Layer 2 Attacks and Mitigation Techniques for the Cisco Catalyst 6500 Series Switches Running Cisco IOS Software

MAC Address Overflow Attack and Mitigation Techniques

Authors:

Kevin Lauerman, CCSP, CISSP 80877
Jeff King, CCIE 11873, CCSP, CISSP 80875

Abstract

Security is at the forefront of most networks and many companies implement a comprehensive security policy encompassing many of the OSI layers, from application layer all the way down to IP security. However, one area that is often left untouched is hardening layer 2 and this can open the network to a variety of attacks and compromises.

This document will have a focus on understanding and preventing Layer 2 attacks on the Cisco® Catalyst® 6500 switching platform. Denial-of-Service (DoS) attacks are always a major concern. DoS attacks can come from internal and external sources. The focal point of this white paper will be to understand how a MAC Address Overflow Attack (DoS Attack) works and what techniques can be used on the Cisco Catalyst 6500 switch running Cisco IOS® Software to mitigate this type of attack. An attack tool called Ettercap was used to execute this attack.

A MacBook Pro and a Lenovo T61P was used for these test and acted as the attacker in some cases and the victim in others. Both computers ran VMware with different guest OSs acting as the attacker or victim.

Note that all the attacks performed in this document were done in a controlled lab environment. It is not recommended that you perform any of these attacks on your enterprise network.

Test Equipment

A Cisco Catalyst 6509E switch with a Supervisor 720-3B running Cisco IOS Software 12.2(33)SXI1 in an Advanced Enterprise Feature Set and a WS-X6748-GE-TX (10/100/1000) Ethernet line card was used. For the Attacker and Victim computers, an Apple MacBook Pro and a Lenovo T61P was used.

The MacBook Pro ran a native Mac OS X version 10.5.7 and also had VMware Fusion (2.0.5) with Ubuntu 9.04 and Windows XP SP2 virtual machines. The Lenovo T61P ran a Windows XP SP2 host OS and also had VMware with a Ubuntu 9.04 Virtual Machine.

WireShark was used for packet analysis in addition to using debugs on the Cisco Catalyst 6509E switch to show how the attack was unleashed and the response/actions of the switch.

MAC Address Overflow Attack

The intent of the MAC Address Overflow attack is for the attacker to be able to overrun the Cisco Catalyst 6509E switches Content-Addressable Memory (CAM) table. This will force packets for all new flows to be flooded out all ports, allowing the attacker to monitor (sniff) incoming packets.

The following hardware/software was used in the tests:

Victim 1:

- Hardware: Lenovo PC
- Software: Windows XP
- IP Address: 10.1.0.51/24
- MAC Address: 00:1c:25:1a:58:86
- NIC: Linksys USB300M (USB-to-Ethernet 10/100)
- Cisco Catalyst 6509E Port: GE 1/13

Victim 2:

- Hardware: MacBook Pro
- Software: Mac OS X 10.5.7 (and an FTP Client)
- IP Address: 10.1.0.61/24 (Static IP, excluded from the DHCP Server Pool of IP Addresses)
- MAC Address: 00:23:df:a0:cf:50
- NIC: Internal 10/100/1000 Ethernet
- Cisco Catalyst 6509E Port: GE 1/1

Victim 3:

- Hardware: Cisco Catalyst 6509E with a Supervisor 720-3B
- Software: Cisco IOS Software 12.2(33)SXI1
- IP Address: 10.1.0.1/24 (Interface VLAN 7)
- MAC Address: 00:d0:01:39:dc:00
- Line Card: WS-X6748-GE-TX (10/100/1000 Ethernet)

Attacker:

- Hardware: Apple MacBook Pro
- Software: Parent OS is OS X 10.5.7. Running Ubuntu 9.04 OS in VMware Fusion
- Attack Tool: Ettercap NG-0.7.3
- IP Address: 10.1.0.60/24 (Static IP, excluded from DHCP Server Pool of IP Addresses)
- MAC Address: 00:23:69:48:b8:9c
- NIC: Linksys USB300M (USB-to-Ethernet 10/100)
- Cisco Catalyst 6509E Port: GE 1/2

Steps for the MAC Address Overflow Attack:

1. Clear the MAC-Address-Table (CAM) before running the attack
2. Ping the switch (Interface VLAN 7 = 10.1.0.1)
3. View MAC-address-table (“show mac-address-table”)
4. Edit the Ettercap Configuration File (“etter.conf”)
5. Open Ettercap attack tool inside Ubuntu OS
   a. Select “Promisc Mode” (Promiscuous)
   b. Select the interface to use (eth2 in our case)
   c. Scan for Hosts on the Subnet

d. View the Hosts List

e. Start Sniffing

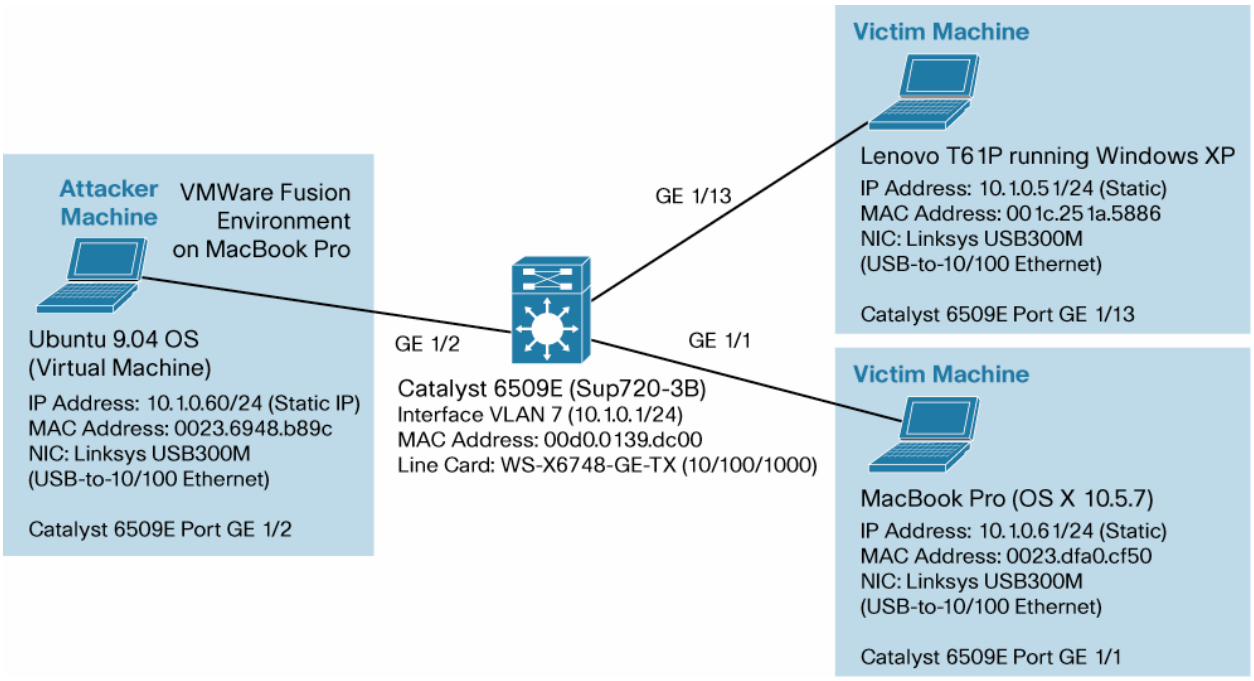
f. View Connections (should be blank at this point)

g. Start Random MAC Address Flooding Attack (“rand_flood” plugin)

6. Run Network Analyzer on Attacker to View Gratuitous ARPs to switch
7. View MAC Address Count (used/available) in CAM Table of switch
8. View MAC Address (CAM) Table
9. View FTP Client (MacBook Pro Victim) session
10. View Connections in Ettercap (10.1.0.61 to 10.1.0.51) for FTP
11. Kill Active FTP Connection between Victims
12. Mitigation for the MAC Address Overflow Attack
13. Summary

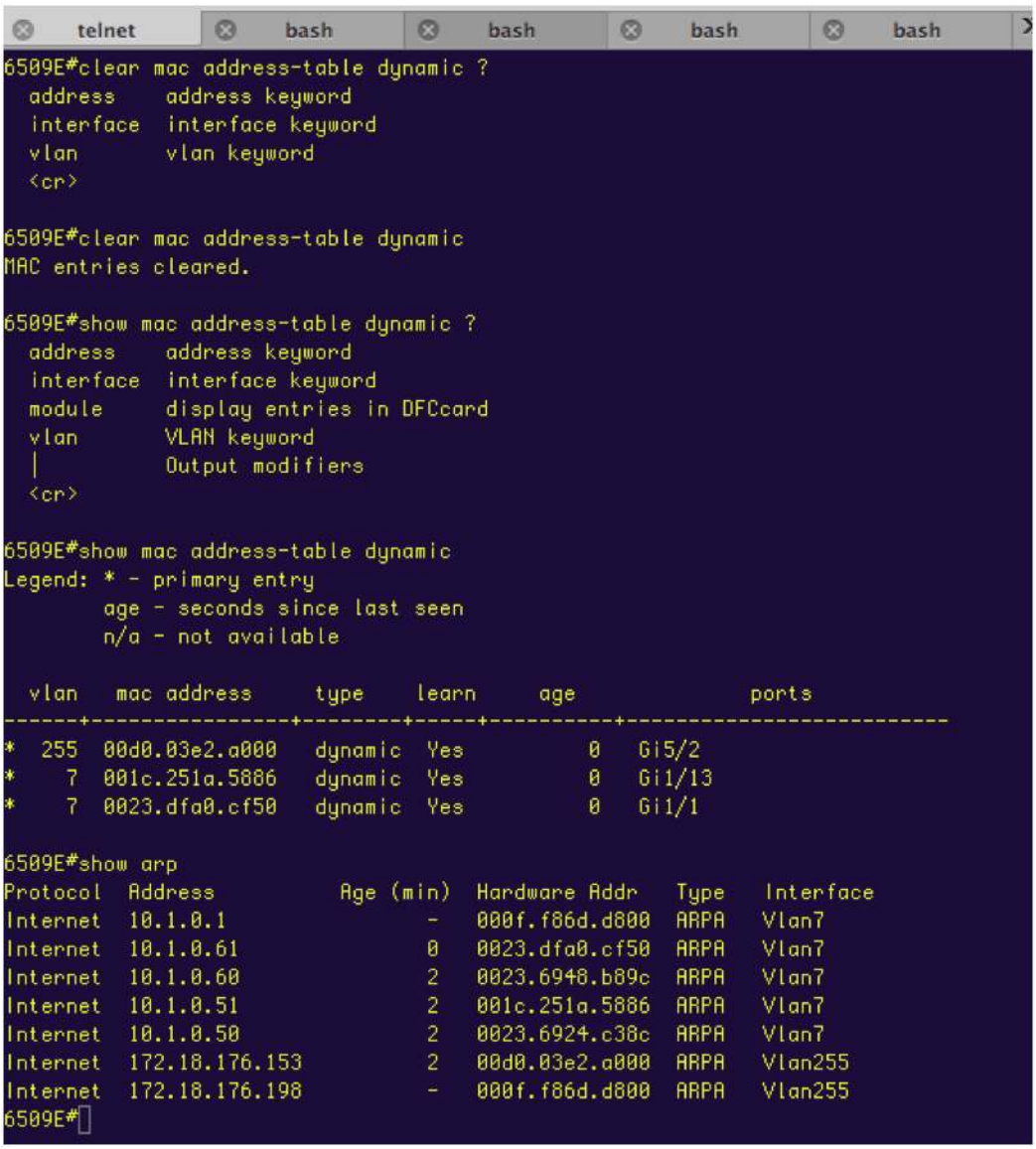
Figure 1 shows a diagram of the components used in the MAC Address Overflow Attack test. Note that the MacBook Pro (Victim) was connected to the Cisco Catalyst 6509E switch via the integrated 10/100/1000 Ethernet port ,and the Ubuntu Virtual Machine (running in VMware Fusion on the MacBook Pro) had a dedicated external Linksys USB300M 10/100 (USB-to-Ethernet) Network Adapter. Each was independent and no NAT or Bridging was being used between the host OS (OS X 10.5.7) and the Virtual Machine (Ubuntu 9.04).

Figure 1.



Before the MAC Address Overflow attack was started, a few configuration tasks were completed. Clearing out the dynamic MAC Address-Table on the Cisco Catalyst 6509E switch was performed first. This was done using the “**clear mac address-table dynamic**” command on the switch. Figure 2 shows this output and also displays the ARP cache.

Figure 2.



Next, using the Network Tools GUI on the Ubuntu 9.04 Virtual Machine, switch connectivity was verified by using ping (10.1.0.1 = Interface VLAN 7 on the Cisco Catalyst 6509E).

Figure 3.

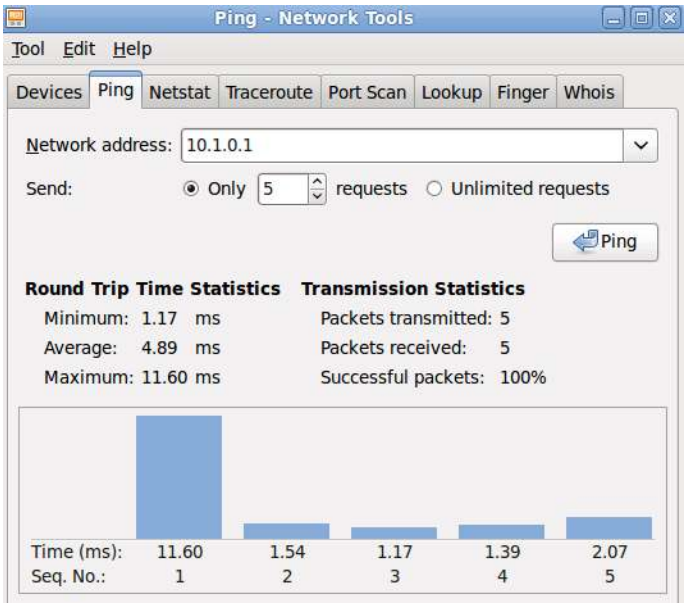


Figure 4 shows a snapshot of the current MAC Address Table on the Cisco Catalyst 6509E switch before the MAC Address flooding attack was started. Note that there are two very similar show commands. There is the "**show mac address-table dynamic**" and the "**show mac-address-table**." Both show the same information.

