address “10.0.2.7”, MAC address “08:00:27:90:88:cf” and “WIN7 OS” were selected as the target machine to attack with the ARP Spoofing technique.

Figure 10 shows the target machine’s ARP Table prior to the attack so that the changes can be seen after the attack.

```
C:\Users\IEUser>arp -a

Interface: 10.0.2.7 --- 0xe
  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     dynamic
  10.0.2.3              08-00-27-4a-51-0f     dynamic
  10.0.2.15             08-00-27-59-1b-51     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 10: WIN7 Machine's ARP Table Prior to the Attack.

As shown in Figure 11 and Figure 12, two different terminals have been initialized, and "ARP Spoofing" attack has been carried out on the gateway (10.0.2.1) and WIN7 Machines (10.0.2.7). As seen, the attacker introduces his/her own MAC address to the gateway as a WIN7 machine and to the WIN7 machine as a gateway through the ARP Reply packets that are constantly forwarded.

```
root@kali:~# arpspoof -i eth0 -t 10.0.2.1 10.0.2.7
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 52:54:0:12:35:0 0806 42: arp reply 10.0.2.7 is-at 8:0:27:59:1b:51
```

Figure 11: Modification of GATEWAY's ARP Table through an ARP Spoof Attack.

```
root@kali:~# arpspoof -i eth0 -t 10.0.2.7 10.0.2.1
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
8:0:27:59:1b:51 8:0:27:90:88:cf 0806 42: arp reply 10.0.2.1 is-at 8:0:27:59:1b:51
```

Figure 12: Modification of WIN7 Machine's ARP Table through an ARP Spoof Attack.

During communication, the ARP Tables of the WIN7 machine and gateway have been modified through ARP Reply packages we have sent. The new ARP Table of the WIN7 Machine is shown in Figure 13. The significant point here is that the MAC address which gateway has is the same as the MAC address the attacker owns.

```
C:\Users\IEUser>arp -a

Interface: 10.0.2.7 --- 0xe
  Internet Address      Physical Address      Type
  10.0.2.1              08-00-27-59-1b-51     dynamic
  10.0.2.3              08-00-27-4a-51-0f     dynamic
  10.0.2.15             08-00-27-59-1b-51     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 13: WIN7 Machine's ARP Table After Attack.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

At the end of the attack, a Web Form using the HTTP protocol created in Figure 14 for testing purposes has been used to indicate one of the possible threats to the target machine. In Figure 15, all the information that the target machine user has entered in the web page in Figure 14 appears to be read clearly by the Attacking Machine with the Wireshark Packet Analysis Tool.

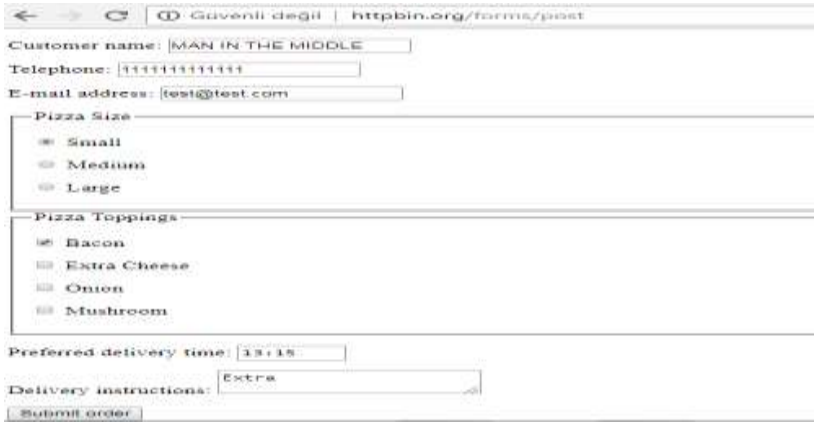

Figure 14: Sample HTTP Form Page.

