

Analysis of a Man-in-the-Middle Experiment with Wireshark

Ming-Hsing Chiu, Kuo-Pao Yang, Randall Meyer, and Tristan Kidder

Department of Computer Science and Industrial Technology
Southeastern Louisiana University, Hammond, Louisiana

Abstract - With the rapid growth of the Internet user population and the magnitude of the applications depending on the Internet these days, network security measures are becoming extremely important. For the Internet users, one of the best defenses against network attacks is to understand the patterns of the attacks and raise the awareness of abnormality as much as possible. In this paper, an experiment was employed to demonstrate a form of active attacks, called Man-in-the-middle (MITM) attack, in which the entire communication between the victims is controlled by the attacker. A detailed description of setting up the system for MITM is included. The victim initiated a few activities that cause the attacks, which were captured by Wireshark at the attacker site and analyzed. The result clearly reveals the pattern of the MITM attack. Some remarks on the preventive measures were made based on the result.

Keywords: Man-in-the-middle attack, Wireshark, ARP

1 Introduction

The *man-in-the-middle attack* (often abbreviated *MITM*) is a well-known form of active attack in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker[1,2]. It allows the attacker to eavesdrop as well as to change, delete, reroute, add, forge, or divert data [3]. For the Internet users, one of the best defenses to MITM attacks is to understand the patterns of the attacks and raise the awareness of abnormality during the attacks. As an effort to demonstrate the characteristics of the attack, an experiment was carried out and the network traffic was captured and analyzed by using the packet sniffer, Wireshark. Previous works that used Wireshark in the similar manner can be found in [4,5].

The paper is organized as follows. Section 2 provides the background information on the capturing and display processes of Wireshark. Section 3 gives a detailed description of setting up an experiment for demonstrating the MITM attack under Linux operating system. A few MITM activities were captured in the experiment and analyzed to search for the patterns of the attack in section 4. Preventive measures and warning signs of the MITM attacks were discussed in section 5. Finally, section 6 provides conclusion of this work.

2 Wireshark

Wireshark (formerly known as Ethereal)[6] is a free and open-source packet analyzer, based on libpcap. It is widely used in network troubleshooting, analysis, protocol development by network professionals as well as educators. It accepts wide range of protocols, such as TCP, IP, ARP, HTTP, and etc. Note that we use the terms packet and frame interchangeably in this paper.

In display mode, Wireshark presents a colorful window with three different areas when you open a captured file with a set of packets. On the top most area of the window is Area 1(listing area), which is the listing of all the captured frames. Each line is a summary of a frame displaying the information depicted on the top heading. When you click on a packet in Area 1, the detailed packet structure is shown on Area 2(detailed area) directly below Area 1. Clicking on a portion of the packet in Area 2 changes the display in Area 3(raw data area), which is the raw data of the frame shown in Area 2.

3 Setting up the system

In general, MITM involves three computers, two victims and one attacker. It is performed by the attacker sending a signal to the first victim telling the victim he is the second victim, and sending a signal to the second victim saying he is the first victim. This creates a Man-in-the-middle effect in which the first victim sends all its packets to the attacker which are then relayed to second victim and vice versa. In the experiment, as depicted in Figure 1, the second victim is a Web server. When the Spoofed connection is made, the victim browses the internet as normal.

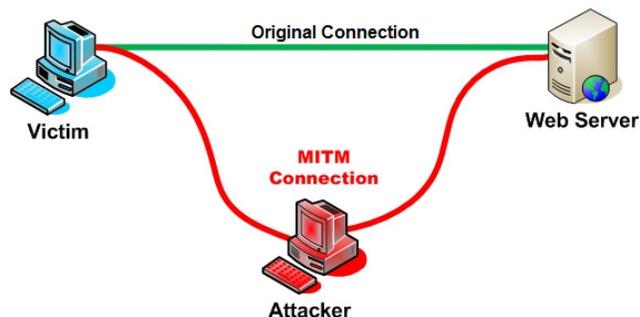


Figure 1: Man-In-The-Middle Attack

The experiment was carried out on a campus lab, under the supervision of the network administrator. Since both first victim and attacker reside in the same LAN (subnet) with a single gateway, we only need to spoof the victim and the gateway. This is because all traffic between the first victim and the Web server must pass through the gateway due to the last hop situation. During the experiment, the attacker was running on a Linux Backtrack3 operating system, since most of the tools needed are included as native applications in the operating system, except *sslstrip* which is to be discussed later.