Figure 4.

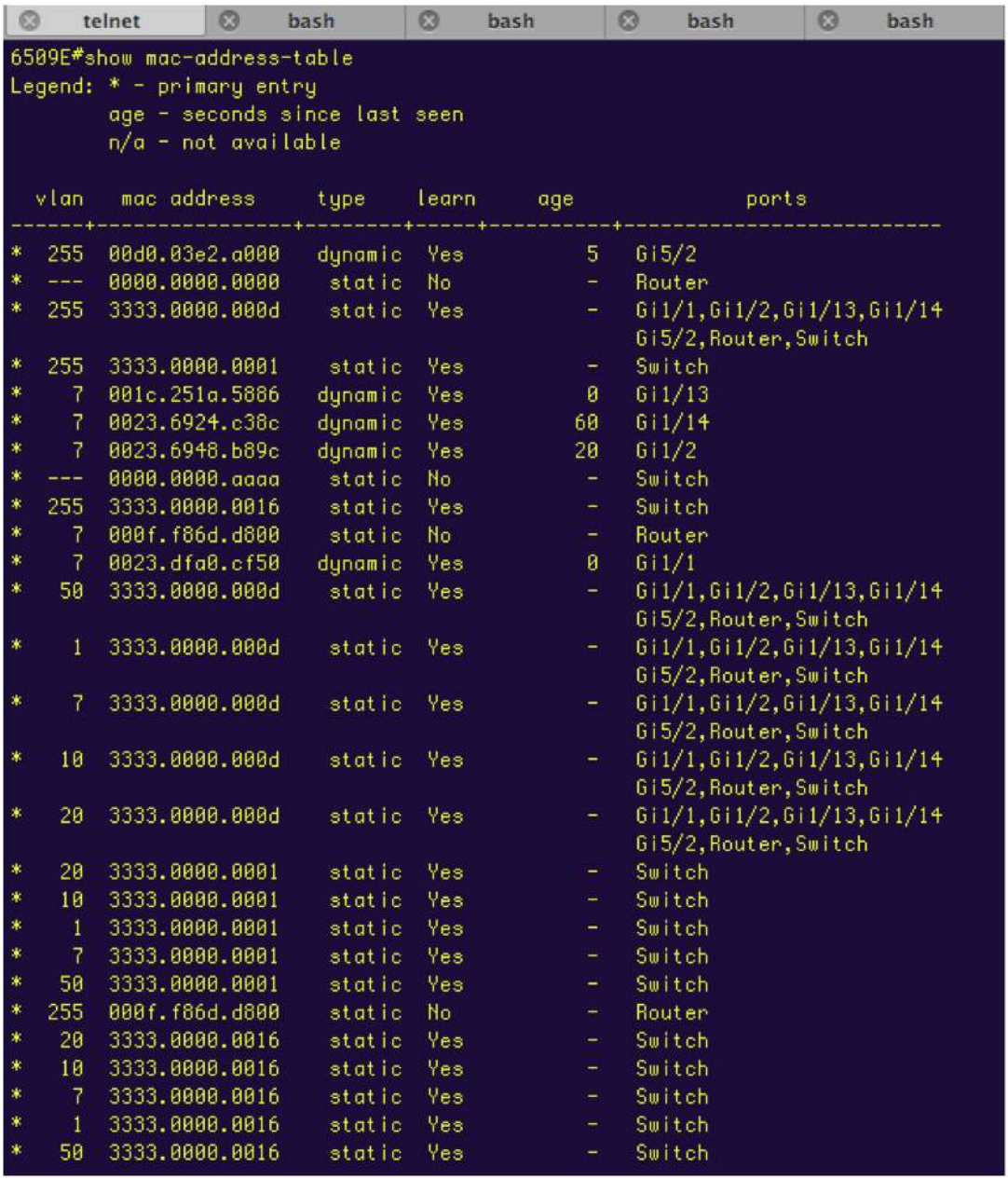


Figure 5 is a screen shot of the Cisco Catalyst 6509E console after clearing the ARP table and MAC Address table. The two hosts (10.1.0.61 – MacBook Pro running OS X 10.5.7 and 10.1.0.51 – Lenovo T61P running Windows XP) Ethernet are disconnected from the switch and then reconnected.

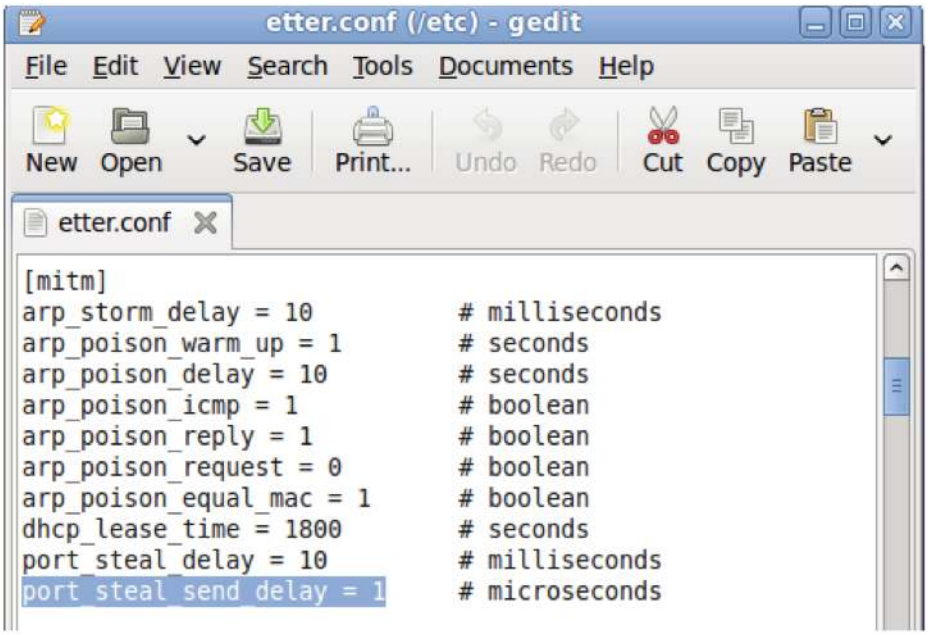
Figure 5.

```
6509E#
6509E#sho arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.0.1                -   000f.f86d.d800  ARPA   Vlan7
Internet  10.1.0.60               0   0023.6948.b89c  ARPA   Vlan7
Internet  10.1.0.50               0   0023.6924.c38c  ARPA   Vlan7
Internet  172.18.176.153          0   00d0.03e2.a000  ARPA   Vlan255
Internet  172.18.176.198          -   000f.f86d.d800  ARPA   Vlan255
6509E#
```

Ettercap Configuration (etter.conf) File Modifications

Note that some modifications were required to be made to the “etter.conf” file on the Ubuntu (Virtual Machine). One of the default settings in the etter.conf file (located in the /etc folder in Ubuntu 9.04) was changed. Also the “**port_steal_send_delay**” was modified from the default value of “2,000” to a value of “1.” Below is the screen shot of this action. This modification was required in order to be able to generate enough traffic to overflow the switch’s CAM table.

Figure 6.



Running the Ettercap Attack Tool

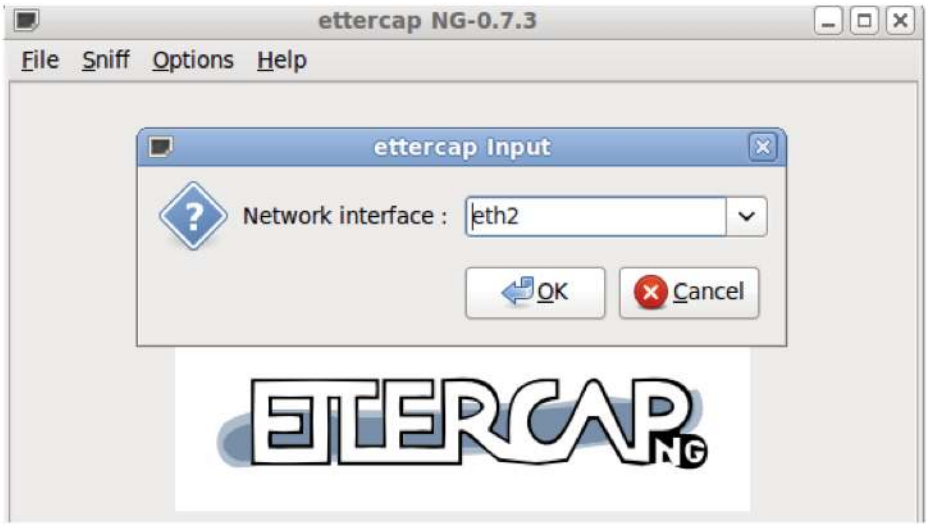
The Ettercap attack tool GUI was opened and “**Promiscuous Mode**” was selected on the Ubuntu Virtual Machine interface (eth2 in our case).

1. Select Promiscuous Mode

Figure 7.

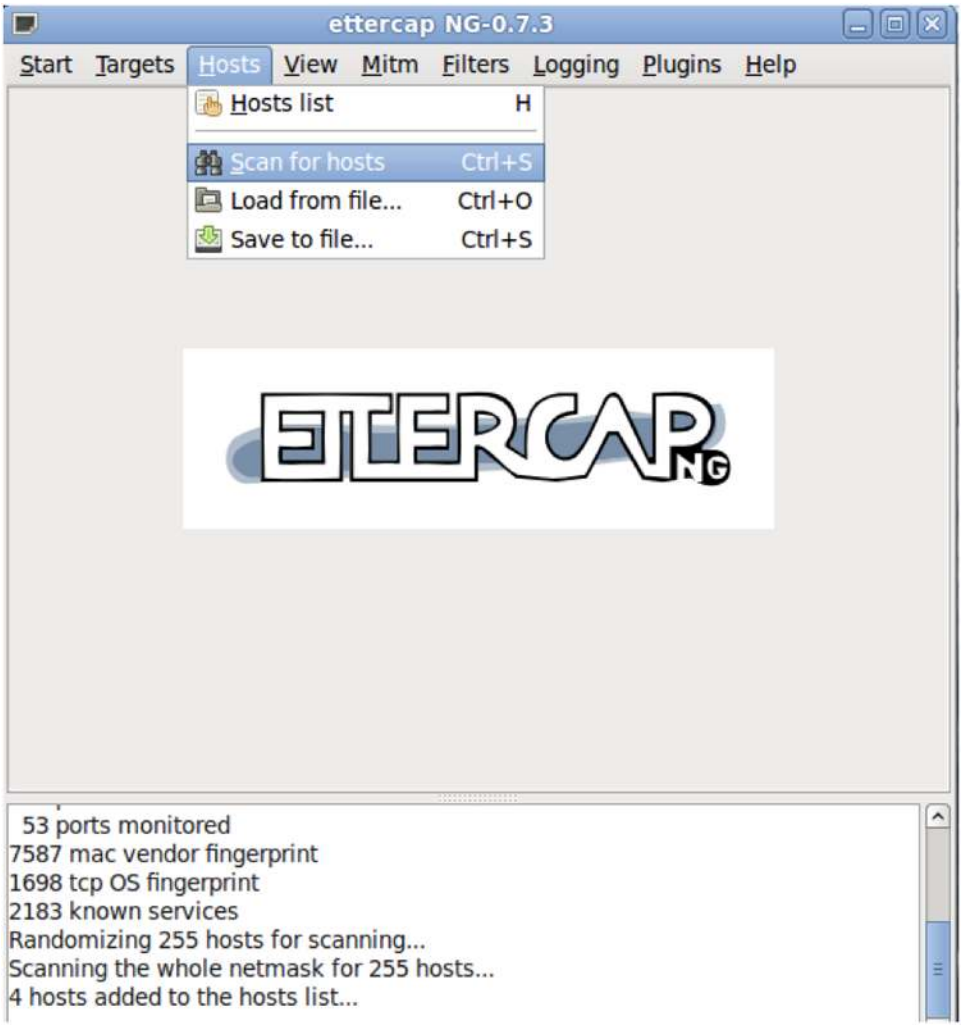
2. Select Interface

Figure 8.



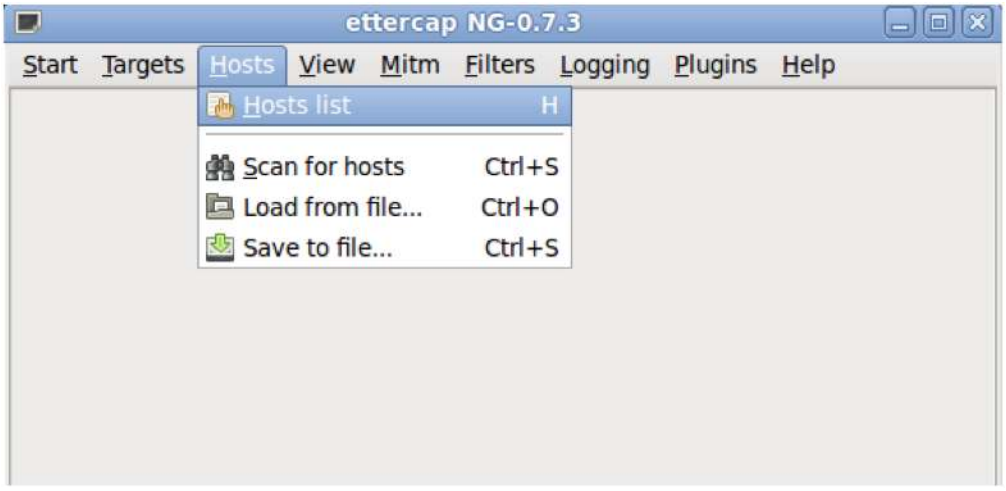
3. Scan for Host on the subnet

Figure 9.



