IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 24, Issue 6, Ser. I (Nov. – Dec. 2022), PP 09-20
www.iosrjournals.org

A survey on Man in the Middle Attack - ARP Spoofing

Srikar Vishnu Datta A[1], Patnam Venkata Koushik[2],
Roopa Thanmayee N[3], Ruby D[4]
[1,2,3] *Students of School of Computer Science and Engineering,VITUniversity, Vellore, Tamil Nadu, India 632014*

Abstract - *The primary goal of any attack is to gain access to the victim's machine without the knowledge or with minimal detection of the victim. The man-In-The- Middle attack is the most common attack that takes advantage of the vulnerability that exists in the ARP protocol. There are various types of MITM attacks and in this experiment, we are going to perform ARP spoofing using the Python Scapy library to poison the ARP table and then sniff the data from the victim machine. MITM attack is a technique that violates the confidentiality and integrity of the communication by inserting a third-party attacker machine in between the communication channel without the knowledge of the client andserver. MITM attacks occur through ARP poisoning through ARP protocol. The Address Resolution Protocol (ARP) is a stateless protocol that maps IP addresses to their corresponding MAC addresses. It maintains a table called ARP cache that keeps a list of MAC addresses and their corresponding IP addresses within a switched LAN network. In this study, we are trying to poison this ARP table and perform the MITM attack to sniff the data of the victim.Web browsers interact using IP addresses and Domain NameService (DNS) helps to resolve the domain names to IP addresses. Attackers always try to find vulnerabilities to attack the websites and any website that is not secured can be easily compromised by the attacker. HSTS (Hyper Text Transfer Protocol Strict Transport Security) is a web security policy introduced to prevent MITM attacks on websites. This HSTS converts all the websites which were trying to be opened insecurely through HTTP to strict HTTPS securely. Inthis study, we are trying to bypass this security by changing the domain names and confusing the HSTS as the rules in HSTS would work for valid Domain names. We are trying to exploit by spoofing/altering the names in such a way that the victim would not notice the difference. This attack can be successful for websites that are opened by searching the keyword in google as they can be redirected to the fake domain names.*

Key Words: Information security, RSA, Man in The Middle (MITM) Attack, ARP Cache poisoning, HSTS, HTTP, DNS, Python, Scapy, etc.

Date of Submission: 27-10-2022 Date of Acceptance: 07-11-2022

I. Introduction

In computer networking, Man in The Middle (MITM) is the most dangerous and common attack. In this study, we are going to explore MITM attacks with ARP spoofing by exploiting vulnerabilities in ARP protocol and also learn preventive measures to protect our networks from such attacks.

Firstly, the protocol does not have a proper authentication mechanism to identify whether the ARP requests and replies are coming from the right hosts. Another vulnerability of ARP is it is a stateless protocol, so each Reply and request from an ARP are viewed independently. So, even if the host doesn't send the request, ARP reply packets may still be transmitted to the host. The protocol also states that if any new entry comes for an already existing entry, then update the ARP cache with the new entry or ARP received. The attacker can easily poison the network by sending or broadcasting ARP reply packets saying he/she is the proper source. Hence, the ARP table of the hosts in the network would be updated with the attacker's MAC address, and this way all the hosts in the LAN will get poisoned.

This survey mainly focuses on Performing ARP spoofing attacks using the Python Scapy library, Poisoning the ARP table, Performing MITM, bypassing HSTS connection, Preventing ARP spoofing or similar attacks using WireShark or other methodologies

II. Literature Survey

Building technologies that provide wireless network protection from intruder attacks is important. In essence, paper [1] addresses methods designed to protect wireless networks from ARP spoofing attacks and lists the benefits and drawbacks of the most popular strategies in comparison to predetermined standards. They have tested many techniques to identify ARP spoofing and have outlined the benefits and drawbacks of each technique. To develop a system that can prevent ARP exploits while keeping in mind the shortcomings of the

previouslypublished approaches, this study may be utilized as a ready reference.

We present a fun method to identify ARP spoofing in the paper [2]. To check for irregularities, we inject TCP SYN packets and ARP requests into the network. Compared to passive approaches, our system is more efficient, intelligent, scalable, and reliable at spotting attacks. Additionally, it has a high degree of accuracy in detecting the crucial mapping of MAC to IP addresses in the event of an actual attack. To sum up, it is a fresh approach that actively engages the network to detect the presence of ARP spoofing assaults. However, it isn't fully developed in a customized stack, and the detection isn't entirely precise.

The study paper [3] presents the frequently used ARP spoofing methods, including interception, malicious attacks, internal and external network sniffing, and more. This proposed a method for IP matching, a method for data monitoring, a method for measuring echo times, a method for analyzing ARP responses, a way for using software tools to discover methods, a new method for caching ARP updates, a method for using switch equipment to control, and other tactics. In addition, a technique to protect networks from ARP spoofing attacksis presented in this research.