There are a few steps involved in the setting up of the attack. A script file that carries out the set up, step by step, automatically was implemented and invoked at the onset of the experiment.

3.1 Enter interface and the Gateway IP address

Both parameters can be obtained by running *ipconfig* tool and the parameters must be entered to the operating system. In our case, the interface is 'eth0' and the Gateway IP address is 147.174.120.1.

3.2 Scan the network to find target IP

Use *nmap* tool to map network for accessible services and use *ipscan* to scan the subnet to find the target IP (victim), then enter the IP. Note that the attacker and the victim reside in the same subnet. The computer with IP = 147.174.120.208 was chosen as the victim.

3.3 Enable IP forwarding

In order for ARP spoofing to work, IP tables need to be prerouted and IP forwarding needs to be enabled. This is done by using *iptables* tool.

3.4 Complete ARP spoof

This step is also called ARP poisoning, in which the attacker take advantage of the ARP protocol by impersonate as victim to gateway, and as gateway to victim. The effect is the modification of IP forwarding table on both gateway's and victim's sites. After this step, attacker gets all the message exchange between Web server and the victim. Table 1 lists the correct MAC addresses and the modified MAC addresses on the IP tables of the gateway and the victim due to the effect of *arp spoof*. This table is useful for references when doing packet analysis in Wireshark. The MAC addresses shown on the table are the generic names for better readability.

Table 1: IP and MAC addresses table after the arp spoof

	IP address	MAC address	Modified address on Gateway's table	MAC on IP	Modified address on Victim's IP table
Gateway	147.174.120.1	PrimaryA_6b:40:99			DellComp_4e:4f:69
Victim	147.174.120.208	AmbitMic_cc:1b:6c	DellComp_4e:4f:69		
Attacker	147.174.120.235	DellComp_4e:4f:69			

4 Activities captured and analyzed

After ARP spoofing was run successfully, the victim initiated a few activities that demonstrate MITM attacks. These activities were captured by Wireshark and a Pcap file was generated at the attacker's site. Below are the analyses of the activities captured during the experiment. To improve the readability and save the space, in figure 2 to figure 6 (showing Wireshark display window), only a single frames will be included on Area 1(listing area). Area 3 (raw data area) will not be displayed. Note that doing a MITM attack produces several ARP frames as well as retransmission frames. Figure 2 shows an ARP frame that appeared several times during the experiment as the result of running *arp spoof*. This frame was sent from the attacker to the victim attempting to impersonate as gateway by inserting its own MAC address. This can be seen in the area showing the detail of Address Resolution Protocol, in which the Sender IP address is the gateway's IP address but the Sender MAC address is that of attacker's.

```

No.    Time           Source                Destination            Protocol    Info
2 0.000384      DellComp_4e:4f:69    AmbitMic_cc:1b:6c     ARP        147.174.120.1 is at 00:08:74:4e:4f:69
Ethernet II, Src: DellComp_4e:4f:69 (00:08:74:4e:4f:69), Dst: AmbitMic_cc:1b:6c (00:d0:59:cc:1b:6c)
Address Resolution Protocol (reply)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (0x0002)
  [Is gratuitous: False]
  Sender MAC address: DellComp_4e:4f:69 (00:08:74:4e:4f:69)
  Sender IP address: 147.174.120.1 (147.174.120.1)
  Target MAC address: AmbitMic_cc:1b:6c (00:d0:59:cc:1b:6c)
  Target IP address: 147.174.120.208 (147.174.120.208)
  
```

Figure 2: ARP Frame Showing the Attempt of the Attacker to Impersonate as Gateway