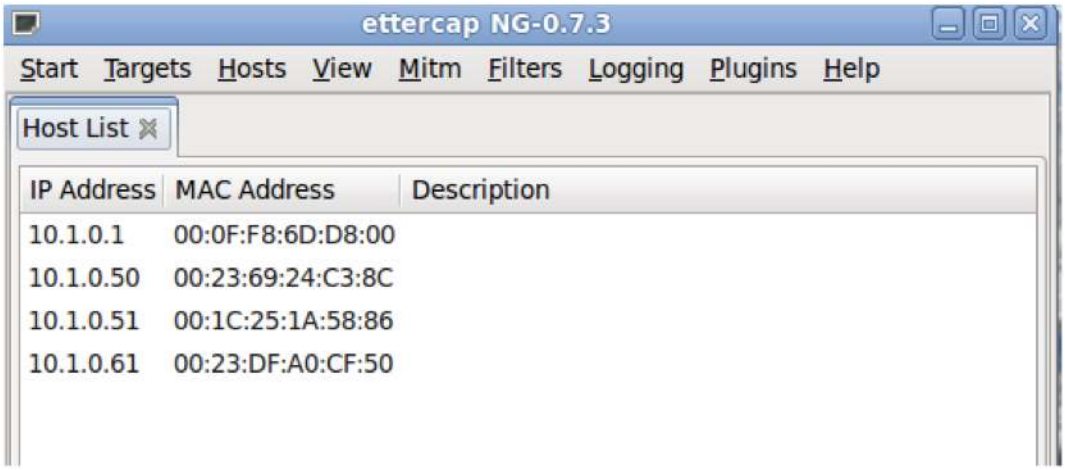
4. View the Host List

Figure 10.



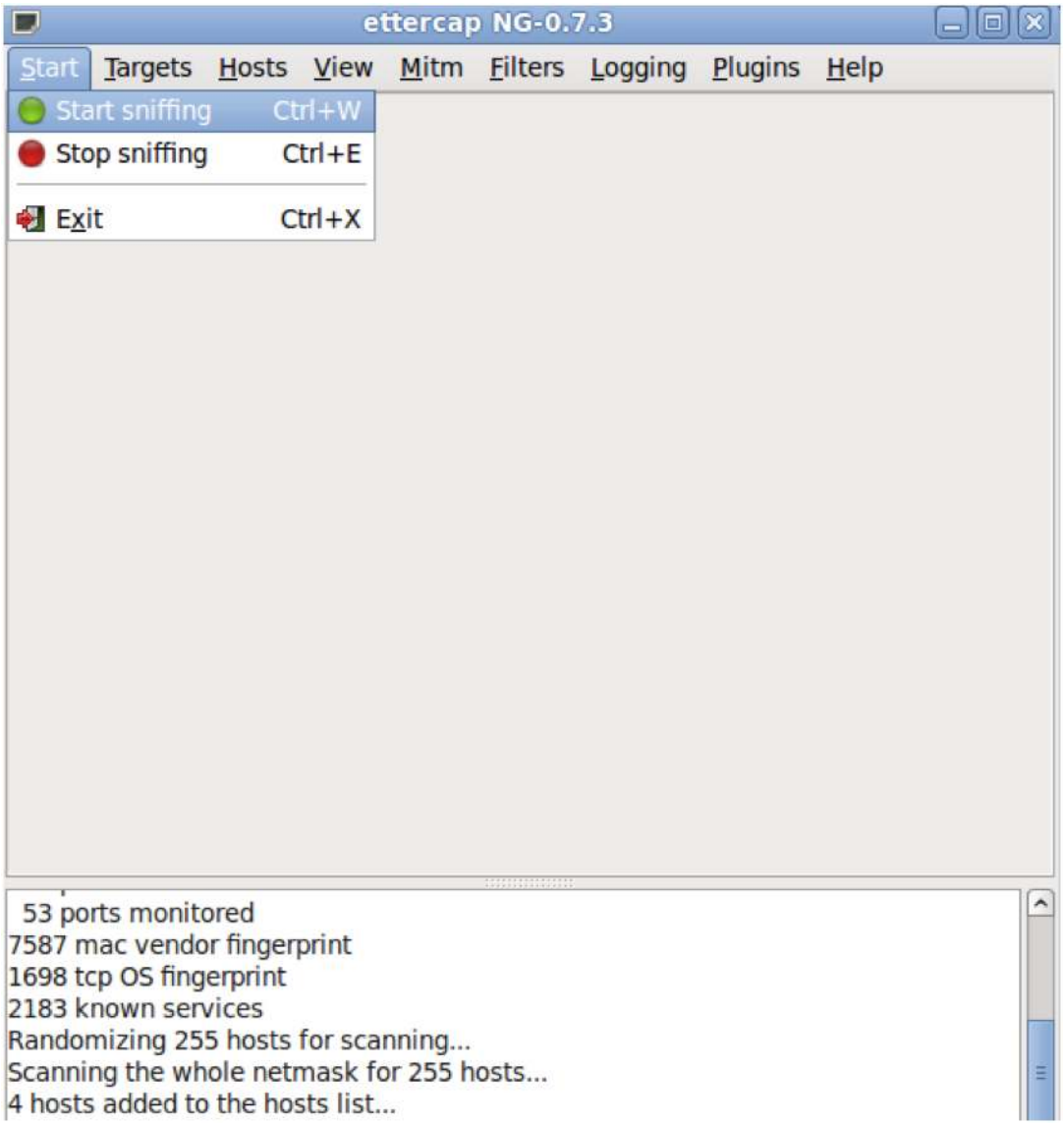
Output from Step 4

Figure 11.



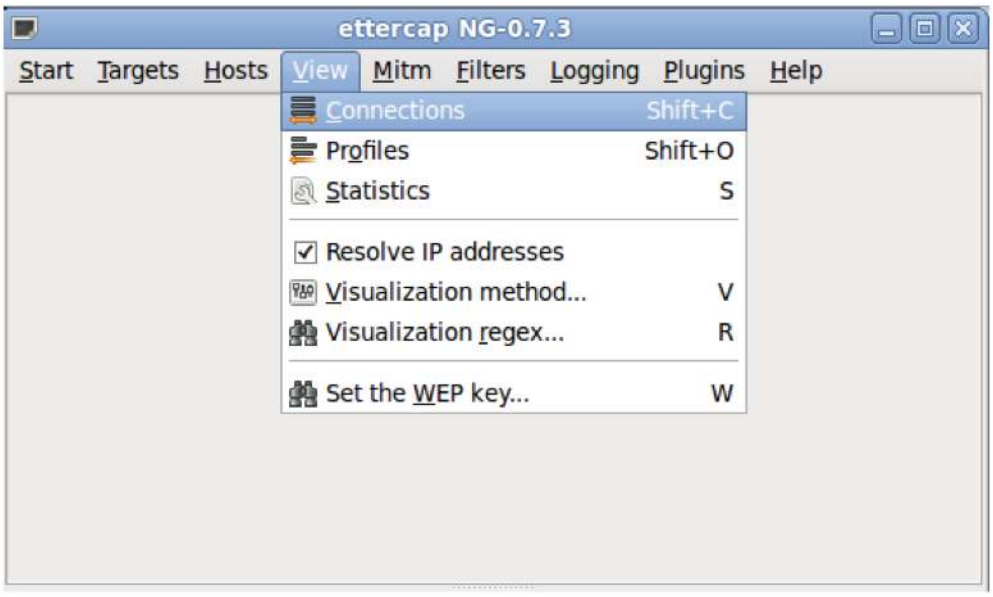
5. Start Sniffing (monitoring) mode

Figure 12.



6. View Connections. Note that no “Targets” were specified.

Figure 13.



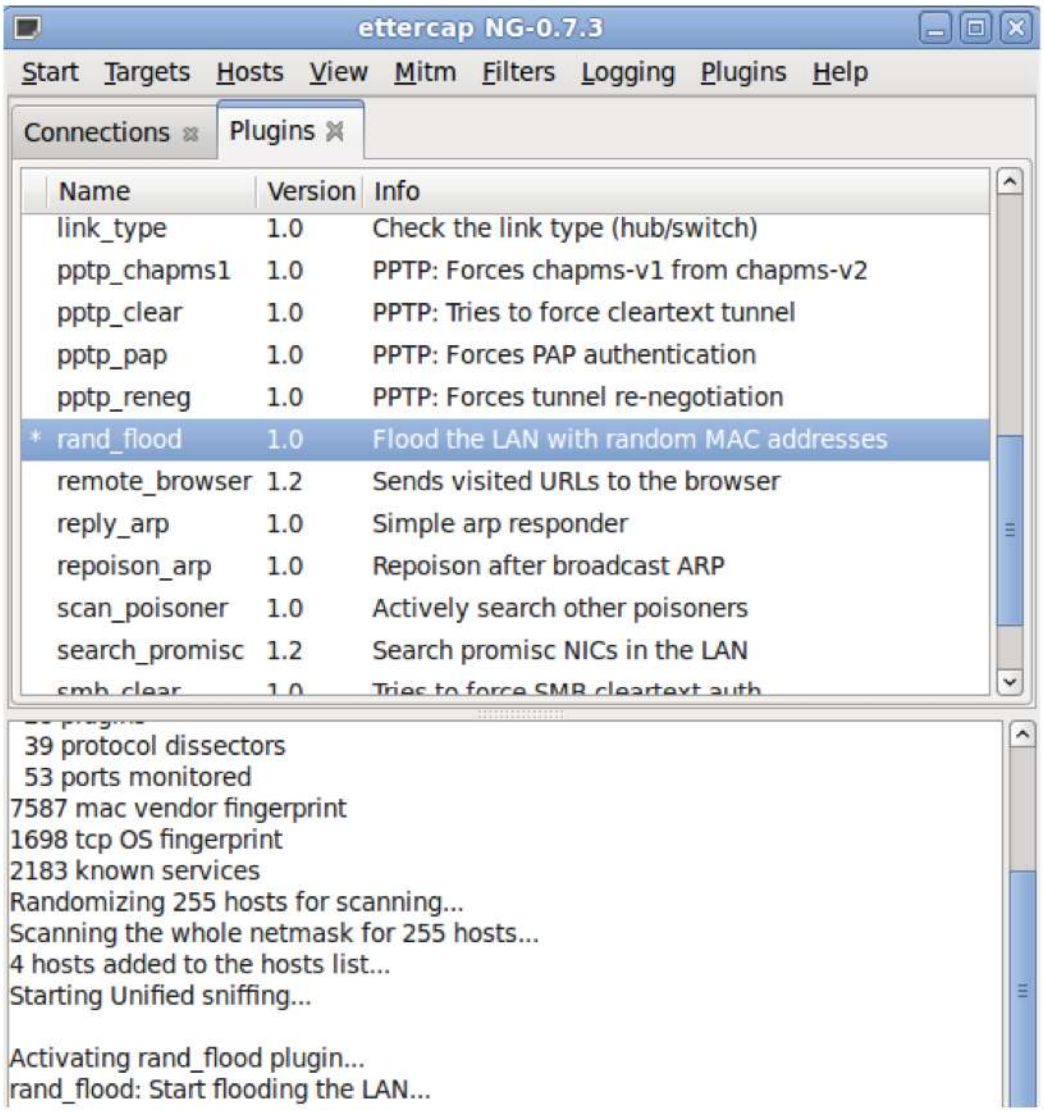
Initiate Ettercap “rand_flood” MAC Address Flooding Attack

Next the “rand_flood” attack plugin in Ettercap was run . That flooded the CAM Table on the Cisco Catalyst 6509E with random MAC Addresses. Once the MAC address table on the switch was exhausted of the available MAC Address space, Ettercap was setup in sniffing mode to see if the FTP session from host 10.1.0.61 (MacBook Pro) to host 10.1.0.51 (Windows XP – Lenovo T61P) would work.

7. Start the Random MAC Address Flooding Plugin

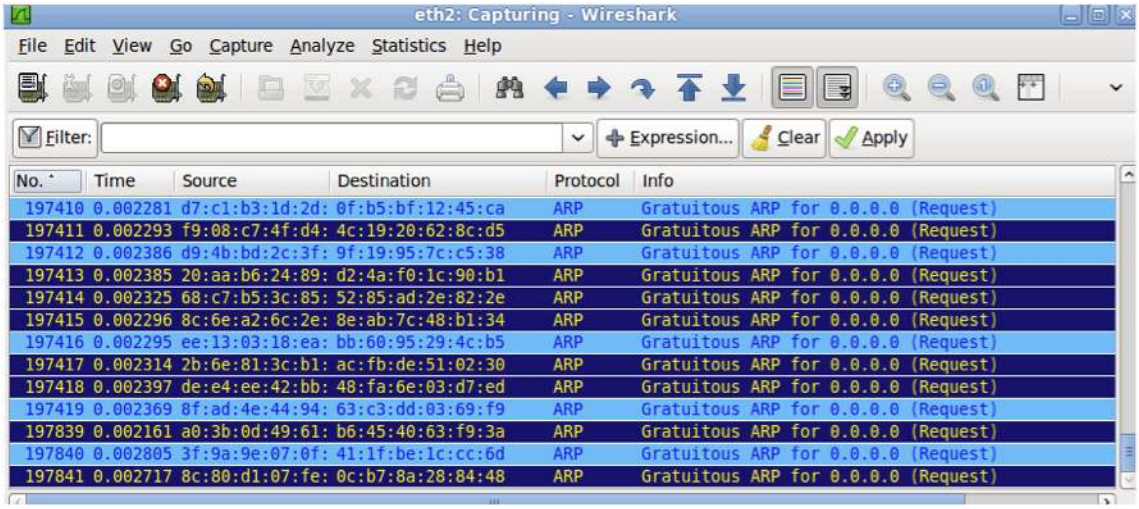
   The snapshots below show the Gratuitous ARPs being sent out by Ettercap to the Cisco Catalyst 6509E switch.