The survey study [4] on man-in-the-middle attacks focuses on how man-in-the-middle attacks are implemented on Diffie-Hellman, the many methods used to carry them out, and consequently, the numerous defenses against the assault. Although complete MITM eradication may be challenging, we make every effort to close all possible entry points for attackers looking to assault a network. A variety of countermeasures are frequently employed to deliver updated and patched operating systems that must be utilized on networks; nonetheless, the security of the network should be given priority while constructing it. In this approach, we try to put into practice strategies that will lessen the impact of the MITM attack on us.

A Man In The Middle -Intrusion Detection System model for attack recognition, isolation, and node reconfiguration was proposed in the research paper [5]. The IDS method aids in preparing the nodes for potential attacks. In the WSN setting, the MITM attack might be more likely. The simulation results demonstrate MITM-success IDS in comparison to a non-secure environment. The main advantage of this model is that, due to its simplicity, it can quickly identify malevolent behavior.

A man-in-the-middle attack is the key SSL attack, according to a study report [6]. ARP poisoning and phishing attacks are the primary threats to SSL. Phishing is a social engineering assault that uses phoney sites or certificates tosteal the user's credentials. When using ARP Poisoning, the attacker uses client-server communication as a middleman. The browser plug-in might not be a good solution and will provide better client-side protection to the user, but the shell script verifies the ARP-IP gateway and all other network devices to identify the modifications made in the ARP cache.

A survey of man-in-the-middle (MIM) assaults in communication networks and countermeasures is presented in the study [7]. Timing information can be used in real-time communication to identify the attack in several circumstances. Address Resolution Protocol (ARP) cache poisoning, DNS spoofing, session hijacking, and SSL hijacking are the most common types of assaults. The shell script running at the backend, which can preserve a record of entries within the ARP cache table (maps IP address and MAC address)], is frequently used to prevent ARP poisoning. The issue with this method is that there is a lot of network traffic caused by frequently produced ARP requests, thus the shell script will only function on Linux and not Windows.

This paper[8] suggests a few techniques to identify and prevent ARP spoofing. To map IP addresses to their corresponding MAC addresses, the ARP protocol is widely used on the internet. It lacks authentication, making it vulnerable to an ARP spoofing attack, which is a type of spoofing attack. The spoofing can then result in DoS attacks, Man-in-the-Middle attacks, etc.

III. Attack Approach

In our approach to the above problem statement, we are going to exploit the vulnerability in ARP protocol by poisoning the ARP table. As this attack is possible for devices that are connected to the same LAN i.e., to the same router, we discover our victim's IP using the net. probe functionality and we set the victim's IP as a target for ARP spoofing. After launching ARP spoofing, we can sniff the victim's activities, and data and capture credentials using the net.sniff module. We are further exploiting this attack by bypassing HSTS connections to capture credentials from strict HTTP websites which usually can't be redirected/downgraded to HTTP.

