MitM Attack: Principles and Tricks

Chen-Yu Chang, Chenliang Wang, Hin Lui Shum, Yuxuan Luo, Yuhan Hu

Abstract – The growth of the internet started last century and thrived in the previous two decades. As we are stepping further to a more connected world with the help of the internet, our privacy and property are more likely to be at the risk of cyber-attacks. There are many types of security threats that attackers can use to exploit insecure applications. Threat actors can run some of these attacks using automated software, while others require a more active role from attackers. Some defense mechanisms focus on countering vulnerabilities from the application design, like XSS, SQL injection, CSRF, and others find their angle on the deeper level, like shell coding. Unlike these attacking methods, Man-in-the-middle (MITM) attacks can be sent between any combination of people, user computers, and servers. There are various ways to attack, so it is tough to identify and defend against them. Therefore, we must understand its principles and counter them and protect our systems. In this research project, we studied the logic behind man-in-the-middle attacks in depth. Moreover, we implement the several main types of man-in-the-middle attacks and techniques into the actual world website, including DOS spoofing, ARP spoofing, DNS spoofing, and SSL stripping. We will discuss how to detect and avoid such attacks through the implementation.

1 Introduction

Man-in-the-middle attacks (MITM) are a common type of cybersecurity attack that allows attackers to eavesdrop on the communication between two targets. The attack takes place in between two legitimately communicating hosts, allowing the attacker to "listen" to a conversation they should typically not be able to listen to, hence the name "man-in-the-middle." To better understand how a man-in-the-middle attack works, let's go through the following example from both sides.

1.1 Offline Man-in-the-Middle Attacks

Offline man-in-the-middle attacks are relatively basic, but they are used worldwide. For example, a man-in-the-middle successfully intercepts a message you send, reads and repackages it, and then sends the new content back to you or the original recipient. When that person replies unknowingly, the man-in-the-middle can continue to intercept and read the messages that were initially sent between the two communicating parties. Since the two parties are not communicating face-to-face, they are unaware of the information being intercepted and stolen.

If the attacker can control how the communication is made, he can even tamper with the message or implement other means of deception. For example, in the above example, the attacker could make illegal requests based on the intercepted message content. Of course, to do so unnoticed, the attacker will often delete any message records related to this attack immediately after completing the attack so that both sides of the communication cannot detect any anomalies.

1.2 Online Man-in-the-Middle Attack

Unlike offline attacks, online attacks often occur in real-time. For example, someone uses a laptop to connect to the free public Wi-Fi of a particular cafe and tries to access a certain bank's website. Subsequently, you may encounter the error message stating that your connection is not private. The message means a certificate error informing that the bank's website doesn't have the appropriate encryption certification. At that time, a man-in-the-middle attack is underway. When confronted with such an error, many people click on it and continue to visit the site. Subsequently, they log in to their bank accounts, send money, pay bills, etc., and everything seems to go on as usual. In fact, the attackers have already set up a fake server beforehand. They will modify the actual page of the target bank slightly or even forge it separately. All the login details you enter will be sent to the backend of the man-in-the-middle server, not to the actual bank server. This explains the error message about the encryption certificate, which actually stems from the fact that the middleman server does not have the same security certificate as the real bank.

2 Types of Man-in-the-middle Attacks

2.1 Mitm First Step - DoS

The basic model of Man-in-the-Middle attack is to act as a gateway to the victim machine and the victim to the gateway. This way, the victim machine sends all its packets to the attacker. If the attacker does not redirect those packets and drop them all, the victim machine cannot connect to the internet. To do so on Linux, perform the following command:

```
$ echo 0 >
/proc/sys/net/ipv4/ip_forward
```

This disables the forwarding operation automatically done by Linux and drops all the packets whose destination is not the local machine.