Figure 14.



8. Run Network Analyzer to Capture Gratuitous ARPs to 6509E Switch

Figure 15.



Next, Wireshark was run on the attacker machine (Ubuntu 9.04 Virtual Machine) that was also running the Ettercap attack tool GUI. After running the MAC Address Overflow attack (Ettercap “rand_flood” attack), it can be seen that all of the MAC Addresses in the Cisco Catalyst 6509E Content Addressable Memory (CAM) Table (Figure 16) were exhausted. Note that the Cisco Catalyst 6509E has a Supervisor 720-3B and is running 12.2(33)SXI1 Cisco IOS Software.

9. View Count of Available/Used MAC Address in CAM Table

Figure 16.

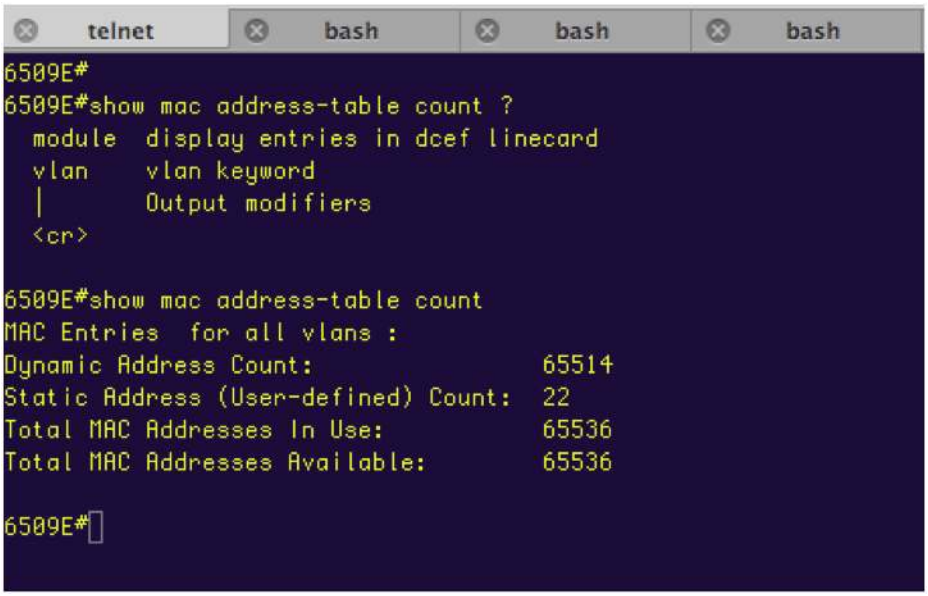


Figure 17 shows the first page (of many) listing the random MAC Addresses flooded in the Cisco Catalyst 6509E’s MAC Address Table from the Ubuntu Attacker machine running Ettercap’s “rand_flood” plugin.

10. MAC-Address-Table Listing (just first page)

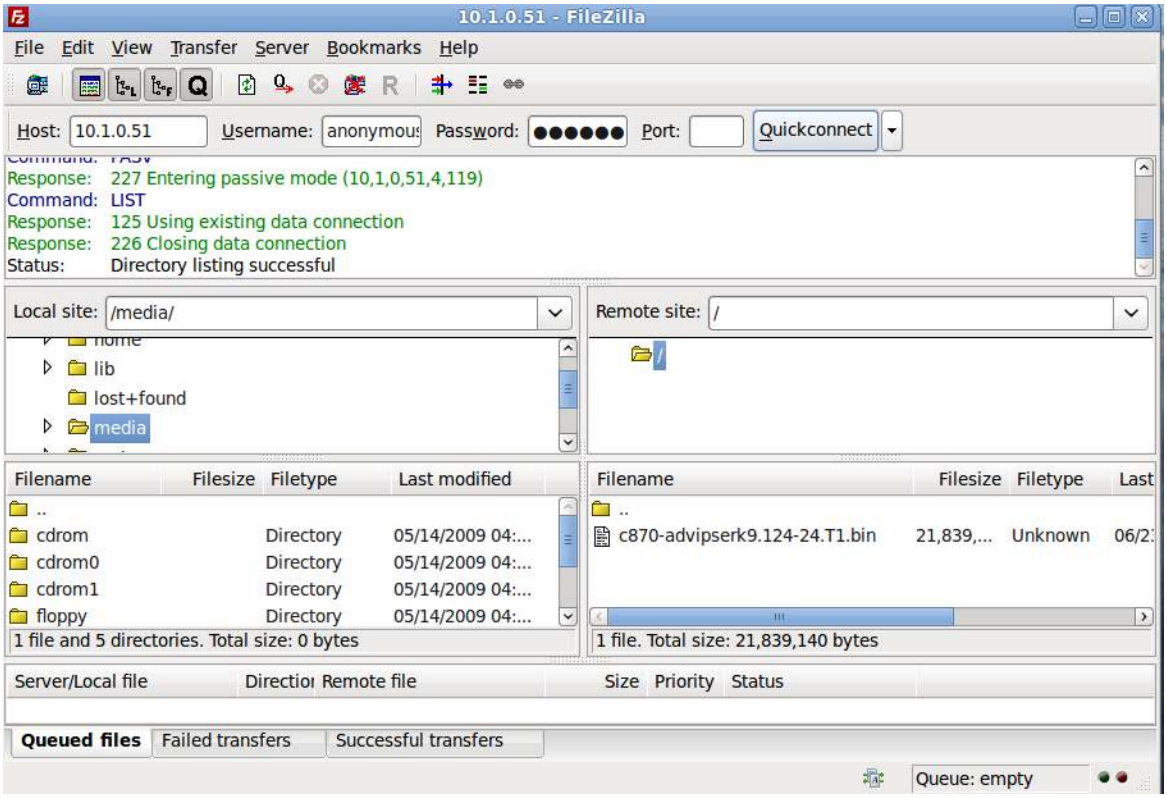
Figure 17.

```
6509E#show mac address-table
Legend: * - primary entry
        age - seconds since last seen
        n/a - not available

  vlan   mac address     type    learn     age              ports
------+---------------+--------+-----+----------+--------------------
----
*    7  f27e.9679.038d   dynamic  Yes       1120   Gi1/2
*    7  b875.907c.4480   dynamic  Yes       1120   Gi1/2
*    7  b0fd.fb4a.c0ec   dynamic  Yes       1120   Gi1/2
*    7  fedc.8f6f.7888   dynamic  Yes       1120   Gi1/2
*    7  f271.1738.db59   dynamic  Yes       1115   Gi1/2
*    7  aa38.204c.d3f7   dynamic  Yes       1110   Gi1/2
*    7  c627.f87a.57d5   dynamic  Yes       1105   Gi1/2
*    7  4432.4415.293f   dynamic  Yes       1100   Gi1/2
*    7  d8b5.021c.bcda   dynamic  Yes       1095   Gi1/2
*    7  06d3.9265.aaca   dynamic  Yes       1095   Gi1/2
*    7  0480.8839.be1c   dynamic  Yes       1090   Gi1/2
*    7  c4b0.ec12.f1df   dynamic  Yes       1090   Gi1/2
*    7  c82f.237b.5933   dynamic  Yes       1085   Gi1/2
*    7  e4d6.8144.4921   dynamic  Yes       1085   Gi1/2
*    7  26a7.623b.8ff0   dynamic  Yes       1085   Gi1/2
*    7  2e1e.9314.5afd   dynamic  Yes       1085   Gi1/2
*    7  8a91.f242.da51   dynamic  Yes       1125   Gi1/2
*    7  0e9c.1a56.8f83   dynamic  Yes       1115   Gi1/2
*    7  1cbb.7004.1039   dynamic  Yes       1115   Gi1/2
*    7  b664.f622.605e   dynamic  Yes       1115   Gi1/2
*    7  aaff.cb41.d2da   dynamic  Yes       1110   Gi1/2
*    7  32d3.915f.7a67   dynamic  Yes       1110   Gi1/2
*    7  9e71.8134.5dd3   dynamic  Yes       1110   Gi1/2
```

In Ettercap on the attacker's machine the "**sniffing**" window was viewed while getting ready to open up an FTP session from the FTP Client (10.1.0.61 – MacBook Pro) to the FTP Server (10.1.0.51 – Lenovo T61P). Keep in mind that the Ettercap "**rand_flood**" plugin was still running and that all 65,536 available MAC Addresses had been consumed on the Cisco Catalyst 6509E.

Figure 18.

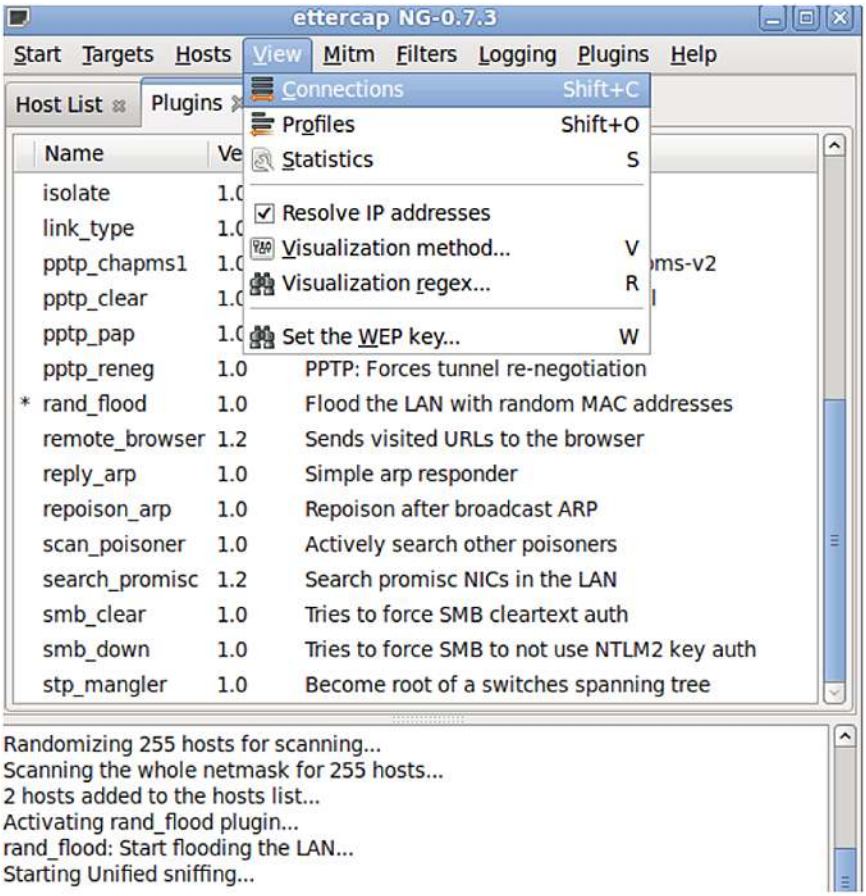


Snapshot of MacBook Pro (Victim 2) with an FTP session to Lenovo Windows XP (Victim 1)

The following shows the connections that showed up in Ettercap for Victim 2 (10.1.0.61 – MacBook Pro). The connections screen was dynamic, and one can see the "**state**" of the connection change from active, to idle, to closed.

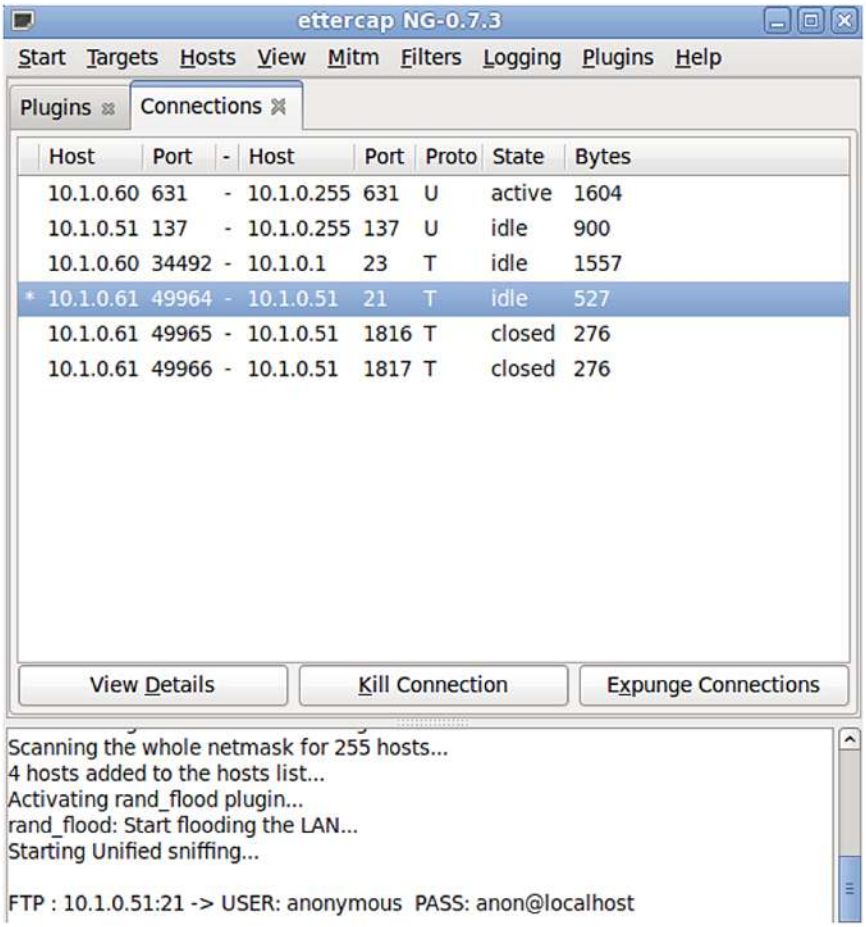
11. View Connections in Ettercap

Figure 19.