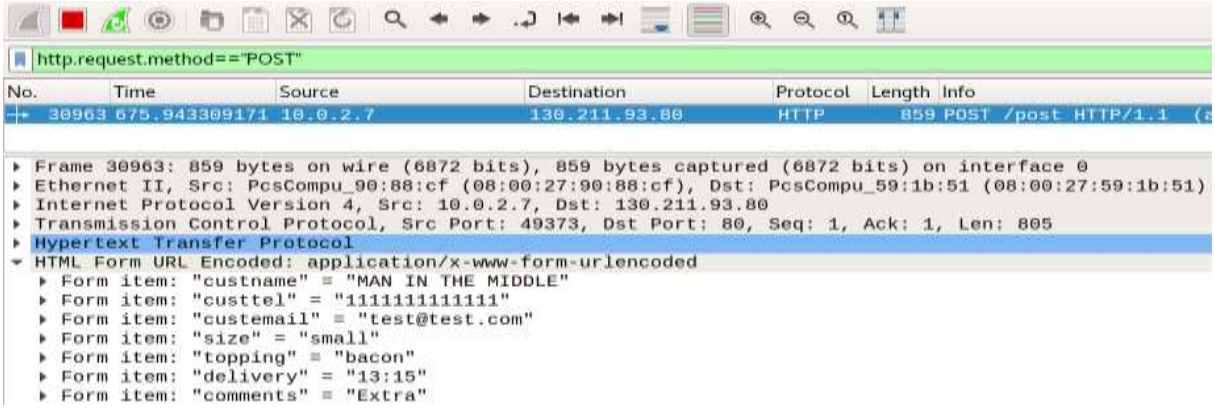

Figure 15: Image of Pages Opened Using HTTP.POST Method on Target Machine.

With the MITM attack, many threats can occur as in Figure 15. It has also been demonstrated in our study how important it is to attack as well as how easy it is.

IV. ARP SPOOFING DETECTION METHODS

It is possible to detect an ARP spoofing attack without the need for additional software if the Port Security settings in the switches and the configuration of network devices are adjusted accurately [9]. Today, however, the use of third party software is widespread for the detection of an ARP Spoofing attack. This section of our study briefly describes the detection methods used to be protected against MITM attacks.

Using Static ARP Tables

The creation of MAC addresses by manually generating the table in the ARP cache memory of each device prevents deception. Although it seems to be preferred in small local networks, it is not a feasible method in large networks because of the workload that often occurs in networks which mobile devices such as laptop computers are frequently connected [10].

Secure ARP (S-ARP)

Secure-ARP is based on the principle of utilizing an asymmetric encrypting (DSA) in order to ensure the integrity and authenticity of ARP messages by adding an additional header to the ARP, which is considered due to the use of the asymmetric encrypting [11].

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Ticket based ARP (TARP)
Certified MAC/IP mappings are generated by the server called Local Ticket Agent (LTA) using ARP messages that are available on the network and verified. These verified mappings are called tickets. These tickets are encrypted and distributed to each connected device by LTA with the LTA public key. The connected device opens the ticket using the LTA public key and it is used for recording the MAC/IP mappings. The use of a single key for encryption allows the cost to be kept lower than S-ARP [12].

Stateful ARP
The most basic weakness of ARP is that it is stateless and takes into consideration any request or reply regardless of past communication [13]. In stateful ARP, each device stores ARP requests in its memory and separate queues for ARP replies. When an ARP reply is received, the device checks if there is a relevant ARP request before this reply. If there is no ARP request for this reply, it assumes that the reply message has been forwarded from the attacker and drops the ARP message [14].

MITM-Resistant (MR) ARP
MR ARP is based on managing a set of IP/MAC mappings of all active devices in the same local network via a long-running table. If Device A correctly recognizes the IP/MAC address mapping of Device B and keeps saving this match for as long as Device B is active, MITM and ARP poisoning between A and B is not possible. The IP addresses, MAC addresses and time counter value assigned to each device are saved in a created table. As long as the devices are active on the local area network, the MAC/IP mappings to be used in the communication are carried out using this table [15].

Kernel-Based Patches
By means of patches such as Anticap and Antidote, devices with Linux-based operating systems can be individually protected for ARP Spoofing. Anticap prevents an ARP reply message with a MAC address different from the cache memory from updating the ARP cache memory and drops this message [16]. If an incoming ARP reply message has a different MAC address than the one in the ARP cache memory, the device with the MAC address in the cache memory is checked to see if it is active. If the device is is active, the incoming ARP reply message is dropped and the MAC address which sends the ARP message is placed on the block list [17]. In both patches, it is assumed that the past IP/MAC mappings saved in the cache are correct. If the attacker can get his/her fake ARP entry registered in cache memory before the legal user, the relevant kernel patches cannot detect ARP spoofing [18].