Components

1. Bettercap tool
2. Kali Linux machine (attacker)
3. Windows machine (victim)
4. Router
5. Wireshark

IV. Flowchart

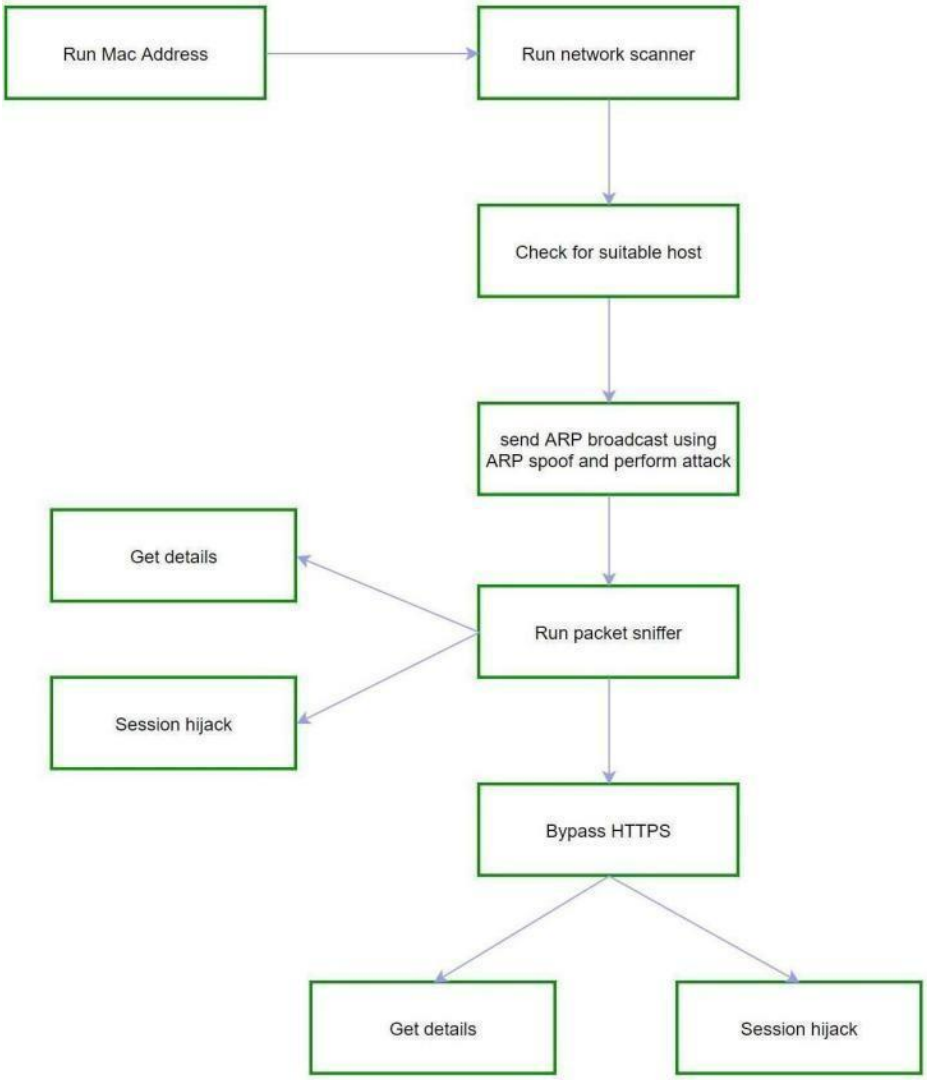
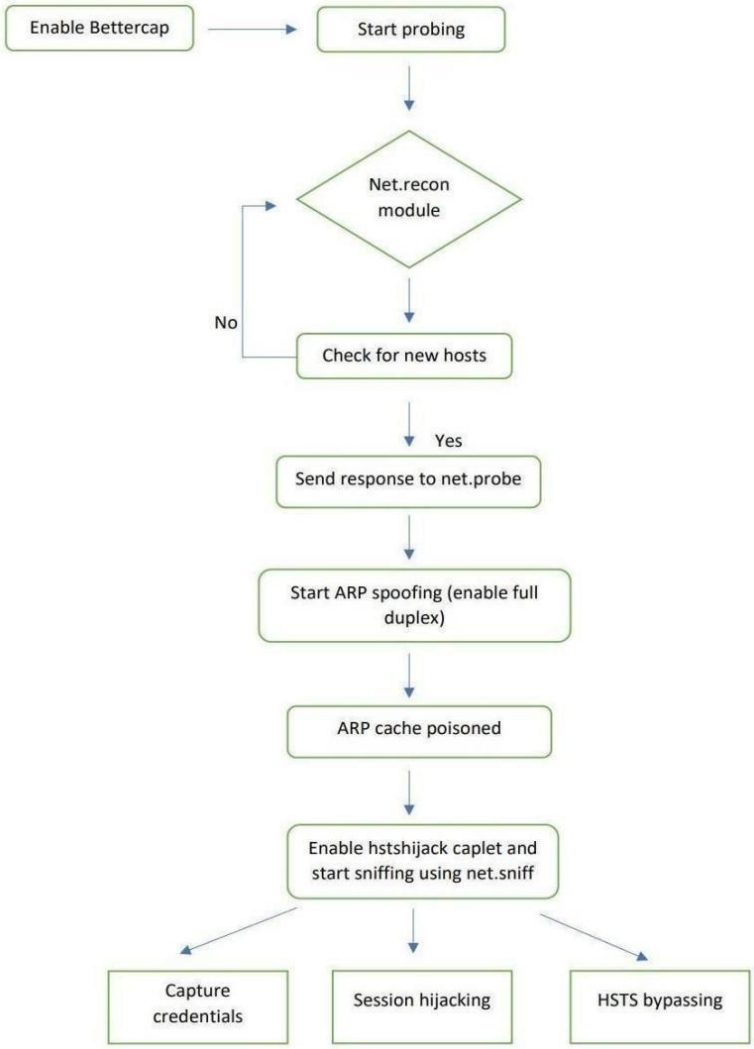


Fig. 4.1: Man in the middle attack using bettercap

The above-mentioned system diagram is the steps or the procedure one can follow using the already built-in tools inkali Linux and taking bettercap as a tool to perform ARP spoofing and using hsts to perform session hijacking or get the user credentials of the users. We can detect ARP spoofing with the help of Wireshark or XARP, which gives the information to the user saying that the user's system isunder attack.

Fig. 4.2 Man in the middle attack using scapy-python



The above-mentioned system diagram mentions how one user can perform ARP spoofing with the help of python scripts developed by us. First, we need to perform a network scan to select a target IP address, and then we can perform an ARP spoof on that particular IP address. After using the ARP spoof python script, with the help of a packet sniffer script, we can get the http requests and also extract usernames and passwords for that particular IP address.