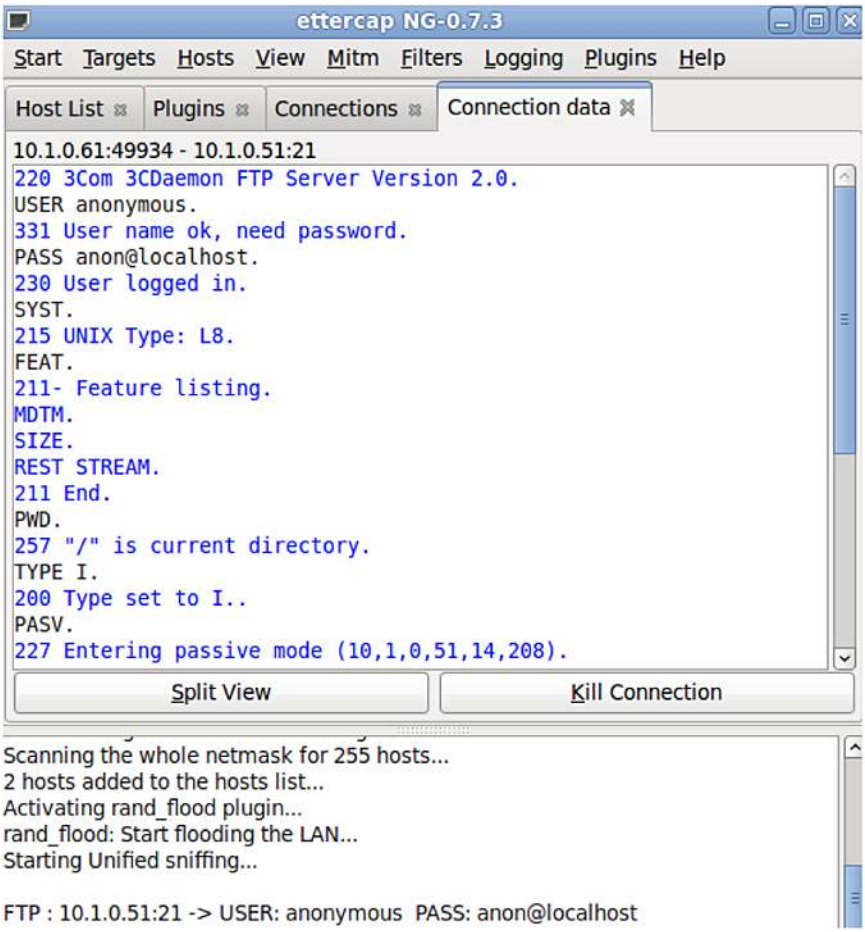
Undetected by the two Victim machines engaged in the FTP session, Ettercap was able to capture the username and password.

Figure 20.



12. Combined View of Active Connection (FTP session)

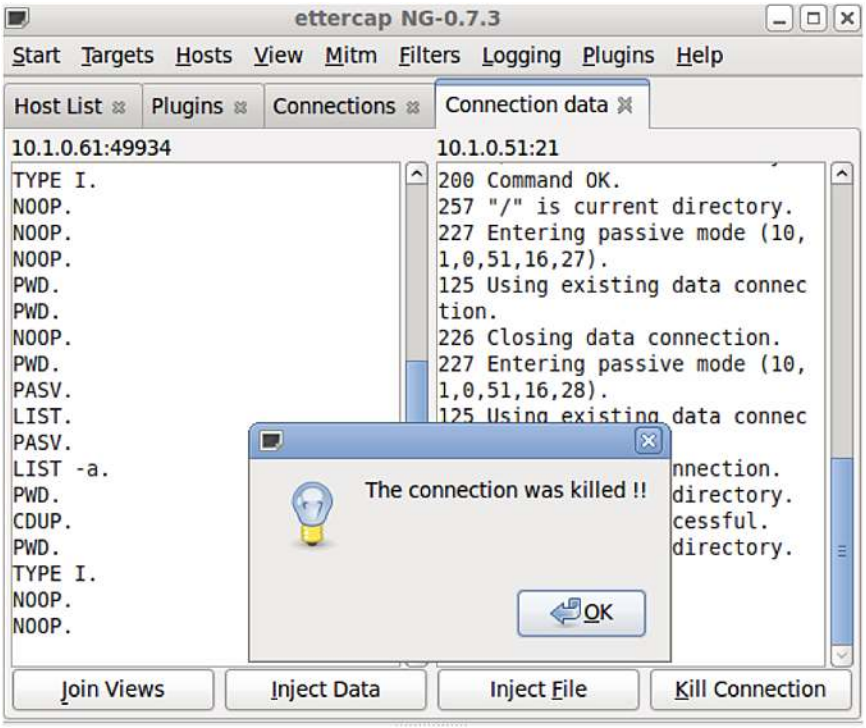
Figure 21.



Besides being able to see the FTP session from both Victims' viewpoints (including username/password/file listings/etc), the FTP connection could also be killed from within Ettercap.

13. Kill the FTP Connection Between Victim Machines

Figure 22.



Mitigation for the MAC Address Overflow Attack

There are a number of actions that can be configured on the Cisco Catalyst 6500 Series Switch to handle mitigation for this type of attack. Each of these mitigation techniques is reviewed in the following section. Take a look at the snapshot below from the Cisco Catalyst 6509E switch.

Figure 23.

```
6509E#config t
Enter configuration commands, one per line.  End with CNTL/Z.
6509E(config)#mac-address-table ?
  aging-time     Set MAC address table entry maximum age
  aging-type     Enable routed MAC entries aging
  learning       Enable a MAC table learning feature
  limit          Enter parameters for mac limit feature
  notification   Enable a Notification feature
  static         static keyword
  synchronize    Synchronize MAC address table entries in the system
```

The first mitigation feature that is discussed is the “mac-address-table limit” command and its related parameters.

Figure 24.

```
6509E(config)#mac-address-table limit ?
  action         Enter action
  interface      interface
  maximum        Enter max allowed entries
  notification   Enter type of Notification
  vlan           vlan number for  mac limit feature
  <cr>
```

The mitigation is started by limiting the number of MAC addresses that can be received on each of the access layer switch ports on the Cisco Catalyst 6509E. Since it is known that the attacker machine was running Ettercap (“rand_flood”) and was connected to GE1/2 on the switch, the global command “**mac-address-table limit interface gi1/2 maximum 15 action limit**” was added. This command limits the number of accepted MAC addresses inbound on port GE1/2 to 15.

Figure 25.

```
6509E#config t
Enter configuration commands, one per line.  End with CNTL/Z.
6509E(config)#mac-address-table limit interface gi1/2 maximum 15 action ?
  warning
  limit
  shutdown

6509E(config)#mac-address-table limit interface gi1/2 maximum 15 action limit
6509E(config)#
6509E(config)#end
```

Note the other two configurable options are “warning” and “shutdown.”

In the snapshot below, it can be seen that GE 1/2 (Attacker machine) reached its maximum limit of (15) MAC address entries. The switch enforced this limit and did not accept any more MAC addresses on the input interface GE 1/2.

Figure 26.

```
6509E#
6509E#show mac-address-table limit interface gi 1/2
 module    port      action    maximum   Total entries
--------+--------+-----------+---------+---------------
   5      Gi1/2     limit        15           15

6509E#
```

It can also be seen that the system messages caught in the logging buffer on the switch indicated that port GE 1/2 reached the maximum limit (15 in our case) configured.

Figure 27.

```
Jul 30 19:53:55.486: %MAC_LIMIT-SP-4-PORT_ENFORCE: Enforcing limit on Gi1/2 with Configured limit 15
Jul 30 19:54:56.118: %MAC_LIMIT-SP-4-PORT_ENFORCE: Enforcing limit on Gi1/2 with Configured limit 15
*Jul 30 19:55:18.262: DHCP_SNOOPING: checking expired snoop binding entries
Jul 30 19:55:56.694: %MAC_LIMIT-SP-4-PORT_ENFORCE: Enforcing limit on Gi1/2 with Configured limit 15
```

Another mitigation technique that is effective for the MAC Address Overflow attack is the **DHCP Snooping** and **Dynamic ARP Inspection (DAI)** combination. These mitigation features are also very effective for the ARP Poisoning (Man-In-The-Middle [MITM]) and DHCP Consumption attacks. The Cisco Catalyst 6509E switch is configured for DHCP Snooping and DAI using the following commands. Note that **DHCP Snooping is required** in order to be able to configure and use the Dynamic ARP Inspection (DAI) feature.

Figure 28.

```
6509E#config t
Enter configuration commands, one per line.  End with CNTL/Z.
6509E(config)#ip dhcp snooping
6509E(config)#ip dhcp snooping vlan 7
6509E(config)#no ip dhcp snooping information option
6509E(config)#ip arp inspection vlan 7
6509E(config)#ip arp inspection log-buffer entries 1024
6509E(config)#ip arp inspection log-buffer logs 1024 interval 10
6509E(config)#
6509E(config)#interface gi 1/47
6509E(config-if)#ip dhcp snooping trust
6509E(config-if)#ip arp inspection trust
6509E(config-if)#
6509E(config-if)#end
```

With the above commands enabled on the switch, the MAC Address Overflow attack is stopped. Prior to enabling DHCP Snooping and DAI, the Ettercap “**rand_flood**” attack exhausted all 65,536 available MAC Addresses in the CAM table. With these mitigation features enabled, here is what was seen.

Figure 29.

```
6509E#sho mac-address-table count
MAC Entries  for all vlans :
Dynamic Address Count:                  6
Static Address (User-defined) Count:  21
Total MAC Addresses In Use:             27
Total MAC Addresses Available:          65536
```

Notice the large difference between the number of available MAC addresses and the number of MAC Addresses In Use.

Note that for this test, because a static IP address was used on the attacker machine (10.1.0.60/24), a static DHCP binding was added in the switch configuration to bind the MAC address and IP address of the attacker machine to interface GE 1/2 on the switch. If this was not done, then the last two mitigation features configured would put the switch port out of service. Here is the format of the command.

Figure 30.

```
6509E#config t
Enter configuration commands, one per line.  End with CNTL/Z.
6509E(config)#ip source binding 0023.6948.b89c vlan 7 10.1.0.60 interface gi 1/2
6509E(config)#end
6509E#
```

Summary

The **MAC Address Overflow attack** is effective if the proper mitigation techniques are not in place on the Cisco Catalyst 6500 series switch. By using publicly (free) and available Layer 2 attack tools found on the Internet, anyone who understands how to setup and run these tools could potentially launch an attack on your network.

MAC address monitoring is a feature present on Cisco Catalyst 6500 Series switches. This feature helps mitigate MAC address flooding and other CAM overflow attacks by limiting the total number of MAC addresses learned by the switch on per-port or per-VLAN basis. With MAC Address Monitoring, a maximum threshold for the total number of MAC addresses can be configured and enforced on a per-port and/or per-VLAN basis. MAC address monitoring in Cisco IOS Software allows the definition of a single upper (maximum) threshold. In addition, the number of MAC addresses learned can only be monitored on a per-port or per-VLAN basis, and not a per-port-per-VLAN. By default, MAC address monitoring is disabled in Cisco IOS Software. However, the maximum threshold for all ports and VLANs is configured to 500 MAC address entries, and when the threshold is exceeded the system is set to generate a system message along with a syslog trap. These default values take effect only when MAC address monitoring is enabled. The system can be configured to notify or disable the port or VLAN every time the number of learned MAC addresses exceeds the predefined threshold. In our test, we used the “mac-address-table limit” command on the access layer port interface to configure the MAC address monitoring feature.

Two other Cisco IOS Software features that can be used to mitigate the MAC Address Overflow attack include **DHCP Snooping** coupled with **Dynamic ARP Inspection (DAI)**. By using the DHCP Snooping and Dynamic ARP Inspection (DAI) features, we are able to stop multiple types of Layer 2 attacks.

DHCP Snooping is a security feature capable of intercepting DHCP messages crossing a switch and blocking bogus DHCP offers. DHCP Snooping uses the concept of trusted and untrusted ports. Typically, the trusted ports are used to reach DHCP servers or relay agents, while untrusted ports are used to connect to clients. All DHCP messages are allowed on trusted ports, while only DHCP client messages are accepted on untrusted ports. As neither servers nor relay agents are supposed to connect to untrusted ports, server messages like DHCPOFFER, DHCPACK, and DHCPNAK are dropped on untrusted ports. In addition, DHCP Snooping builds and maintains a MAC-to-IP binding table that is used to validate DHCP packets received from untrusted ports. DHCP Snooping discards all untrusted DHCP packets not consistent with the information in the binding table. For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces. The DHCP Snooping binding table contains the MAC address, IP address, lease time in seconds, and VLAN port information for the DHCP clients on the untrusted ports of a switch. The information that is contained in a DHCP-snooping binding table is removed from the binding table when its lease expires or DHCP Snooping is disabled in the VLAN.

Dynamic ARP Inspection (DAI) is a security feature that helps prevent ARP poisoning and other ARP-based attacks by intercepting all ARP requests and responses, and by verifying their authenticity before updating the switch's local ARP cache or forwarding the packets to the intended destinations. The DAI verification consists primarily of intercepting each ARP packet and comparing its MAC address and IP address information against the

MAC-IP bindings contained in a trusted binding table. DAI discards any ARP packets that are inconsistent with the information contained in the binding table. The trusted binding table is dynamically populated by DHCP snooping when this feature is enabled. In addition, DAI allows the configuration of static ARP ACLs to support systems that use statically configured IP addresses and that do not rely on DHCP. DAI can also be configured to drop ARP packets with invalid IP addresses, such as 0.0.0.0 or 255.255.255.255, and ARP packets containing MAC addresses in their payloads that do not match the addresses specified the Ethernet headers.

Another important feature of DAI is that it implements a configurable rate-limit function that controls the number of incoming ARP packets. This function is particularly important because all validation checks are performed by the CPU, and without a rate-limiter, there could be a DoS condition.