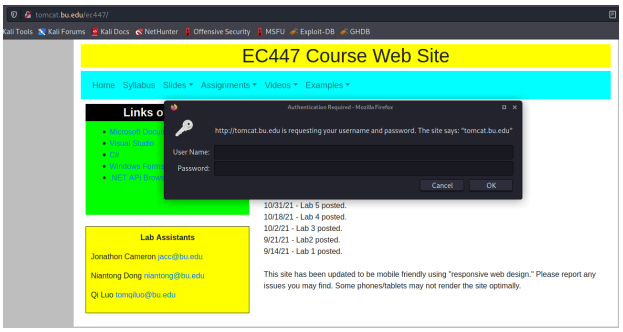
2.2 ARP-Spoofing

2.2.1 Principles

ARP spoofing is the most common way to redirect and sniff the internet traffic and can be used as a foundation for other complex and higher-level Man-in-the-middle attacks. The Address Resolution Protocol (ARP) is a communication protocol that maps internet layer addresses (IPv4/IPv6) to link-layer addresses (MAC). Since ARP is a stateless protocol, network hosts do not have a mechanism to "remember" any mappings. Therefore, hosts will automatically cache any ARP packets they receive, which is easy for the attacker to change the host's IP address to the attacker's MAC address.

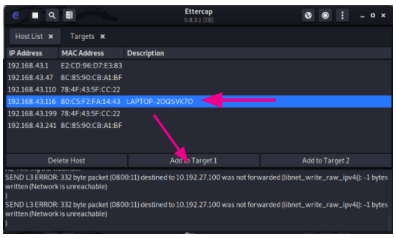
2.2.2 Real-World Practice: Gain Sensitive Credentials

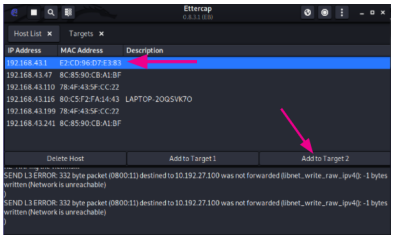
We will practice ARP spoofing under a real-world scenario and sniff the network traffic of a Windows device with a Kali Linux laptop, ultimately we will gain access to the EC447's course website as a student who forgot the credentials.



In this scenario, the assets are the class slides, and the vulnerabilities are the primitive ARP protocol and HTTP connection. To perform a systematic and convenient ARP spoofing, we can use Ettercap. A cross-platform tool designed explicitly for Man in the Middle attack. To use it, we first click on search for targets, which will scan all the hosts based on the size of the current network. For private networks like home WiFi and personal hotspots, hosts are typically 256. For large enterprises like Boston University, the hosts can be more than 10,000, and it would take a long time to scan for all the hosts. After the scan, we can find the target list by clicking on the button next to the magnifying glass. The IP addresses and the MAC addresses will show as a list.

In the target list, we can choose the victims we want to eavesdrop on. First, we need to find our victims from the IP addresses. In this case, our victim is 192.168.43.116. Next, we add it to our target 1, then add the network gateway to target 2. We can see our current target under the "Targets" tab. Since we have already located the victim's IP address, there is no need to add more targets. However, we might need to add all targets except for the gateway to Target 1 section if we are unsure about it. After choosing our target, we can launch the ARP poison from the drop-down menu on the top right corner.



