# CYBER ATTACKS EXPLAINED: THE MAN IN THE MIDDLE

**Due to the encouraging feedback this series of articles has received, we decided to explore yet another type of cyber intrusion□the Man In The Middle (MITM) attack, which is widely favoured by attackers. Unfortunately, many systems administrators don't seem to understand its severity. In this article, we will explore ways to protect FOSS infrastructure.**

In a previous article, we learnt about various packet-spoofing attacks, and I also demonstrated how it can result in an MITM attack. However, since this type of attack is favoured by attackers, I thought it worthwhile to dedicate an article to it. MITM attacks are used mainly to steal information and data. They are possible by exploiting inherent vulnerabilities of the TCP/IP protocol at various layers. Technically speaking, these are a derivative of packet sniffing and spoofing techniques. If carried out properly, users can be completely oblivious to this attack, making it difficult to detect and stop. MITM attacks are sometimes called bucket brigade attacks, eavesdropping attacks, or connection hijacking attacks.

Please refer to Figure 1. At a macro level, the steps involved are to intercept traffic, break the authentication chain, and impersonate the endpoints seamlessly. The objective is to steal the session, and thus the information being transmitted.

As we know, TCP/IP works on a three-way handshake (between SYN, SYN-ACK and ACK) that establishes a connection, which then uses packet sequencing and data acknowledgements to send or receive data. Data is passed between the (OSI) physical and application layers. At Layer 2 and Layer 3, Ethernet and IP packet datagrams are formed, whereas at the presentation layer, the cryptographic SSL session is established, assuming that the application is planning to use it. It is important to note that each of these layers can potentially participate in the MITM attack.

The table in Figure 2 maps the OSI layers to various possible MITM attacks. To steal the session, the attackers need to first use a packet sniffer. After gathering packets, they choose which layer to attack□ARP protocol, IP or SSL. Next, the attackers use advanced tools to inject packets to steal the session without the knowledge of the victims whose session is being hijacked. Although MITM uses IP spoofing at its base, it goes a mile beyond that in order to gain control by choosing sessions from one or more layers to be hijacked. For example, the attackers can use ARP spoofing and also cryptographic session stealing to achieve complete control over an e-commerce transaction. Also, compared to just spoofing packets, MITM attacks modify the intercepted data and re-transmit it, making the victim believe the packet is from a legitimate source (which, in fact, is just another victim).

Since it can happen at various layers, the severity of an MITM attack can be very high. For example, a legitimate employee of a firm can use a MITM attack to download the HR database being viewed by a fellow HR employee, by stealing the session of the TCP port being used by the SQL server and the HR desktop application. Or credit card numbers could be stolen by hijacking a browser session connected to a shopping cart Web application. Advanced attackers can steal sessions established at lower OSI layers such as Layer 3, to dump all the packets and gain a deep insight into applications running on the particular computer, and the usage pattern and other personal details of the person using it. It is also possible to record an entire authentication session, and decipher the user ID and password information from it, which can be used by an attacker to impersonate the victim and create more damage. We again wish to emphasise that MITM is a stealth type of attack whereby the victims cannot detect the presence of hacker on the network, and hence this attack should be taken very seriously. The good news is that there are a few ways to stop or at least mitigate the impact of this attack, which you will soon learn about. Now let□s take a look at various MITM attacks that take place at different networking layers.
**ARP Poisoning**: ARP does MAC address-to-IP address translation. In a normal situation, when a TCP/IP protocol stack running on one computer wants to send a

packet to a destination IP address, it first looks into its local cache for a mapping of the IP with a MAC address. If the entry can□t be found, it starts an ARP broadcast, asking for a MAC address for the given IP address. The machine that owns that IP address responds to the ARP query with its own MAC address. The source then uses it to create the Ethernet frame, and starts transmission to that address, and also stores the IP-to-MAC entry in its local cache, to speed up future communication to the same IP. ARP is a stateless protocol, and the cache doesn’t have its own security mechanism, which yields a serious vulnerability. The MITM being based on packet spoofing and forging, the attacker sends spoofed ARP packets on the LAN to associate the attacker’s own machine□s MAC address with the IP address of the target host. This forged packet can be as simple as an ICMP packet typically used to ping a host. Traffic meant for the target now reaches the attacker’s machine, due to the cached IP-to-MAC entries.  The attacker can repeat this on another machine, to view communication between both the targets□thus making it a LAN-specific MITM attack. Once control is achieved, the attacker simply collects packets from the sender, and forwards them to the receiver, while recording the packet stream inbetween. Since there is no data lost, both victims have no clue; the attacker steals data in hiding. ARP poisoning is widely used in internal attacks, i.e., by attackers on the same LAN.