DAI associates a trust state with each interface on the system, similar to DHCP Snooping. Packets arriving on trusted interfaces bypass all DAI validation checks, while those arriving on untrusted interfaces go through the DAI validation process. In a typical network configuration for DAI, all ports connected to host ports are configured as untrusted, while all ports connected to switches are configured as trusted. With this configuration, all ARP packets entering the network from a given switch will have passed the security check. By default, DAI is disabled on all VLANs, and all ports are configured as untrusted.

As discussed earlier, DAI populates its database of valid MAC address to IP address bindings through DHCP snooping. It also validates ARP packets against statically configured ARP ACLs. It is important to note that ARP ACLs have precedence over entries in the DHCP snooping database. ARP packets are first compared to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, then the packet will be denied even if a valid binding exists in the database populated by DHCP snooping.

Note that configuring DHCP Snooping is a prerequisite to configure Dynamic ARP Inspection (DAI). It is also worth noting that if you plan to use any static IP addresses are planned to be used when configuring DHCP Snooping and DAI, a static IP-to-MAC address mapping must also be entered in your Cisco Catalyst 6500 switches configuration. For instance, lets say that we want to assign a static IP-to-MAC mapping for the IP address 10.1.0.60 with the MAC address of 0023.6948.B89C for interface GE1/2 on VLAN 7 is required. The global configuration command on switch would be:

```
ip source binding 0023.6948.B89C vlan 7 10.1.0.60 interface Gi1/2
```

References

Solder, Carl (2006). “Cisco Networkers”

Cisco Systems, Inc. (2007). “Infrastructure Protection on Cisco Catalyst 6500 and 4500 Series Switches.”

Cisco Systems, Inc. (2009). “Configuring DHCP Snooping”

Cisco Systems, Inc. (2009). “Configuring Dynamic ARP Inspection.”

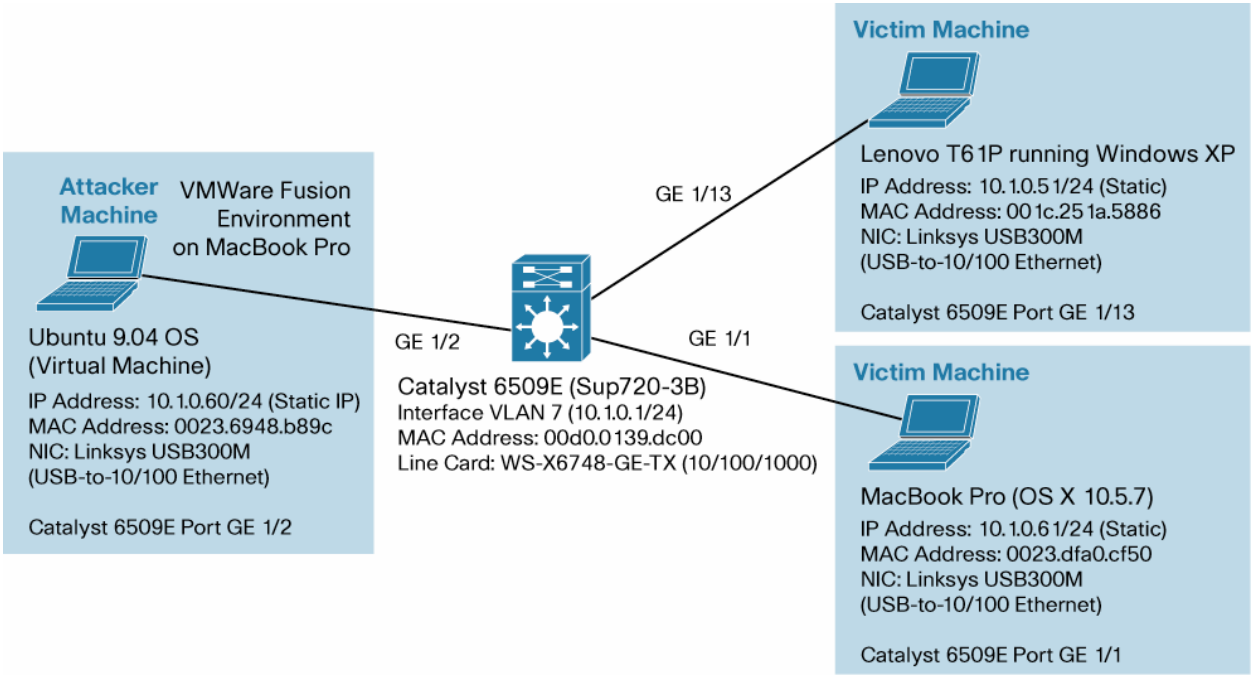
Cisco Systems, Inc. (2009). “Design Zone for Security.”

Hodgdon, Scott (2009). “CSSTG SE Residency Project Plan.”

Cisco Catalyst 6509E Configuration

(Mitigation configured)

MAC Address Overflow Attack




```
6509E#wr t
Building configuration...
Current configuration: 7513 bytes
!
upgrade fpd auto
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service counters max age 5
!
hostname 6509E
!
boot-start-marker
boot system flash disk0:
boot system disk0:s72033-adventerprisek9-mz.122-33.SXI1
boot-end-marker
!
security passwords min-length 1
logging buffered 4096 debugging
```

```
enable password xxxxx
!
no aaa new-model
ip subnet-zero!
!
             !
!************************************************************
!  Configuration for enabling DHCP Snooping
!************************************************************
ip dhcp snooping vlan 7
no ip dhcp snooping information option
ip dhcp snooping
no ip domain-lookup
!
!
!
!*************************************************************************
! Configuration for enabling Dynamic ARP Inspection (DAI)
!*************************************************************************
ip arp inspection vlan 7
ip arp inspection log-buffer entries 1024
ip arp inspection log-buffer logs 1024 interval 10
!*****************************************************************************************
! Because we are using a Static IP Address in Scenario 1, we must configure
! a source binding since we have enabled DHCP Snooping and Dynamic
! ARP Inspection (DAI). This will bind the MAC address on the NIC on the
! Attacker Machine to the static IP address connected to port GE1/2 on vlan 7
!*****************************************************************************************
ip source binding 0023.6948.B89C vlan 7 10.1.0.60 interface Gi1/2
!
!
!
!
!
```

```
!*************************************************************************************
! The following command limits the number of accepted MAC addresses
! inbound on port GE1/2 (Attacker Machine) to 15.
!*************************************************************************************
mac-address-table limit interface gi1/2 maximum 15 action limit
!
!
vtp domain CISCO
vtp mode transparent
mls netflow interface
no mls flow ip
mls cef error action reset
!
redundancy
 keepalive-enable
 mode sso
 main-cpu
  auto-sync running-config
spanning-tree mode pvst
!
!
!*************************************************************************************
! The "errdisable recovery cause all" command will automatically add the following
configuration
! commands to the switch
!*************************************************************************************
errdisable recovery cause udld
errdisable recovery cause bpduguard
errdisable recovery cause security-violation
errdisable recovery cause channel-misconfig
errdisable recovery cause pagp-flap
errdisable recovery cause dtp-flap
errdisable recovery cause link-flap
errdisable recovery cause gbic-invalid
errdisable recovery cause l2ptguard
errdisable recovery cause psecure-violation
errdisable recovery cause dhcp-rate-limit
errdisable recovery cause mac-limit
```

```
errdisable recovery cause unicast-flood
errdisable recovery cause vmps
errdisable recovery cause storm-control
errdisable recovery cause arp-inspection
errdisable recovery cause link-monitor-failure
errdisable recovery cause oam-remote-failure
errdisable recovery cause loopback
errdisable recovery interval 30
fabric timer 15
!
vlan internal allocation policy ascending
vlan access-log ratelimit 2000
!
vlan 7,10,20
!
vlan 50
 remote-span
!
vlan 255
!
!
!
!
!*************************************************************************
! Victim 2 connected to GE1/1. Victim 2 is a MacBook Pro running OS X 10.5.7. The
! IP address of Victim 2 is 10.1.0.61/24. This static IP address has been excluded
from
! the DHCP Server's IP Pool of addresses.
!*************************************************************************
interface GigabitEthernet1/1
 switchport
 switchport access vlan 7
 switchport mode access
!
```

```
!*******************************************************************************
! GE1/2 Connected to “Attacker” Ubuntu 9.04 Virtual Machine running on the
! MacBook Pro inside VMware Fusion. The Ubuntu 9.04 Attacker Machine will
! have an IP address of 10.1.0.60/24. Note that this static IP address is excluded
! from the DHCP Server’s IP Pool of addresses.
!*******************************************************************************
interface GigabitEthernet1/2
switchport
 switchport access vlan 7
 switchport mode access
!
interface GigabitEthernet1/3
  switchport
 switchport access vlan 7
 switchport mode access
 shutdown
!
interface GigabitEthernet1/4
 no ip address
 shutdown
!
interface GigabitEthernet1/5
 no ip address
 shutdown
!
interface GigabitEthernet1/6
 no ip address
 shutdown
!
!
interface GigabitEthernet1/7
 no ip address
 shutdown
!
interface GigabitEthernet1/8
 no ip address
 shutdown
```

```
!
interface GigabitEthernet1/9
 no ip address
 shutdown
!
interface GigabitEthernet1/10
 no ip address
 shutdown
!
interface GigabitEthernet1/11
 no ip address
 shutdown
!
interface GigabitEthernet1/12
 no ip address
 shutdown
!
!*******************************************************************************
! GE1/13 is connected to Victim 1, the Lenovo PC running Windows XP.
! It has an IP address of 10.1.0.51/24.
!*******************************************************************************
interface GigabitEthernet1/13
switchport
 switchport access vlan 7
 switchport mode access
!
interface GigabitEthernet1/14
 switchport
 switchport access vlan 7
 switchport mode access
 shutdown
!
interface GigabitEthernet1/15
 switchport
 shutdown
!
interface GigabitEthernet1/16
```

```
 no ip address
 shutdown
!
interface GigabitEthernet1/17
 no ip address
 shutdown
interface GigabitEthernet1/18
 no ip address
 shutdown
!
interface GigabitEthernet1/19
 no ip address
 shutdown
!
interface GigabitEthernet1/20
 no ip address
 shutdown
!
interface GigabitEthernet1/21
 no ip address
 shutdown
!
interface GigabitEthernet1/22
 no ip address
 shutdown
!
interface GigabitEthernet1/23
 no ip address
 shutdown
!
interface GigabitEthernet1/24
 no ip address
 shutdown
interface GigabitEthernet1/25
 no ip address
 shutdown
!
```

```
interface GigabitEthernet1/26
 no ip address
 shutdown
!
interface GigabitEthernet1/27
 no ip address
 shutdown
!
interface GigabitEthernet1/28
 no ip address
 shutdown
!
interface GigabitEthernet1/29
 no ip address
 shutdown
!
interface GigabitEthernet1/30
 no ip address
 shutdown
interface GigabitEthernet1/31
 no ip address
 shutdown
!
interface GigabitEthernet1/32
 no ip address
 shutdown
!
interface GigabitEthernet1/33
 no ip address
 shutdown
!
interface GigabitEthernet1/34
 no ip address
 shutdown
!
interface GigabitEthernet1/35
 no ip address
```

```
 shutdown
!
interface GigabitEthernet1/36
 no ip address
 shutdown
!
interface GigabitEthernet1/37
 no ip address
 shutdown
!
interface GigabitEthernet1/38
 no ip address
 shutdown
!
interface GigabitEthernet1/39
 no ip address
 shutdown
!
!
interface GigabitEthernet1/40
 no ip address
 shutdown
interface GigabitEthernet1/41
 no ip address
 shutdown
!
interface GigabitEthernet1/42
 no ip address
 shutdown
!
!
!
interface GigabitEthernet1/43
 no ip address
 shutdown
!
interface GigabitEthernet1/44
```

```
 no ip address
 shutdown
!
interface GigabitEthernet1/45
 no ip address
 shutdown
!
interface GigabitEthernet1/46
 switchport
 switchport access vlan 7
 switchport mode access
!
!*********************************************************************************
! GE1/47 connects to the Cisco 881 Router (port 0/5) that is acting as the external
! DHCP Server for scenario 2.
!*********************************************************************************
interface GigabitEthernet1/47
 description *** Connects to 881 Router Acting as DHCP Server [10.1.0.200] port 0/5
***
 switchport
 switchport access vlan 7
 switchport mode access
!*********************************************************************************
! Configuring GE1/47 as a trusted port for both DHCP Snooping and Dynamic ARP
! Inspection (DAI).
!*********************************************************************************
 ip arp inspection trust
 ip dhcp snooping trust
!
interface GigabitEthernet1/48
 switchport
 switchport mode dynamic auto
!
interface GigabitEthernet5/1
 no ip address
 shutdown
!
interface GigabitEthernet5/2
```

```
 no ip address
shutdown
!
interface Vlan1
 no ip address
 shutdown
!
!
!************************************
! Victim 3 is Interface VLAN 7.
!************************************
interface Vlan7
 ip address 10.1.0.1 255.255.255.0
!
!
!
router eigrp 100
 network 10.1.0.0 0.0.0.255
 network 172.18.176.0 0.0.0.255
 no auto-summary
!
ip classless
!
no ip http server
no ip http secure-server
!
control-plane
!
dial-peer cor custom
!
alias exec sdsb show ip dhcp snooping binding
!
line con 0
 exec-timeout 0 0
line vty 0 4
 session-timeout 800
 password 12345
```

```
 login
 length 0
 transport preferred none
 transport input all
 transport output none
line vty 5 15
 login
 transport input lat pad udptn telnet rlogin ssh
!
scheduler process-watchdog terminate
scheduler switch allocate 1000 1000
scheduler allocate 400 100
mac-address-table aging-time 480
!
end
```

Appendix

Introduction to Ubuntu Install Guide for L2 Attack Tools

This appendix is intended for those users that are not familiar with the Ubuntu Linux OS and want to be able to quickly download and use the Layer 2 attack and monitoring tools (Ettercap, Yersinia, packETH and Wireshark) that are utilized throughout these series of Layer 2 attack whitepapers. The Synaptic Package Manager in Ubuntu is the GUI application for installing software packages in the dpkg format with the file extensions of ".deb." This dpkg format was the first Linux packaging to integrate dependency information. The Debian Package Mgmt. system database tracks which software packages are installed, which version is installed, and other packages that it is dependent on. This allows you to automatically identify, download, and install all dependent applications that are part of your original application installation selection. The Synaptic Package Manager by default will only point to those dpkg repositories that are supported by Ubuntu Linux. If you want the Synaptic Package Manager to point to repositories that are not supported by Ubuntu, you must add those repositories to /etc/apt/sources.list.