V. Methodology

The approach using Built-in Keywords:

To perform this attack, we are using Kali Linux as an attacker machine and a MITM tool called bettercap for spoofing and sniffing the info also, we are employing a windows machine because of the victim. Before the attack, we'd like to possess the victim's IP address and therefore the network interface to be known. We've created a Bettercap caplet to bypass HSTS. Inside the caplet, we've created spoofed domain names for valid domain names (like Facebook.com for facebook.com) to bypass HSTS protection. Since HSTS may be a mitigation that was introduced to supply security from SSL stripping and it might not allow any insecure connections to websites which must be secure. to bypass this security, we are creating similar domain names and redirecting the usersto those fake domains to capture the info.

Firstly, we'd like to start using the MITM tool bettercap in our kali Linux machine to start the spoofing process. we'll perform ARP spoofing and poison the ARP table within the network. to spot all the hosts, we'd like to send probe packets to all or any of the IPs within the network using the net.probe module. The response will be detected by the internet.recon module which periodically reads the ARP table to see for any new hosts on the network. Then we'd like to enable full duplex ARP spoofing to make the target think that we are the router and therefore the router to think that we are the target. Next, we'll set the ARP target to the victim machine which is windows in our study. Then we've to show spoofing in our attacker machine so that the spoofing module will perform ARP spoofing for the chosen host on our network by sending crafted ARP packets thereby performing the MITMattack. Next, we sniff the traffic with the assistance of the net.sniff module in Bettercap and we can see all the traffic that's passing through the victim machine. However, it's difficult to capture HTTPS and HSTS traffic.

Now, to perform the HSTS bypass, we use the caplet that we've mentioned earlier and replicate it within the path where bettercap stores its caplets and then launch the hstshijack caplet from the terminal. This caplet allows us to urge the small print of the user whoever logins into our spoofed name web applications. This will be achieved by creating insecure google browsers and whenever any user tries to go to secured sites, our caplet will run and redirect the user to the fake sites. The user unknowingly clicks on the webpage which looks exactly just like the trusted website and enters his details. we will gather all the knowledge that the user enters as we've already performed MITM and begun sniffing the info.

Proposed Approach Using Python:

Our approach is to perform the steps mentioned above without using any tools that are already built into the kali machine. The python scripts written by us can be helpful and are easy to understand. These python scripts can be run on any OS and only on Kali Linux. With the help of the codes written in python, we plan on changing the MAC address of the hacker's machine so that the hacker won't be caught. After changing the MAC address, we then need to scan the network of that particular network, so that the hacker can search for a target in the same network. After the hacker has chosen his target IP address, his next aim is to spoof the ARP table of both the target machine and the gateway, so that both machines will not get doubted. After spoofing, both the ARP tables get updated and now the hacker acts as a man in the middle. After spoofing, the hacker can now get access to the user details of the victim's machine. The hacker can also sniff the packets of the victim's machine. With the help of sniffing, the hacker can get HTTP requests, passwords, usernames, and many moredetails.

The python scripts that were prepared by us were MAC address changer, Network Scanner, ARP spoofing, Packet Sniffing, and ARP Spoof detection. In most of these scripts, we used a few major library packages known as Scapy, Optparse, and Subprocess.

A Python application called Scapy gives the user the ability to sniff, send, forge, and dissect network packets. With the help of Scapy, we can construct tools that can be used to scan orattack networks. This is the most popular tool used for interacting with the packets and also manipulating them. Scapy can perform the majority of traditional activities, including tracerouting, scanning, unit testing, probing, network discovery, and attacks, with ease. This can be used as an alternative for nmap, tcpdump, and tshark.

Optparse is a python package used to write command line tools inside the python code. With Optparse, you construct an instance of OptionParser, fill it with options, and then parse the command line in a more declarative manner. Optparse enables users to specify options and needs in GNU/POFIX syntax and returns messages that are helpful to other users.

With the help of the subprocess module, you may launch new processes, join their input/output/error pipes, and get their return codes. The module and its associated functions are meant to take the place of several earlier ones.

To prevent ARP spoofing or similar attacks, we can use Wireshark to monitor suspicious activities by enabling it to Detect ARP storm requests. So, whenever any host tries to discover or sends probe packets, Wireshark will monitor and send an alert to the user. There are also other prevention methods such as using static tables instead of dynamic, using the XARP tool to monitor the ARP cache, using VPNs or encrypted protocols, using packet filtering firewalls, etc.