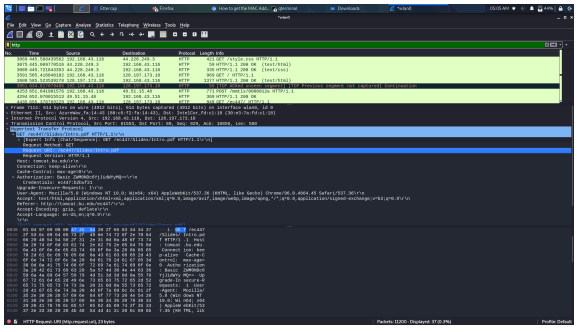


On the victim's side, we can see a change in the ARP table. The physical address of 43.1 is e2-cd. After the ARP poison, we can see it has changed to an address identical to another machine. This means the ARP poison is successful, and the network packets of the victim will be delivered to the attacker instead of the gateway. Here we have successfully put ourselves into the middle.

```
Interface: 192.168.43.116 --- 0x14
  Internet Address      Physical Address      Type
  192.168.43.1          e2-cd-96-d7-e3-83     dynamic
  192.168.43.47         8c-85-90-cb-a1-bf     dynamic
  192.168.43.110        78-4f-43-5f-cc-22     dynamic
  192.168.43.255        ff-ff-ff-ff-ff-ff     static
```

```
Interface: 192.168.43.116 --- 0x14
  Internet Address      Physical Address      Type
  192.168.43.1          30-e3-7a-fd-c1-18     dynamic
  192.168.43.47         8c-85-90-cb-a1-bf     dynamic
  192.168.43.110        78-4f-43-5f-cc-22     dynamic
  192.168.43.141        30-e3-7a-fd-c1-18     dynamic
```

Now we can use Tcpdump or Wireshark to listen to the traffic. To locate the packets sent to the target website, we need to specify the IP destination and protocol. In this case, the protocol is HTTP, and we can find the IP address by pinging the URL. Here we can see the packet containing the credentials.

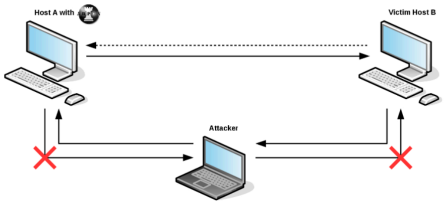


2.2.3 ARP Spoofing Mitigation

There are several ways to mitigate ARP spoofing attacks. First, Static ARP Tables is the nuclear option that can stop almost every ARP-related attack. It makes sure the MAC addresses are statically allocated to IP addresses, and every entry needs to be done manually. This can effectively prevent ARP spoofing, making the network system very hard to manage.

Another mitigation focuses on the inspection of the dynamics of the network. If there is a spike in the number of packets, then an abnormality should be detected, and the sender will be banned from the network.

For personal or home use, ArpOn is a powerful and lightweight protection software specifically for ARP spoofing, which provides three options for Dynamic, Static, and Hybrid ARP implementations. If we want to run Static or Hybrid ArpOn, we need to manually type in the static IP-MAC address to the Config file. Running the ArpOn on Host A can prevent A from sending packets to the attacker, but if this is a full-duplex situation (i.e. the attacker has poisoned both hosts), the attacker can still see the packets sending back from Host B to Host A. Given that most home-use WiFi does not provide a configurable interface, we cannot prevent the attacker from poisoning the ARP cache of the network gateway. Therefore, it is not as powerful as enterprise-level mitigations but is a lightweight and simple way to improve your ability to prevent ARP spoofing.



2.3 DNS-Spoofing

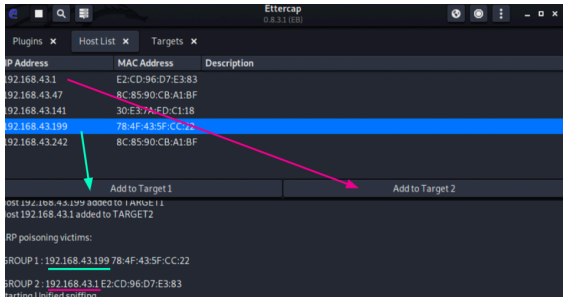
2.3.1 Principles

Domain Name Server (DNS) spoofing is an attack in which attackers alter DNS records and redirect victims' online traffic to a fraudulent website that resembles its intended destination. The victims are then prompted to log in with their credentials on a fraudulent website, potentially giving away their passwords and sensitive information to attackers. Also, the redirected websites may be used to install viruses or worms on the victims' computers and have long-term vulnerabilities.