This appendix covers the complete installation of the Ettercap application, modification of its initial configuration file (parameter values that provide privilege level, rerouting capabilities, remote browser capabilities, etc to ettercap) and where and how to launch the application. The other L2 attack and monitoring tool installations (Yersinia, packETH and Wireshark) are not covered in their entirety as the process is similar. These applications do differ in where and how they are launched and the appendix will cover these unique differences in detail.

Listed below is a summary of the different attack scenario's and which L2 attack tools were used:

- **STP MiTM (Vlan) Attack**: Yersinia, Ettercap, Wireshark
- **STP MiTM (ISL) Attack**: Yersinia, Ettercap, packETH, Wireshark
- **ARP MiTM Attack**: Ettercap, Wireshark
- **MAC Overflow Attack**: Ettercap, Wireshark
- **DHCP Consumption Atack**: Yersinia, Wireshark

Ettercap

Ettercap Installation via Synaptic Package Manager

Select "**System>Administration>Synaptic Package Manager**" to load Synaptic; the GUI application to download, install and remove applications on Ubuntu Linux.

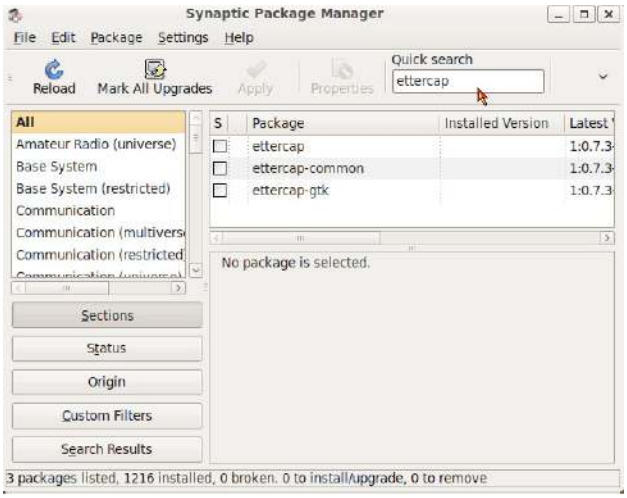
Figure 31.

You will be required to enter your password to perform Administrative Tasks such as installing software.

Figure 32.

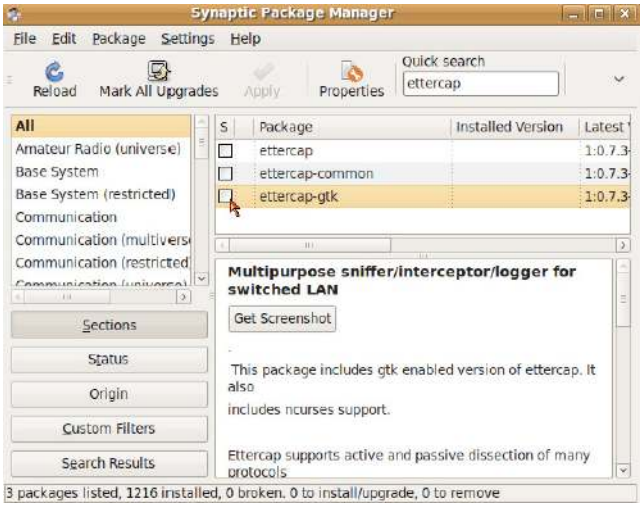
Type "**ettercap**" in the Quick Search Field and the Synaptic Package Manager will locate the application for installation.

Figure 33.



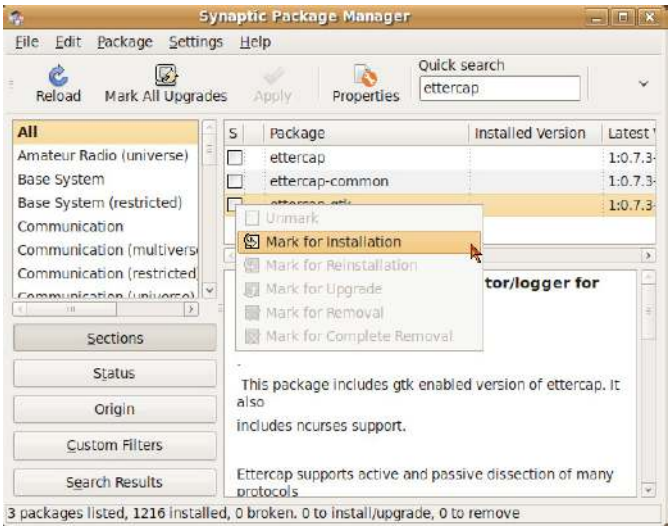
Select the "**ettercap-gtk**" application for the GUI version of Ettercap by left clicking your mouse on the application.

Figure 34.



Left click on "**Mark for Installation**."

Figure 35.

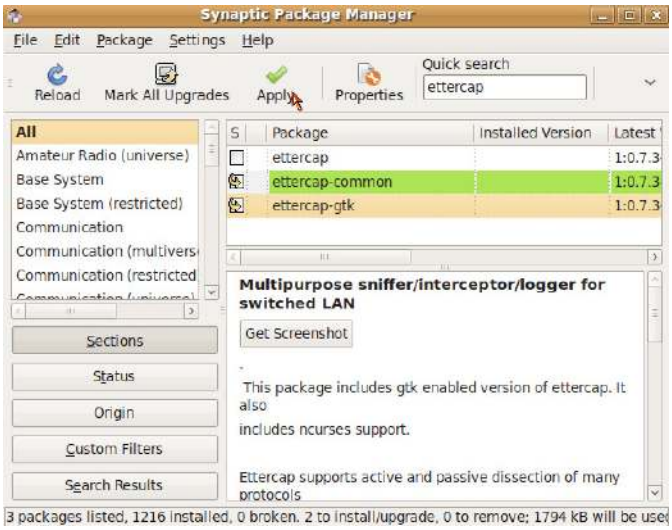


Additional application dependencies have been identified. Mark this additional application for installation.

Figure 36.

Ettercap has been marked along with its application dependency on "**ettercap-common**." Go ahead and hit "**Apply**" to install both applications.

Figure 37.



Hit "**Apply**" in the summary screen to confirm your installation selections.

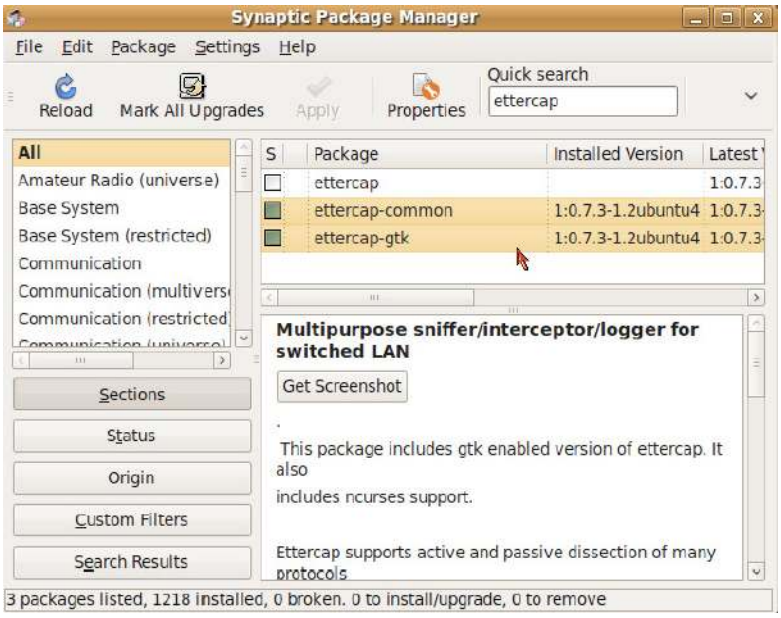
Figure 38.

When your installation is complete, the "**Changes applied**" window will pop up. Close this window to proceed.

Figure 39.

Synaptic will indicate that the applications have been successfully installed by the filled green boxes next to the application.

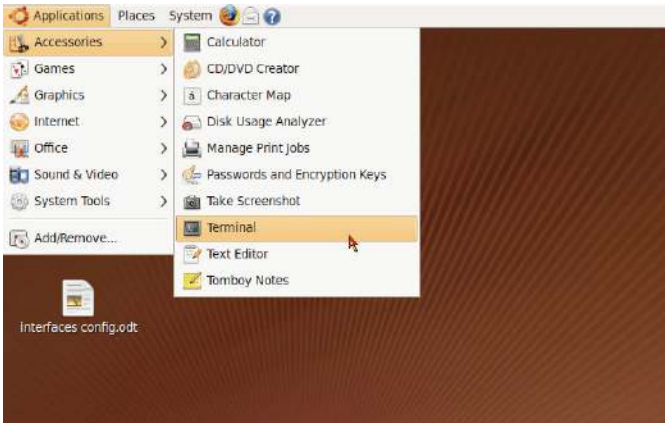
Figure 40.



Modification of Ettercap initialization file "/etc/etter.conf"

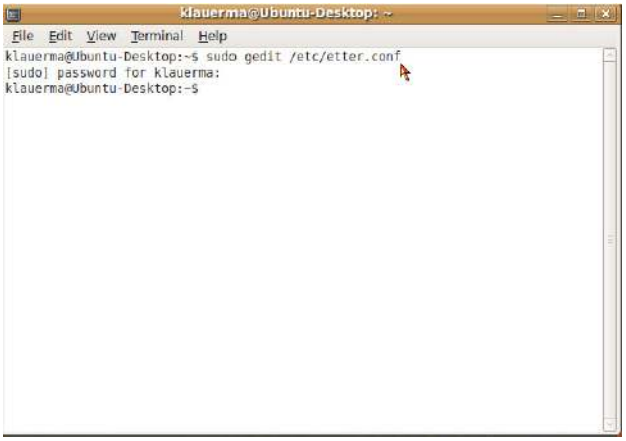
We are going to use the terminal shell to modify the **"/etc/etter.conf**" file. Select "**Applications>Accessories>Terminal**" to open a terminal window.

Figure 41.



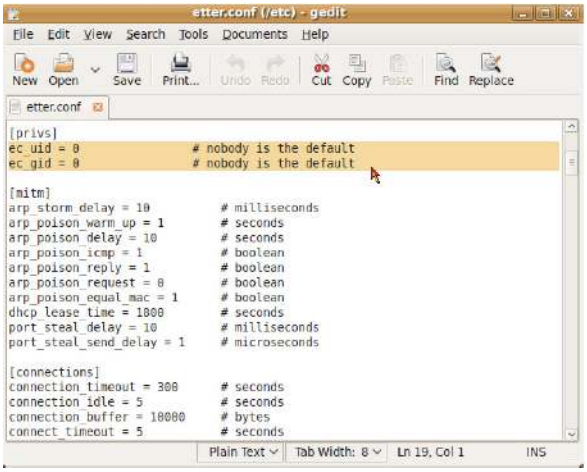
When the terminal window opens, type "**sudo gedit /etc/etter.conf**" to open the file with the appropriate privileges to modify the file. You will be prompted for the root password to continue.

Figure 42.



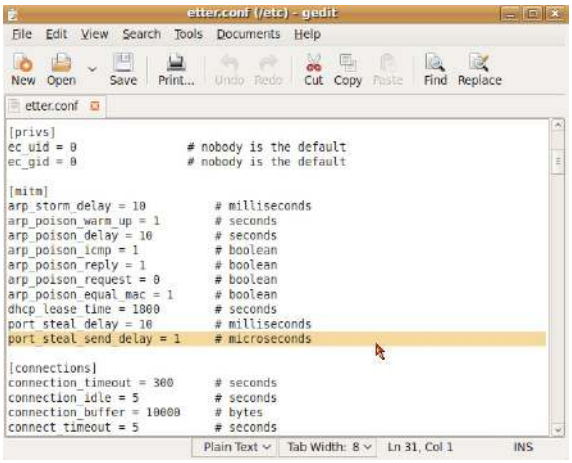
The "**ec_uid**" and "**ec_gid**" values both need to be changed to "**0**" to provide the appropriate privilege level to Ettercap upon execution. The default values are "**65534**." Change these to "**0**" as depicted below.

Figure 43.



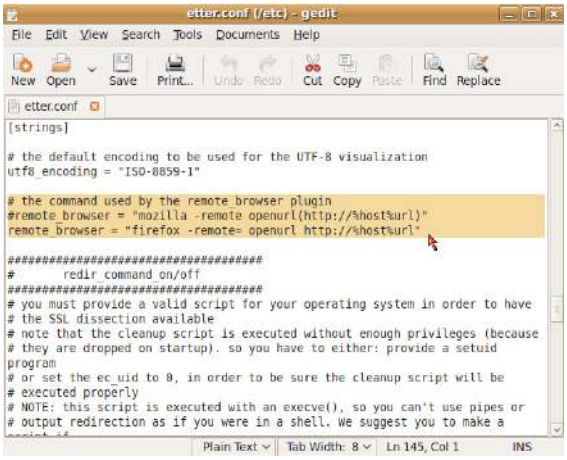
The "**port_steal_send_delay**" value needs to be changed to "**1**" microseconds. The default value is "2000" microseconds for the port steal send delay. We need to populate arp tables faster than Cisco's default ARP table timeout values can clear them.