Dynamic ARP Inspection
First, to explain the DHCP Snooping, it is a method that is used to prevent devices connected to switches, which automatically receive IP, from receiving IP from unauthorized DHCP servers [19]. Dynamic ARP Inspection (DAI) is a security feature that verifies ARP packets on the network. In the new generation switches, DAI examines records and drops ARP packets which contain invalid IP/MAC address mappings. While carrying this out, a reliable database, formed by utilizing DHCP Snooping, is used [20].

V. DESIGN OF THE MITM DETECTION TOOL DESIGN AND TESTING

ARP spoofing detection methods which we have found in our literature research are not frequently used in places such as houses, offices, open areas where the networks are not served on reliable network devices. Therefore, there is a need for a detection and prevention system against MITM attacks using the ARP Spoofing technique in such places.

It is especially crucial to monitor internet traffic and to prevent MITM attacks, which have been carried out by deceiving gateway, where much personal information has been seized. Hence, the ARP Tables of the computers are required to be continuously monitored. It is hardly possible to perform this manually. Thus, the MITM Detection Tool has been written using the Python script language which can be used on the LAN to facilitate the users' task.

Coding

```
import os
import time
```

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
import ipaddress
import struct
from IPy import IP
from uuid import getnode as get_mac
from subprocess import Popen, PIPE
import re
import socket
import csv
import subprocess
import sys

print "MAN IN THE MIDDLE ATTACK DETECTION TOOL"

# Get ARP table
def get_arp_table(IP):
    cmd = "arp -a"
    process = subprocess.Popen(cmd, stdout=subprocess.PIPE, stderr=None, shell=True)
    output = process.communicate()
    interfaceBlocks = output[0].split('Interface:')

    for interfaceBlock in interfaceBlocks:
        lines = interfaceBlock.split('\r\n');
        interfaceIP = lines[0].split('---')[0].strip();

        if interfaceIP == IP:
            names = ['Internet Address', 'Physical Address', 'Type']
            reader = csv.DictReader(
                lines[1:], fieldnames=names, skipinitialspace=True, delimiter=' ')
            next(reader)

            return [block for block in reader]

# Get IP Address
def get_ip():
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        # doesn't even have to be reachable
        s.connect(('10.255.255.255', 1))
        IP = s.getsockname()[0]
    except:
        IP = '127.0.0.1'
    finally:
        s.close()
    return IP

IP = get_ip()

arp_table = get_arp_table(IP)

mac_address = arp_table[0]['Physical Address']
mitm = False

print "Gateway's MAC Address is ", mac_address, "\r\n"

print "Looking for MITM"
for line in arp_table[1:]:
    if line['Physical Address'] == mac_address:
```

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

```
        mitm = True

if mitm:
    print "MITM Attack is Detected"
else:
    print "No MITM Attack Right Now"
```

Test Environment

A machine with IP address 192.168.10.20 has been selected to test the operation of our tool and ARP Tables are shown in Figure 16 and Figure 17.

```
Interface: 192.168.1.20 --- 0x8
  Internet Address      Physical Address      Type
  192.168.1.1           58-2a-f7-9a-08-5b     dynamic
  192.168.1.21          08-00-27-59-1b-51     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 16: The ARP Table Before Exposed to ARP Spoofing Attack.

```
Interface: 192.168.1.20 --- 0x8
  Internet Address      Physical Address      Type
  192.168.1.1           08-00-27-59-1b-51     dynamic
  192.168.1.21          08-00-27-59-1b-51     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Figure 17: The ARP Table After Being Exposed to a ARP Spoofing Attack.

The MITM Detection Tool is operated and Figure 18 shows the detection image of the MITM attack. It has been found that the tool which we have developed in the test has detected MITM attacks rather quickly and efficiently.