**DNS MITM:** This method is rarely used, but can create a serious security breach. An attacker first sniffs network traffic specifically for DNS queries on port 53. Gathered information is analysed to identify all DNS servers on the network. The attack uses ARP spoofing to fool the querying machine into sending DNS requests to the attacker□s computer, rather than the real DNS server. Every DNS query contains a unique identification number to be able to map it to the response received. For the MITM, the query response packet needs this number, but with a bogus query answer. In real life, a DNS MITM is tough to execute, and is also time-consuming; however, it leads to a great deal of network insecurity. Please refer to Figure 3, which explains this attack in more detail. What could be achieved is up to the attacker□s imagination. As an example, the attackers can divert all requests to a

Web server to their own IP address(es), create a phishing website, and steal credentials. If this attack is planted externally, the damage can actually be beyond imagination. In another form of DNS MITM attack, the attackers stress the DNS server by sending a large number of queries, and meanwhile intercept its traffic, to send bogus responses to DNS queries.

**DHCP MITM:** Similar to the DNS attack, here, the DHCP server□s queries and responses are intercepted. This helps the attacker gain a complete picture of the network, such as hostnames, MAC addresses, IP addresses, DNS servers, etc. This information is further used to plant advanced attacks to steal information. Also, like the DNS attack, attackers can use a denial-of-service attack on a real DHCP server to keep it busy, while they spoof responses to DHCP queries. A DHCP MITM just acts as a catalyst in the data-stealing process. However, the impact can be further increased either by expiring the IP lease on all hosts, thus causing a DHCP broadcast storm on the network, or by assigning the same IP to all hosts, to render the network practically non-functional.

MITM attacks from the application perspective

Cookie hijacking: As we know, a cookie is a small piece of text stored by a browser on the user□s machine and stores information related to the website that created it, usually to □remember□ users' online membership details or profile settings, etc. Stored cookies can be accessed by the browser as well as other applications. By writing cookie-stealing scripts or programs, part or all of the cookie content can be captured, interpreted and presented to the Web server, in order to impersonate the user, who now becomes a victim in this scenario. There are three ways to capture cookie details□directly access it on the computer, hijack the browser session, or sniff Web requests and responses transmitted over the wire. Some websites store login session and authentication information in the cookie□which, if hijacked, can let attackers use the server without being challenged for user ID and password authentication. This could lead to someone□s email account being hijacked, or even their online profile being impersonated, thus resulting in possible monetary losses.

**Man in the browser:** This is an Internet browser attack wherein a Trojan virus or a malicious script runs in a browser instance to gain control of the Web session. To do this, attackers could exploit inherent un-patched browser vulnerabilities, or vulnerabilities introduced due to the mis-configuration or non-hardening of the OS on which the browser is running. Since the attack takes place in the same security context as the user of the browser, security mechanisms such as SSL and multi-factor authentication prove to be useless. Multiple OS' running a variety of browsers exhibit different vulnerabilities, which could be easily exploited by a program written in JavaScript or similar scripting language.

To explain the real danger caused by this attack, let□s consider a user trying to purchase goods on the Internet when the browser is already compromised. The user sees what is on the screen and submits the request, but what the Web server sees is what is actually submitted to it, which can be quite different. If the user submitted his own address for the goods to be shipped to, the address being submitted can be a fictitious one, set up by the attacker to collect the goods□while the user who paid for them will not have a clue about it.

**SSL MITM**: As we know, the SSL protocol works on public key cryptography concepts, whereby a server is hosted with a digital certificate for online data encryption. This certificate is provided by some trusted authority such as VeriSign, to endorse the identity of the website URL. In a normal situation, when a client (a browser) tries to connect to the secure server, the server provides its certificate to the browser. The browser then checks it against its own list of valid and trusted certificates, and verifies the information contained in it. Once the certificate is accepted, the server and browser both negotiate on a common level of encryption to be used. They then create a key, and transfer data in the encryption form agreed to. If the certificate has expired or is not trusted, the browser flashes a warning to alert those who are using it. In order to compromise this situation, attackers sniff the traffic between browser and Web server, and insert their computer in the network by using standard spoofing techniques. From this point onwards, the attacker□s computer runs a proxy tool for the SSL communication. This is done in such a way that the Web server negotiates the SSL key with the attacker□s machine, while the communication with the browser is pure text. This results in exposing all the sensitive information to the attackers□ computer, which is duly sniffed, recorded and stored for malicious usage. In another case, if the attackers gain physical access to the victim computer temporarily, they install a bogus certificate on it, and then use standard IP spoofing to divert traffic, establish an SSL connection with the Web server using the trusted certificate, and also with the browser, by using the bogus certificate. This is done to ensure that the URL is seen as HTTPS, and the actual communication is indeed SSL□helping the attackers to hide, but data is being stolen in the process.