VI. Results

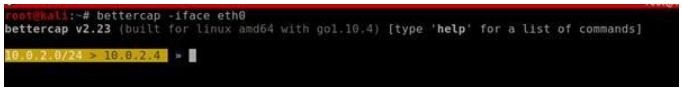


Fig. 6.1.1: Activating the BetterCap tool

Fig. 6.1.2: Discovers the devices in BetterCap

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

Interface: 10.0.2.15 --- 0xa
  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     dynamic
  10.0.2.3              08-00-27-64-c6-75     dynamic
  10.0.2.4              08-00-27-fd-5a-ea     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           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

C:\Users\IEUser>_
```

Fig. 6.1.3: ARP table of a victim (before an attack)

Fig. 6.1.4: Enabling full duplex mode

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

Interface: 10.0.2.15 --- 0xa
  Internet Address      Physical Address      Type
  10.0.2.1              08-00-27-fd-5a-ea     dynamic
  10.0.2.3              08-00-27-64-c6-75     dynamic
  10.0.2.4              08-00-27-fd-5a-ea     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           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

C:\Users\IEUser>_
```

Fig. 6.1.5: Mac address has been spoofed (after attack)

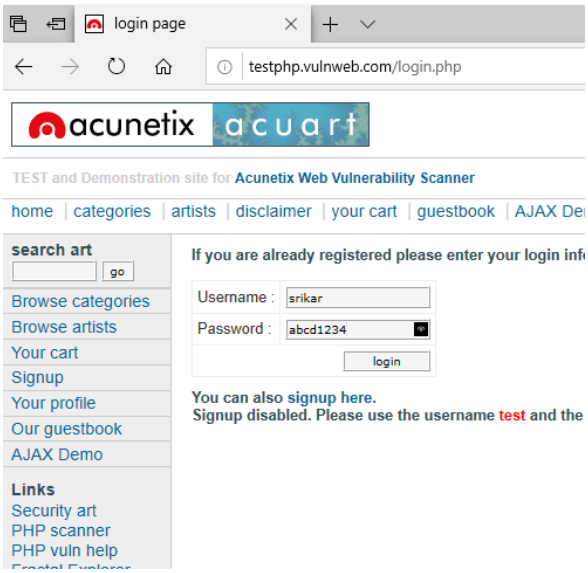


Fig. 6.1.6: Victim login to a website with password“srikar1234”

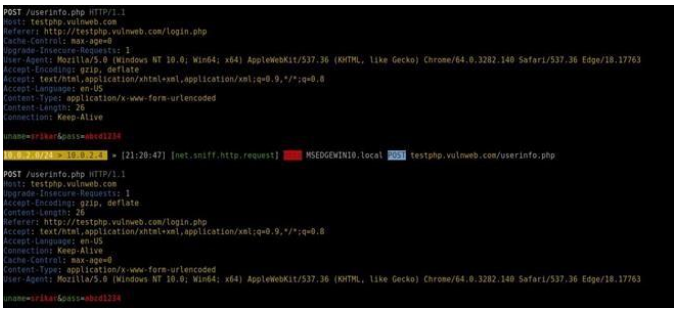


Fig. 6.1.7: Password is sniffed.

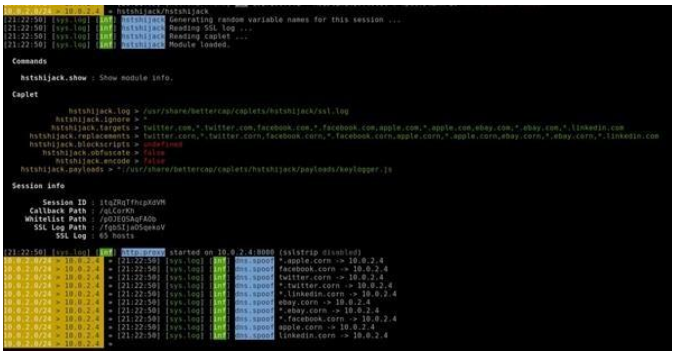


Fig. 6.1.8. Loading the modified caplet ofhstshijack.

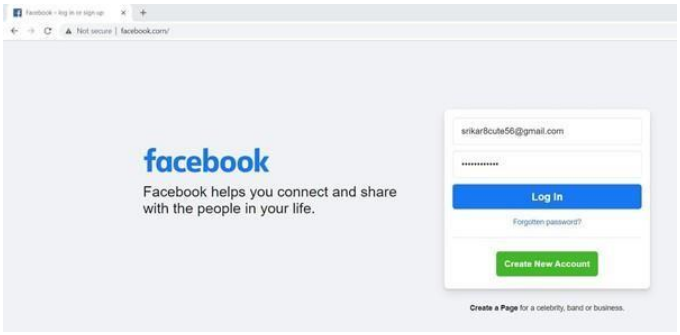


Fig. 6.1.9 Accessing facebook.com (which useshsts) and performing DNS spoofing, observing that the domain has loaded to .corn instead of .com

Fig. 6.1.10 On entering Login, we can see that thelogin details have been captured successfully

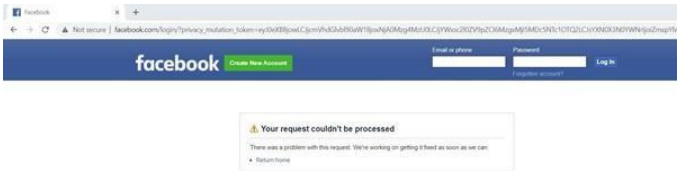


Fig. 6.1.11 Since we created a fake page, it says itcould not process. The victim thinks there might be an issue and tries to access it later.