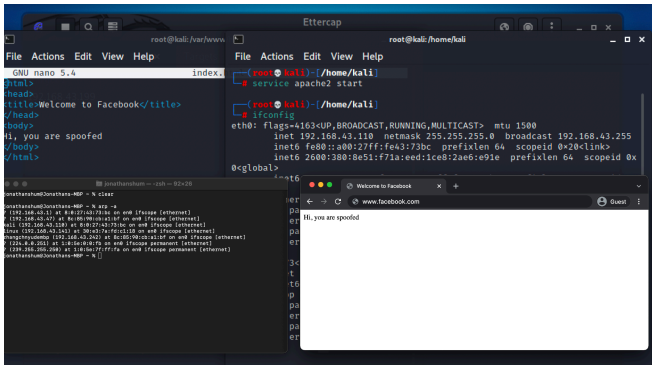
2.3.2 Real World Practice

DNS Spoofing is often performed on public networks, for example, free WiFis from cafes and restaurants. Since everyone can connect to the internet via Wi-Fi, it provides a dangerous environment for attackers to perform DNS spoofing and gain valuable information from users.

In our example, we use Kali Linux to simulate the environment to perform a DNS spoofing attack using ettercap and its plugins. Ettercap is a cross-platform tool designed explicitly for MitM attacks. Connect to the public network then run ettercap -g command to have a visual gui of ettercap running. Then it can scan for all hosts connected to the wireless network, if the attacker does not know the specific ip of the victims, he may still perform a mass attack targeting all hosts in the network. The attacker places the victims' ip into target 1 and router's ip into target 2.



In ettercap, DNS spoofing is an attack that relies on ARP spoofing to run, to activate DNS spoofing, click on the menu on the side and check the DNS spoofing plugin, a star will show on the side upon successful completion. Since then, the MAC address of the router will be the same as the attacker's MAC address. In /usr/share/ettercap/etter.dns, add a couple lines of the domain name server that wished to be redirected and attackers' ip. Creating an index.html with a similar design of the original page, for example facebook.com.Then run apache command and start an apache HTTP server using the website created. When the user visits facebook.com, the page will be redirected to the attacker's page and ARP spoofing will read the entered credentials by the user. Here is a successful result of the DNS spoofing redirecting facebook.com to a simple html created on the local server.



2.3.3 DNS Spoofing Mitigation

There are many ways to mitigate DNS spoofing attacks. Firstly, most websites nowadays use HTTPS connection instead of HTTP, most browsers provide warnings to the users when they are connecting to an insecure website. Also the important websites such as social media platforms, banks, and major shopping companies have implemented a feature named HSTS preload list. HSTS is a response header that tells a browser to

always use SSL/TLS to communicate with your site. It doesn't matter if the user, or a link they are clicking, specifies HTTP, HSTS will remove the ability for a compatible browser to use HTTP and will enforce the use of HTTPS. Although, this still leaves a user vulnerable on their very first connection to a site though. HSTS preloading lists fix it as it is a list compiled by Google and is utilised by Chrome, Firefox and Safari. The browser is already aware that the host requires the use of SSL/TLS before any connection or communication even takes place. This removes the opportunity an attacker has to intercept and tamper with redirects that take place over HTTP.

Secondly, prevent using public networks and entering credentials or valuable information when using these networks. In any case of using such networks, look for the secure connection symbol when connected to the websites. Also, users can run arp -a command in the terminal to see if there are 2 ip addresses with the same MAC address. If so, there might be some danger in the current network and the user should avoid using it.

2.4 SSL Stripping

2.4.1 Does HTTPS Stop Man-in-the-Middle Attacks

HTTPS (Hypertext Transfer Protocol over Secure Socket Layer) is a secure HTTP channel with security as the goal, simply a secure version of HTTP. Its implementation is to add an SSL layer under HTTP. The security basis of HTTPS is SSL, so the encryption details require SSL.