**Wireless MITM:** Similar to wired attacks, in the case of wireless networks, the data stealing can actually be a bit easier. It exploits design vulnerabilities in Wi-Fi networks, whereby a dummy access point is set up on the network, with the same SSID signature, and a stronger signal. The next step is to disrupt client traffic to the real legitimate wireless router, either by spoofing or by subjecting the router to a DoS attack using a packet storm. Once the router gives up, a Wi-Fi connection is established by the dummy router, which starts the packet sniffing and data theft. There are a few other ideas, such as logging into the real router by using a brute-force method, and creating a wireless bridging with the dummy router; or faking the MAC address of the real router to fool and lure Wi-Fi clients to the dummy router, etc. Due to the use of wireless signals, it is unfortunately possible for the attacker to steal data undetected, almost forever. Due to a lack of knowledge, it is very common to have wireless routers at home or work, which are not configured correctly, or not secured using appropriate keys or passwords.

Other MITM attacks: It is important to note that MITM attacks are possible in the critical networking components, as well as with new technologies. Routers and managed switches can be spoofed resulting in a higher security impact, and the firewalls can also be fooled to deliver legitimate traffic to a rogue machine. Technologies such as mobile computing and Bluetooth are vulnerable to MITM attacks, but those are out of the scope of this article.

## Protecting FOSS systems

While the MITM attack is a difficult one to tackle, there are a few preventive mechanisms that all network administrators should adopt in their infrastructure. Since the heart of an MITM attack is packet sniffing and spoofing, utmost care needs to be taken while designing the network to introduce network monitoring points. Using intrusion prevention systems proves to be a very good choice for large companies, where the network is complex, and hence more susceptible to MITM. Since this attack exploits the inherent vulnerabilities of the unprotected TCP/IP protocol, techniques such as IPSec security can be used to ensure traffic authentication at lower OSI layers.

For a data centre hosting FOSS systems, the very first thing to implement is MAC address security. This can be achieved by choosing a host-based model, or configuring the managed switches and DHCP systems appropriately. To protect DNS systems, one can use DNSSEC, which digitally signs each DNS query request and its response, thus endorsing its legitimacy. In case of very critical IT infrastructures, using manual 'hosts' entries instead of DNS servers can actually help fix DNS hijacking problems. For FOSS-based Web infrastructures, it is recommended that you use stronger ciphers implemented around TLS technology instead of SSL. Also, properly authenticated session management and client-side authentications such as client certificates should be used. Another idea is to use multi-factor authentication implemented on secure protocols. As we learnt earlier, an MITM attack planted on a browser can bypass all these security measures; hence, it is important to implement the best secure coding practices to deal with such situations in the Web applications.

Linux FOSS systems come with a built-in, but usually less known feature called □source address verification□. It is a kernel feature which, when turned on, starts dropping packets which appear to be arriving from the internal network, but in reality are not. Most of the latest kernels, such as Ubuntu and CentOS, do support it, but if your Linux distro does not, it is time to upgrade. Modifying the hosts.conf file to add □nospoof on□ is another level of defence to try. In terms of detection, for smaller

Linux networks, a nice utility called arpwatch is very useful. It keeps track of MAC and IP addresses, records all changes and can be scripted to alert administrators upon a possible attack. Scripting can also be done to go through network interface logs and look for anomalies in terms of source address forging.

**Summary**

A MITM attack is really difficult to tackle, and hence should be taken seriously by the IT management team. It can result in data theft, causing severe reputation and monetary losses to firms. The bottom line is that you must have a correctly defined security perimeter defence design, ensure server and network component hardening, implement a robust patch management system, and follow the best security practices. Since this attack may not be visible, being vigilant in terms of network problems and performance always helps detect it, before a data theft can occur.

Source: http://www.opensourceforu.com/2013/01/cyber-attacks-explained-the-man-in-the-middle-part-8/