6.2 Using Python

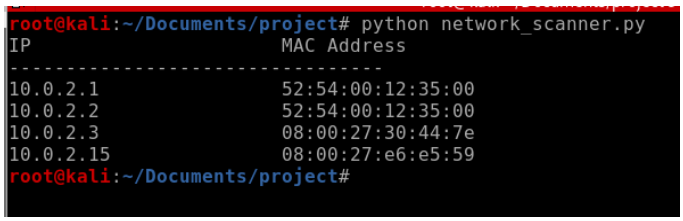


Fig.6.2.1: Checking for a device in the network

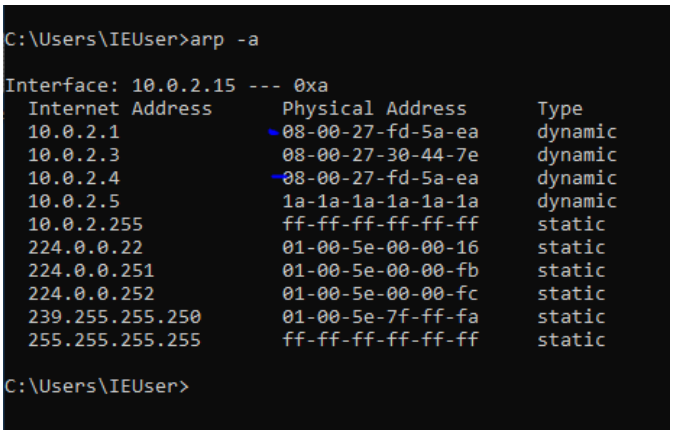


Fig. 6.2.2: ARP of the victim after the attack

Fig. 6.2.3: Running arp spoof

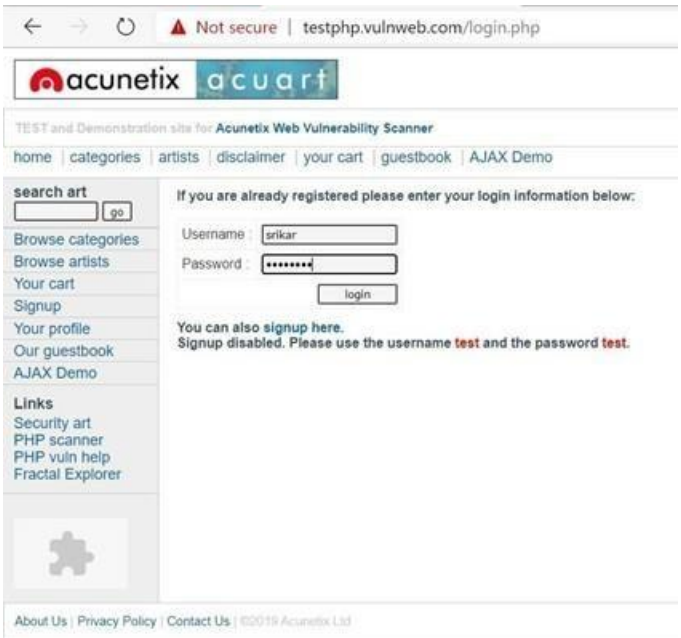


Fig. 6.2.4: Vulnweb.com to test the connection

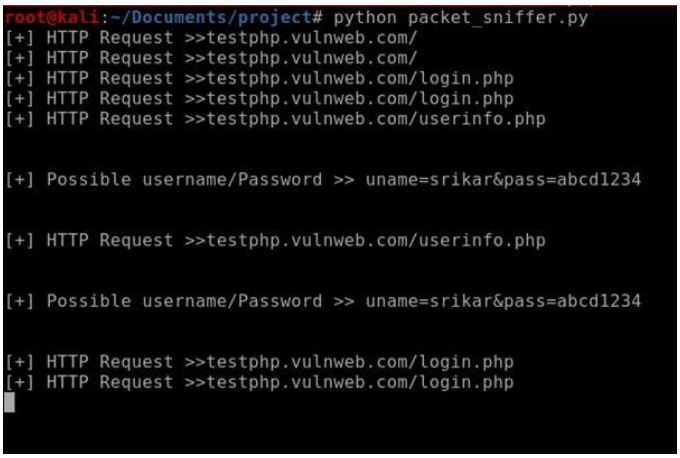


Fig. 6.2.5: Successful capture of data