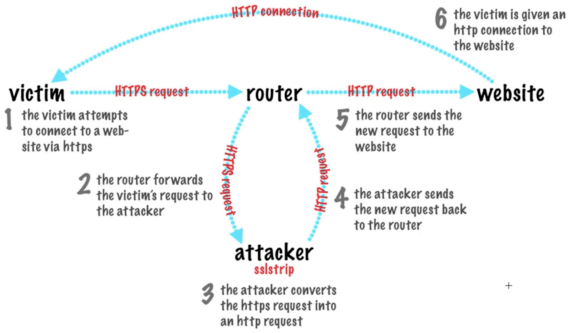
SSL protocol is located between TCP/IP protocol and various application protocols. It is an international standard encryption and authentication communication protocol that provides a reliable end-to-end security service for TCP, providing confidentiality and integrity between two communicating individuals.

In general, HTTPS protects against man-in-the-middle attacks. However, there are ways attackers can defeat HTTPS, removing the additional security afforded to your connection via encryption, such as SSL stripping.

2.4.2 Principles

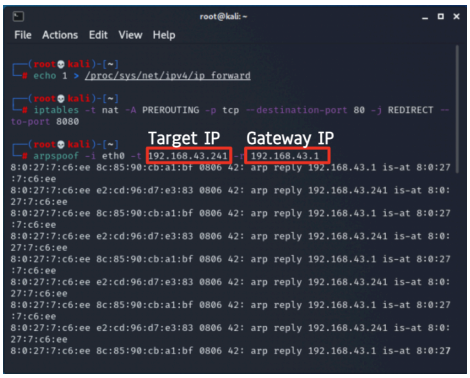
SSL (Secure Sockets Layer) is a method of encrypting traffic through the Internet and authenticating a server's identity. Small data files are digitally bound with a cryptographic key to an organization's details. When installed on a web server, it activates the padlock and the HTTPS protocol and allows secure connections from a web server to a browser. This technology is mostly used in credit card transactions, data transfer, and logins securing browsing of social media sites. SSL certificates move your connection from an HTTP connection to an HTTPS connection. HTTP is an insecure connection between a web browser and a website. The reason HTTP is so insecure is due to its lack of encryption of data, which is what HTTPS does. HTTPS, or Hypertext Transfer Protocol Secure, implements data-in-transit encryption to ensure that even if a Man in the Middle (MITM) attack occurred, that data would not be readable to the threat actor.

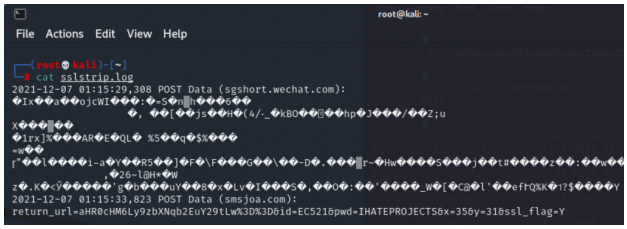
However, there is still a method of attacking through HTTP and HTTPS, SSL stripping. SSL stripping requires an arp spoofing to insert the attacker's computer in the connection between the victim and the website, as discussed previously. Then, the attacker can start doing SSL stripping. When the victims want to visit a website, they attempt to connect to a website through HTTPS to the router. It sends the HTTPS request to the attacker as we already did arp spoofing. When the attacker constructs SSL stripping happens: the attacker alters the request from HTTPS to HTTP back to the router so that the website returns with an HTTP connection to the victim. This way, the victims are in an insecure environment.



2.4.3 Real World Practice - Retrieve Username and Password

We first did an ARP spoofing specified in the previous sections, using the target's IP address and the Gateway IP address. Then, we installed the SSL stripping code based on python. We opened another terminal for running the SSL stripping. We need to look for a website that can be supported by both HTTP and HTTPS so that the program can transfer the format during stripping. We found a Korean website that includes a login page that operates the submit form action. In the first place, we enter the website, and the URL is in HTTPS; however, when we start the SSL stripping, the page turns into an HTTP website. When we type in the username and password, all those actions are recorded in a log file to get all the POST data. In the file, the username and password are clearly recorded so that the attacker can log in with that information successfully.