4.1 Victim browses www.google.com:

Since the attacker is effectively acting as the relaying station of the messages exchanged between the gateway and the victim, each message coming from gateway/victim will generate a retransmitted message to victim/gateway. Figure 3(frame # 15) and 4(frame # 16) are the original and the retransmitted HTTP frames that request for Web page from www.google.com respectively. Note that the only difference between these two frames is the source and destination MAC addresses at the Network Access Layer, Ethernet II, displayed in detailed area. The pattern of the MAC addresses of the source and destination pair in each frame clearly demonstrates the relaying behavior of MITM attack. The reply from www.google.com shows similar retransmission pattern except the MAC addresses of the source and destination pair are reversed.

No.	Time	Source	Destination	Protocol	Info
15	1.868159	147.174.120.208	74.125.229.18	HTTP	GET / HTTP/1.1
Frame 15: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface eth0 Ethernet II, Src: AmbitMI_cc:1b:6c (00:d0:59:cc:1b:6c), Dst: DellComp_4e:4f:69 (00:08:74:4e:4f:69) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 74.125.229.18 (74.125.229.18) Transmission Control Protocol, Src Port: neoiface (1285), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452 Hypertext Transfer Protocol GET / HTTP/1.1/\r\n Accept: */*\r\n Accept-Language: en-us\r\n Accept-Encoding: gzip, deflate\r\n User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; GTB6.5)\r\n Host: www.google.com\r\n Connection: Keep-Alive\r\n [truncated] Cookie: PREF=ID=b0305f3bbdf5afb2:U=be4f9f74cb21bf4b:TB=5:TM=1268761617:LM=1280796687:S=0Cq:\r\n\r\n					

Figure 3: Google page request originated from the victim

No.	Time	Source	Destination	Protocol	Info
16	1.879019	147.174.120.208	74.125.229.18	HTTP	[TCP Retransmission] GET / HTTP/1.1
Frame 16: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface eth0 Ethernet II, Src: DellComp_4e:4f:69 (00:08:74:4e:4f:69), Dst: PrimaryA_6b:40:99 (00:20:9c:6b:40:99) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 74.125.229.18 (74.125.229.18) Transmission Control Protocol, Src Port: neoiface (1285), Dst Port: http (80), Seq: 1, Ack: 1, Len: 452 Hypertext Transfer Protocol GET / HTTP/1.1/\r\n Accept: */*\r\n Accept-Language: en-us\r\n Accept-Encoding: gzip, deflate\r\n User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; GTB6.5)\r\n Host: www.google.com\r\n Connection: Keep-Alive\r\n [truncated] Cookie: PREF=ID=b0305f3bbdf5afb2:U=be4f9f74cb21bf4b:TB=5:TM=1268761617:LM=1280796687:S=0Cq:R1P:\r\n\r\n					

Figure 4. Google page request relayed by the attacker

4.2 Victim logs in to an insecure login site: projecteuler.net

This example shows that the password in an insecure login can be easily snatched by way of MITM attack. Again, we show the two frames that were originated from the victim and the relay from the attacker to the Web site.

As depicted in Figure 5 (frame 113) and Figure 6 (frame 114), the password can be seen on the last line of the detail area, in which the username is iTruth and the password is CMPS309. The Attacker may alter the login information in the retransmission, as a result, the victim cannot login.

No.	Time	Source	Destination	Protocol	Info
113	5.000459	147.174.120.208	95.154.244.24	HTTP	POST /index.php HTTP/1.1 application/x-www-form-urlencoded
Frame 113: 556 bytes on wire (4448 bits), 556 bytes captured (4448 bits) on interface eth0 Ethernet II, Src: AmbitMI_cc:1b:6c (00:d0:59:cc:1b:6c), Dst: DellComp_4e:4f:69 (00:08:74:4e:4f:69) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 95.154.244.24 (95.154.244.24) Transmission Control Protocol, Src Port: hp-sci (1299), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502 Hypertext Transfer Protocol Line-based text data: application/x-www-form-urlencoded username=iTruth&password=CMPS309&login=Login					