Fig. 6.2.6: running sslstrip to get content from

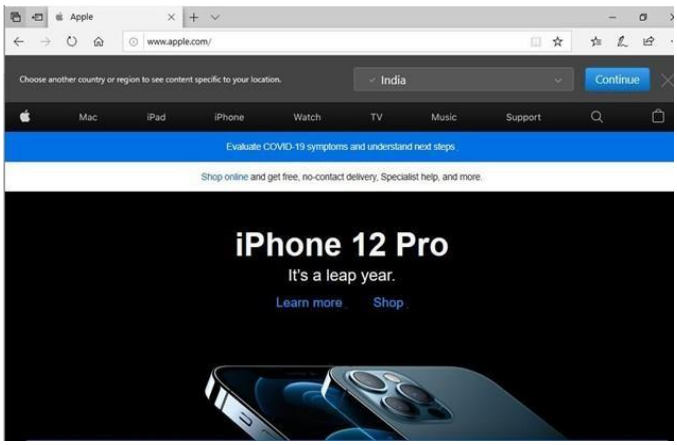


Fig.6.2.7 accessing apple.com after running sslstrip and we can observe that it does not load with https

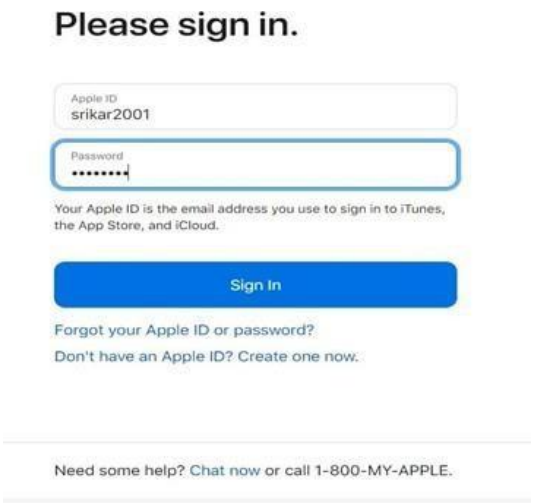


Fig.6.2.8 Entering the login details

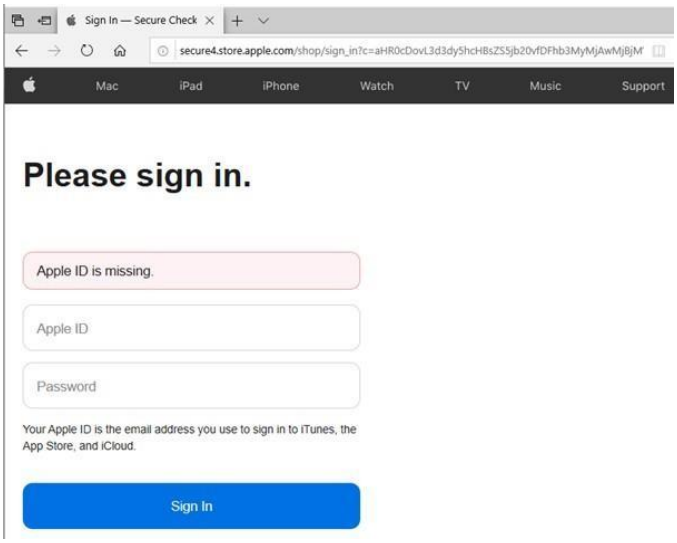


Fig.6.2.9 When we enter the details and hit sign in

Fig.6.2.10 When we observe it, the login detailshave beencaptured.