However, when we tried to change the URL from HTTP to HTTPS, the website successfully changed back to a secure website. Therefore, users can always be careful about which kind of website they are logging on to so that there will be no data leaks.

3 How to detect and protect against a Man-in-the-Middle

3.1 Strong Router Login Credential

Since MitM attacks have strict network environment requirements, setting up a high-security LAN can block MitM attackers from the very first step.

3.2 Virtual Private Network

VPN creates tunneling between source machine and destination machine, where the tunneling transmission contains almost no useful information. Therefore, using VPN does leverage the risk of being attacked by MitM.

3.3 Force HTTPS

As showcased in the previous sections, the major vulnerabilities MitM exploits are the low-security level of HTTP. Thus, avoiding HTTP and forcing HTTPS significantly decreases the risk of being attacked.

3.4 DAI (Dynamic ARP Inspection)

An essential reason why ARP spoofing can succeed is that the source of ARP reply is not authenticated. Dynamic ARP inspection verifies the source of ARP messages according to trusted databases built at runtime by DHCP snooping; if an ARP packet is not valid, DAI drops it.

3.5 Why does checking ARP table duplication not work

A naive approach to detect ARP spoofing is to check duplication in the ARP table: if there exist two identical MAC addresses in the ARP table, that means the machine is under a MitM attack. However, this can be defeated by spoofing MAC addresses to pretend to be different. For instance, on Kali Linux, the command `macspoof` can achieve this goal easily.

4 Summary

In this project, we generally implement and execute different types of man-in-the-middle attacks. Firstly we exploited how DOS and ARP spoofing worked and tried to apply it to real-world websites. Based on the ARP spoofing, we studied DNS Spoofing. Like how ARP resolves IP addresses to MAC addresses on a LAN, DNS resolves domain names to IP addresses. Additionally, we learned about SSL stripping, which is an advanced form of manipulating internet protocol to strip and remove the SSL configuration present on the websites. But detecting a man-in-the-middle attack can still be difficult without proper steps. If you don't actively search to determine if your communications are being intercepted, a man-in-the-middle attack could potentially go unnoticed until it's too late. By studying the logic behind man-in-the-middle attacks in-depth and implementing different types of MITM attacks, we find out specific ways to detect and protect against various types of attacks. If you know what to expect and what to look for, you have a much better chance of avoiding a MITM attack. In turn, your data will remain secure and firmly in your hands.

References

1. Man in the Middle Attack: Tutorial & Examples | Veracode. https://www.veracode.com/security/man-middle-attack
2. Man in the Middle (MITM) Attacks | Types, Techniques, and ....

https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/
3. ArpON::ARP handler inspection. https://arpon.sourceforge.io/
4. ARP Handler Inspection https://arpon.sourceforge.io/documentation.html#2
5. Ettercap https://www.ettercap-project.org/
6. Ubuntu ARP Manual http://manpages.ubuntu.com/manpages/xenial/man8/arpon.8.html
7. SSL Stripping Source Code https://github.com/juanelas/sslstrip
8. How To Enable The Network In Kali Linux Virtual Box? https://www.hacking-tutorial.com/tips-and-trick/how-to-enable-the-network-in-kali-linux-virtual-box/#sthash.cvVbVBOZ.dpbs
9. Detailed Guide to Preventing SSL Stripping https://www.encryptionconsulting.com/detailed-guide-to-preventing-ssl-stripping/
10. Insecure Website Used in Demo https://www.smsjoa.com
11. What is SSL? https://support.rebel.com/hc/en-us/articles/227833687-What-is-SSL-and-what-is-it-used-for-
12. How to Use SSL Strip on Kali Linux? https://www.youtube.com/watch?v=OtO92bL6pYE