Figure 5: Login to Projecteuler originated from the Victim

No.	Time	Source	Destination	Protocol	Info
114	5.000857	147.174.120.208	95.154.244.24	HTTP	[TCP Retransmission] POST /index.php HTTP/1.1
Frame 114: 556 bytes on wire (4448 bits), 556 bytes captured (4448 bits) on interface eth0 Ethernet II, Src: DellComp_4e:4f:69 (00:08:74:4e:4f:69), Dst: PrimaryA_6b:40:99 (00:20:9c:6b:40:99) Internet Protocol, Src: 147.174.120.208 (147.174.120.208), Dst: 95.154.244.24 (95.154.244.24) Transmission Control Protocol, Src Port: hp-sci (1299), Dst Port: http (80), Seq: 1, Ack: 1, Len: 502 Hypertext Transfer Protocol Line-based text data: application/x-www-form-urlencoded username=iTruth&password=CMPS309&login=Login					

Figure 6: Login to Projecteuler relayed by the attacker

4.3 Other example

In a separate experiment, the attacker used Wireshark to capture eavesdrop of Instant Messenger. Figure 7 depicts a composite picture, in which Messenger window and the corresponding Wireshark displays are linked by the red marker. Note that the filter “MSNMS” was applied to obtain the displays.

Figure 7: A snapshot of the eavesdrop of Instant Messenger

5 Preventative measures

To the average victim, the MITM attack is relatively hard to detect. This is particularly true when the victim is engaged in non-secure transactions as shown in the previous examples. In fact, there are some free tools available for detecting the anomaly when MITM is being performed. Thus help preventing the user from becoming a victim of the attack. A good example is the tool, called *DecaffeinatID*, which monitors user's gateway MAC address. If that changes, as in the case during a MITM attack, it notifies the user with a popup box as shown in Figure 8.



Figure 8: A warning Message of Potential MITM Attack

Some variants of MITM attack, such as those equipped with the *sslstrip* tool, are capable of compromising SSL/TSL type of security measures, making the interception of the secured data possible. To some extent, though, when dealing with secure login sites, the attacker is relying on the victim's ignorance to achieve success. For instance, when logging into sites like Chase.com or live.login.com, it verifies the certificate. When a MITM attack is being performed, the victim will receive a certificate warning. If the victim accepts the false certificate, then the attacker will intercept the login information; however, if the victim does not, they will not be able to login to that site, which is a wiser choice. In summary, the awareness of abnormality is important in the preventative measures.

6 Conclusions

For the Internet users, one of the best defenses against network attacks is to understand the patterns of the attacks and raise the awareness of abnormality as much as possible. We use an experiment to demonstrate a form of active attacks, Man-in-the-middle (MITM). Wireshark was used to capture and analyze the MITM activities in the experiment. From the result, we identified the characteristics of the MITM attack. We also make some remarks on the preventative measures and emphasize the importance of awareness of the abnormality. We found that Wireshark is an indispensable tool in carrying out the experiment which is suitable in disseminating the knowledge of the MITM attack in the classroom environment.

7 References

- [1] http://en.wikipedia.org/wiki/Man-in-the-middle_attack
- [2] M. E. Whitman, H. J. Mattord, *Principles of Information Security*, Thomson Course Technology, 2005.
- [3] D. Radcliff, "What Are They Thinking?" *Network World*, March 1, 2004
- [4] M. A. Qadeer, M. Zahid, A. Iqbal, M. R. Siddiqui, "Network Traffic Analysis and Intrusion Detection Using Packet Sniffer," *Proc. Of Seond. Int. Conf. on Comm. Software and Networks*, 2010, pp 313-317.
- [5] S. Wang, D. Xu, S. Yan, "Analysis and Application of Wireshark in TCP/IP Protocol Teaching," *Proc. Of Int. Conf. on E-Health Networking, Digital Ecosystem and Technologies*, 2010, pp 269-272.
- [6] <http://www.wireshark.org/>