6.3 Detection

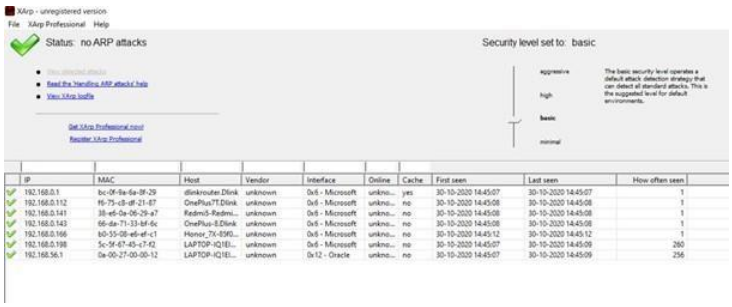


Fig.6.3.1 Using the XARP tool for detection

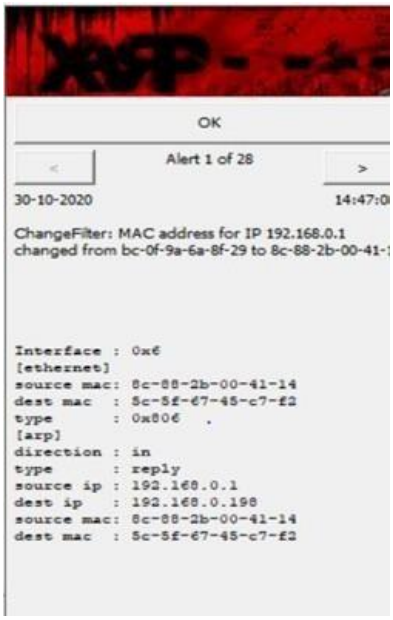


Fig.6.3.2 Alert the victim

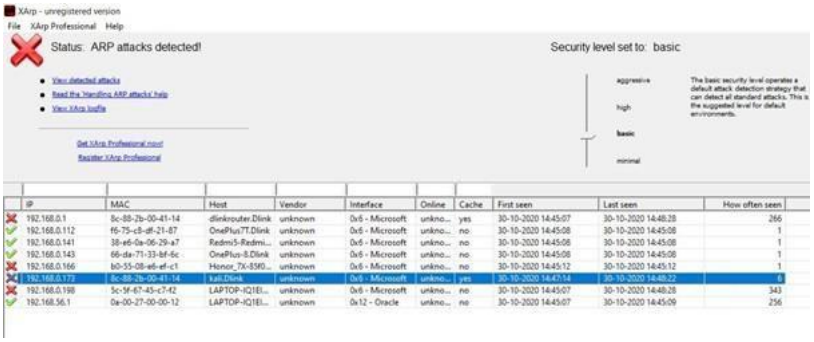


Fig.6.3.3 XARP tool after anattack.

VII. PREVENTION

7.1 Using VPN (Virtual Private Network):

The locations where a MITM attack might take place will be detected by using a VPN, but not all of them. In particular, it will shield the data traveling between your device and the VPN gateway, preventing your ISP (or most governments) from carrying out a targeted MITM attack on you.

7.2 Using static ARP tables:

Static IP addresses must be utilized inside the network, and this must be ensured by the entity that owns the network. Two entries will relate to the same MAC address only in case of ARP spoofing. We are all aware that the gateway or router relates to the primary entry. As a result, we must block communication with the IP address, just as in the second entry. The entry made by the packet that was spoofed is represented by the second entry. Therefore, we will only prevent communication with IP addresses listed in the second entry.

7.3 Using switches:

Using switches is a good measure when we are connected through LAN (wired). So, now the attacker should also be connected through the same mode to perform the attack. When he tries to scan for the devices, he/she would be confused to get the proper device from the list due to the presence of switches.

7.4 Using the XARP tool:

XARP is an open-source tool that will warn us immediately if there are any changes in the ARP table. This tool is a verygood preventive and detection mechanism.

7.5 Frequently checking the ARP table:

If we frequently check our ARP table, we might be able to prevent the attack. But everyone can't understand the mechanism of the arp -a command.

7.6 Proposing a new protocol to prevent thespoofing:

We create a replacement protocol in which every host is required to flood the network with search packets as soon as it discovers an item in its ARP table. Details about the source and destination MAC and IP addresses will be included in the probe packet.

VIII. Conclusion

In this survey, we learned how a small vulnerability can be exploited and can cause a great threat. The ARP protocol itself has vulnerabilities that cannot be avoided unless we use another protocol to communicate between layer 2 and layer 3. In every environment with security concerns, there will always be some trapdoor that can be used to exploit the vulnerability. If we want to perform any attack, we need to thoroughly study the underlying application or protocol or the environment and its methodology. To perform an attack or to provide security in both cases we need to think of both perspectives to achieve our goal. An attacker can achieve his goal in many ways and one such way is by using existing vulnerabilities and exploiting them. Another way is by simply confusing the user with domain names and achieving their goal. In our study, we experimented with both scenarios and achieved our goal.

References

[1]. Singh, Jaideep & Dhariwal, Sandeep & Kumar, Rajeev. (2017). A Detailed Survey of ARP Poisoning Detection and Mitigation Techniques. International Journal of Control Theory and Applications. 9

[2]. Ramachandran, V. and S. Nandi. "Detecting ARP Spoofing: An Active Technique." ICISS (2005)

[3]. Yang Liu, Kaikun Dong, Lan Dong, and Bin Li. 2008. Research of the ARP spoofing principle and a defensive algorithm. WTOC 7, 5 (May 2008), 413–417.

[4]. Kapil M. Jain, Manoj V. Jain, Jay L. Borade."A Survey on Man in the Middle Attack", IJERT Volume 02, Issue 10 (March 2016).

[5]. Mohapatra, Hitesh & Rath, Subhashree & panda, subarna & Kumar, Ranjan. (2020). Handling of Man-In-The-Middle Attack in

WSN Through Intrusion Detection System. 8. 1503- 1510.
[6]. Pateriya, Pushpendra & S Kumar, Srijith. (2012). Analysis on Man in the Middle Attack on SSL. International Journal of Computer Applications in Technology
[7]. Gangan, S. (2015). A Review of Man-in-the-Middle Attacks. ArXiv, abs/1504.02115.
[8]. Gunjan Agrawal, 2019, Detection and Prevention of ARP-Spoofing Attacks, IJERT Volume 08, Issue 09 (October 2019).

Srikar Vishnu Datta A. "A survey on Man in the Middle Attack - ARP Spoofing." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 24(6), 2022, pp. 09-20.