```
C:\ProgramData\Anaconda2\python.exe C:/Users/brscl/PycharmProjects/MITMDetection/mitm_detection.py
MAN IN THE MIDDLE ATTACK DETECTION TOOL
Gateway's MAC Address is  08-00-27-59-1b-51

Looking for MITM
MITM Attack is Detected

Process finished with exit code 0
```

Figure 18: Detection of MITM Attack..

VI. CONCLUSION

This study has helped us to work on the need to take action against MITM attacks, which are often used by attackers. Then, our aim was to create a MITM Detection Tool which detects ARP Spoofing Attacks on LANs. This study suggests a better perspective to the users, software developers, and security administrators about the key features of the MITM Detection tool that can be used as a solution. This tool does not require high fees for third party software and provides the users with flexibility in taking measures against MITM attacks using ARP Spoofing/Poisoning technique. We believe that software developers will be able to design more powerful and

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

better MITM Attack Detection tools. We also suppose that our study will be a guide for future academic studies on MITM and ARP Spoofing Attacks.

REFERENCES

1. Conti M, Dragoni N, Lesyk V. A survey of man in the middle attacks. IEEE Communications Surveys Tutorials. 2016; 18(3): 2027-2051
2. Nayak GN, Samaddar SG. Different flavours of man-inthe-middle attack, consequences and feasible solutions. 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT). 2010; 5: 491-495
3. Bhushan, B, Sahoo G, Rai AK. Man-in-the-middle attack in wireless and computer networking; A review. 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA) (Fall). 2017:1-6.
4. Plummer DC. An Ethernet address resolution protocol-RFC 826 [online] 1982 [cited 2018 Jun 19] Available from URL: https://tools.ietf.org/html/rfc826
5. Kurose JF, Ross KW. "The Link Layer: Links, Acess Networks, and LANS". In: Computer Networking A Top-Down Approach. 6th ed. Pearson Education: New Jersey; 2013: 465-469
6. Abad LC, Bonilla RI. An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks. 27th International Conference on Distributed Computing Systems Workshops. 2007; 60: 60-67
7. Alharbi T, Durando D, Pakzad F, Portmann M. Securing ARP in Software Defined Networks. 2016 IEEE 41st Conference on Local Computer Networks (LCN). 2016; 523-526.
8. Shukla S, Yadav I. An innovative method for detection and prevention against ARP spoofing. International Journal of Computer Science and Information Technology & Security. 2015; 5(1): 207-214
9. Carnut M, Gondim J. ARP Spoofing Detection on Switched Ethernet Networks: A Feasibility Study. [online]. 2003 [cited 2018 Jun 09] Available from URL: http://www.postcogito.org/Publications InEnglish/arpspoof-detection-v10final2.pdf
10. Demuth T, Leitner A. ARP Spoofing and Poisoning: Traffic Tricks. Linux Magazine. 2005; 56: 26-31
11. Bruschi D, Omaghi A, Rosti E. S-ARP: A Secure Address Resolution Protocol. 19th Annual Computer Security Applications Conference, 2003. Proceedings. 2003; 66-74.
12. Lootah W, Enck W, McDaniel P. TARP: Ticket-based Address Resolution Potocol. 21st Annual Computer Security Applications Conference (ACSAC'05). 2005; 116-124.
13. Bakhache B, Rostom R. Kerberos secured Address Resolution Protocol (KARP). *5th International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. 2015; 210-215.
14. Trabelsi Z, El-Hajji W. Preventing ARP attacks using a Fuzzy-based Stateful ARP Cache. 2007 IEEE International Conference on Communications. 2007; 1355-1360.
15. Nam SY, Kim D, Kim J. Enhanced ARP: Preventing ARP poisoning-based man-in-the-middle attacks. IEEE Community Letters. 2010;14(2):187-189.
16. Barnaba M. Anticap [online]. 2003 [cited 2018 Jun 01] Available from URL: https://github.com /antifork/anticap
17. Teterin I. Antidote. [online]. 2002 [cited 2018 Jul 10], Available from URL: https://www. securityfocus.com/archive/1/299929
18. Ramachandran V, Nandi S. Detecting ARP Spoofing: An Active Technique. In: Jajodia S., Mazumdar C. (eds) Information Systems Security (ICISS 2005) Lecture Notes in Computer Science. 2005; 3803: 239-250
19. Mangut AH, Al-Nemrat A, Benzaid C, Tawil ARH. ARP Cache Poisoning Mitigation and Forensics Investigation. IEEE Trustcom/BigDataSE/ISPA. 2015; 1: 1392-1397
20. Cisco. Catalyst 6500 Guide Book [online]. 2009 [cited 2018 May 29] Available from URL: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SXF/native/configuration/ guide/swcg.pdf

CITE AN ARTICLE

Celiktas, B., Tok, M. S., & Unlu, N. (2018). MAN IN THE MIDDLE (MITM) ATTACK DETECTION TOOL DESIGN. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*,*7*(8), 90-100.