Figure 44.



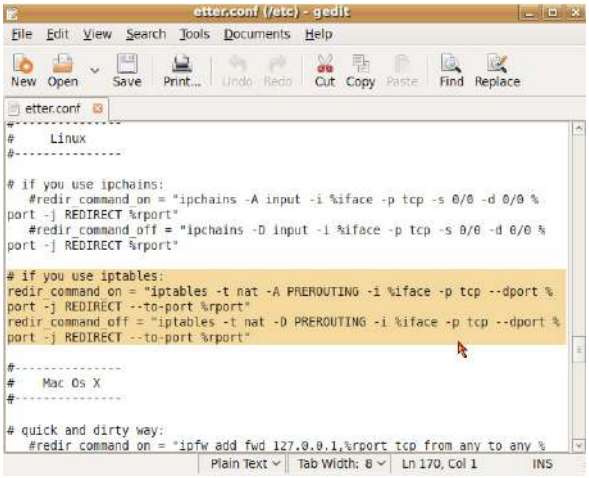
Comment out "#" the original remote browser command line and add the remote_browser command line as entered below—**mozilla is replaced with firefox**. This fixes the MiTM remote browsing plugin within ettercap. This enables us to view the same web pages as a victim in real time.

Figure 45.



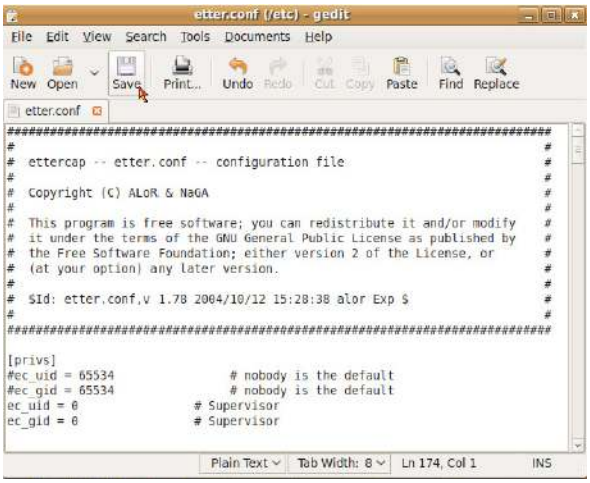
Uncomment the iptables commands by removing the "#" symbol. This will allow you to reroute traffic when performing a MiTM attack on behalf of the gateway and the victim.

Figure 46.



When you have completed these modifications, hit the "**save**" tab at the top of the editor.

Figure 47.



Launching Ettercap

Ettercap can be launched from "**Applications>System Tools**" or you can right click on "**ettercap**" from "**System Tools**" and install a launcher to the desktop.

Figure 48.

Ettercap can now be launched from the Ubuntu Desktop.

Figure 49.

Double clicking on the Ettercap icon launches the following Ettercap application.

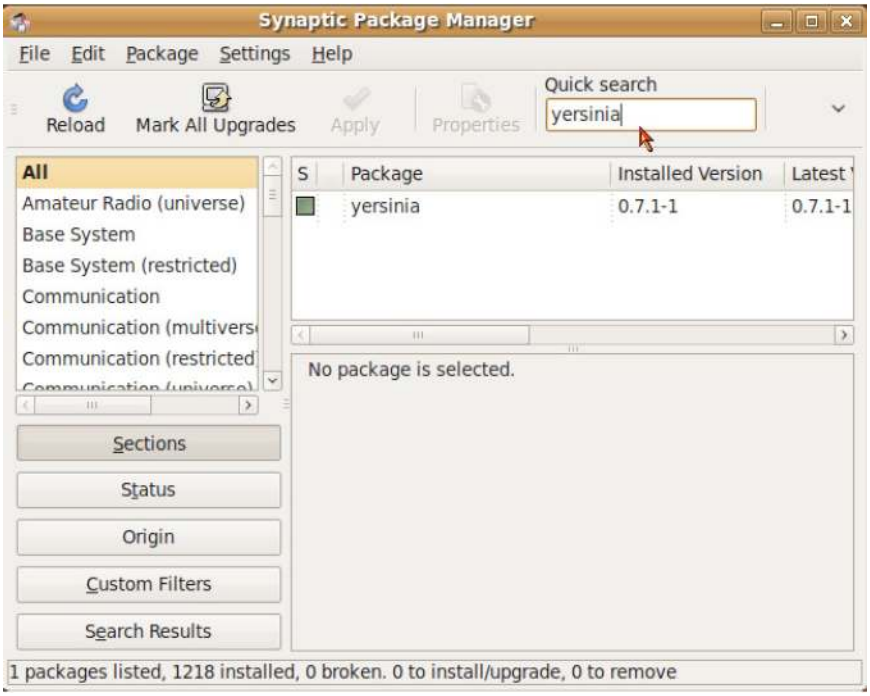
Figure 50.

Yersinia

Yersinia Installation via Synaptic Package Manager

Follow the same process as outlined in Figures 53 to 62 to install the Yersinia (Figure 73) application. The below figure depicts the currently installed application—as denoted by the green box next to the application.

Figure 51.



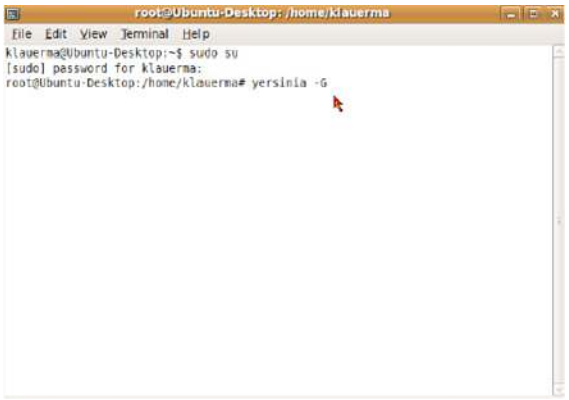
Launching Yersinia's Graphical Interface

Yersinia must be launched from a terminal window. Click on "**Applications>Accessories>Terminal**" to open a terminal window.

Figure 52.

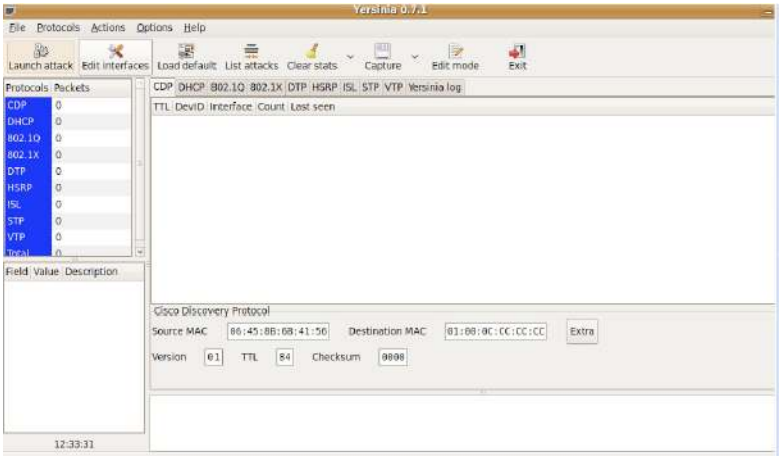
Give the terminal window root privilege by typing the command "**sudo su**" at the command prompt. You will be required to provide a root password. There are several different options to launch yersinia from the CLI—"-G " for Graphical or "-I" for interactive. The "**yersinia –G"** option loads the Graphical version of Yersinia depicted in Figure 75.

Figure 53.



The default Yersinia screen when using the Graphical option for launching the application.

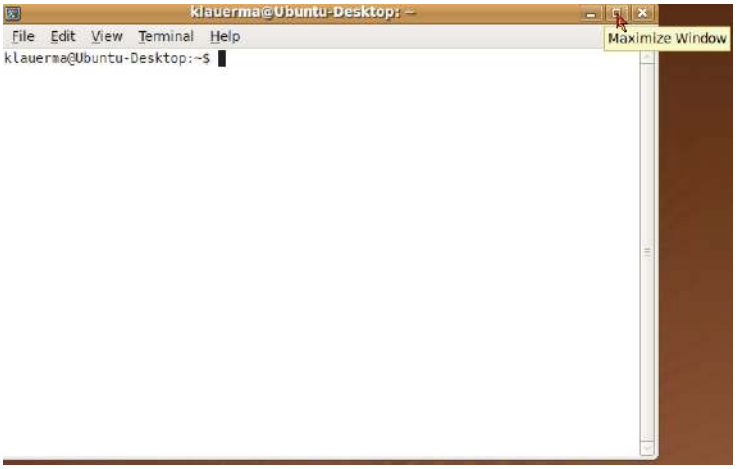
Figure 54.



Launching Yersinia's Interactive Interface

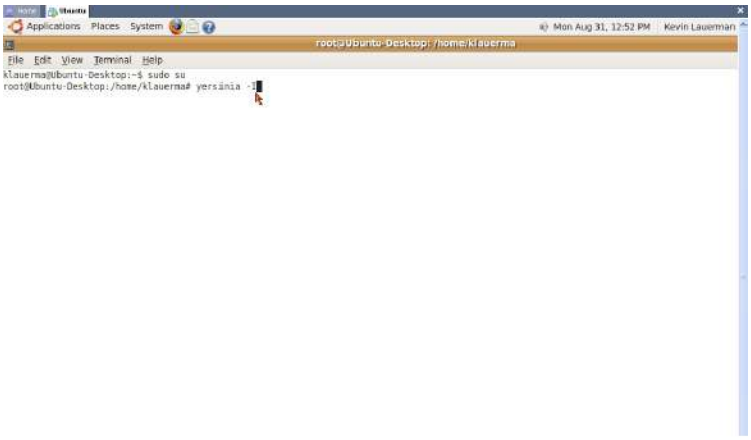
To launch the Interactive interface of Yersinia, you must be using a full size terminal window. Make sure you maximize your terminal session before launching the application with the "**-I**" option.

Figure 55.



Once the window is maximized launch Yersinia with the "**-I**" option from the CLI.

Figure 56.

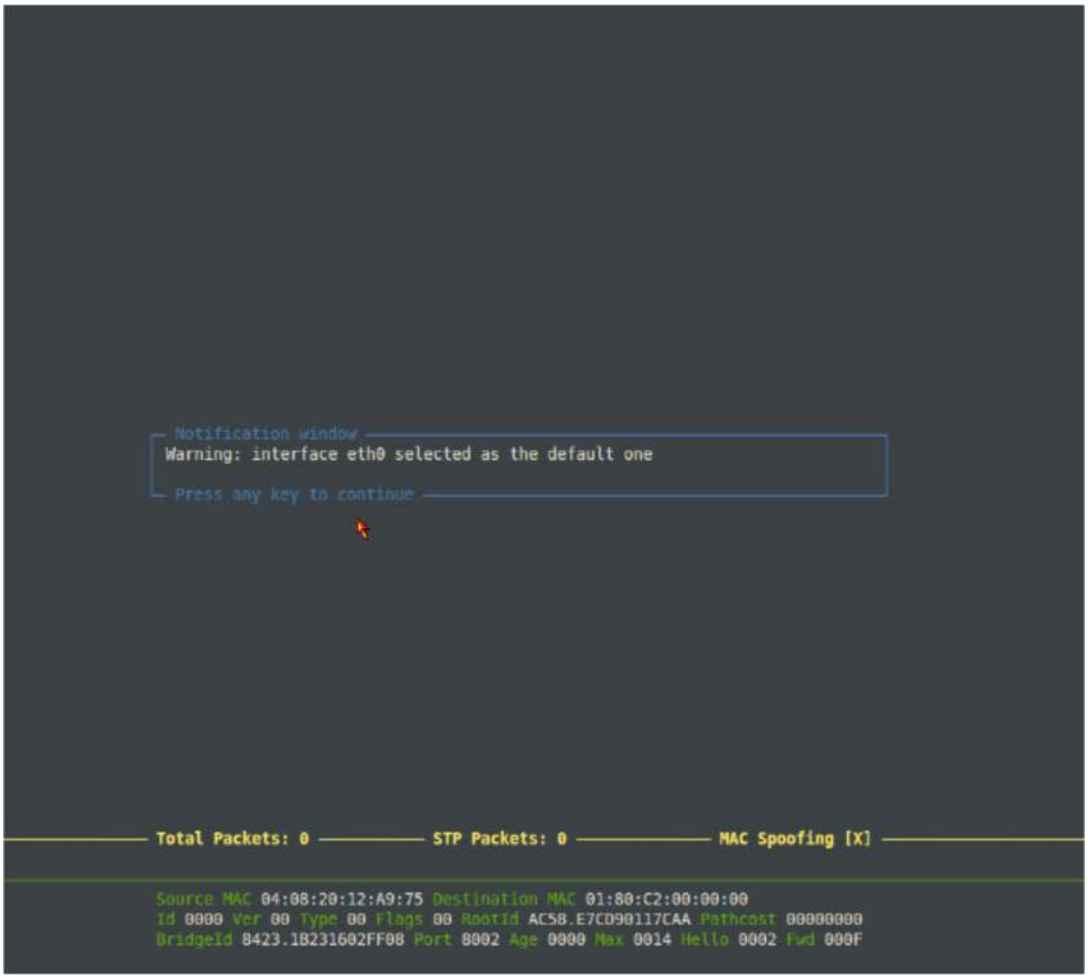


If your window was not maximized, you will receive the error message depicted below.

Figure 57.

The initial dialog screen informs you that eth0 has been selected as the default interface. You have an option to load an additional Ethernet interface if desired or to change the default interface. Hit any key to proceed.

Figure 58.



Type a lower case "**h**" to pull up the help screen of available commands. This is just another interface for using Yersinia. Our examples of L2 attacks all use the Graphical version of Yersinia. This should be enough to get you going if you choose to use the Interactive interface to Yersinia.

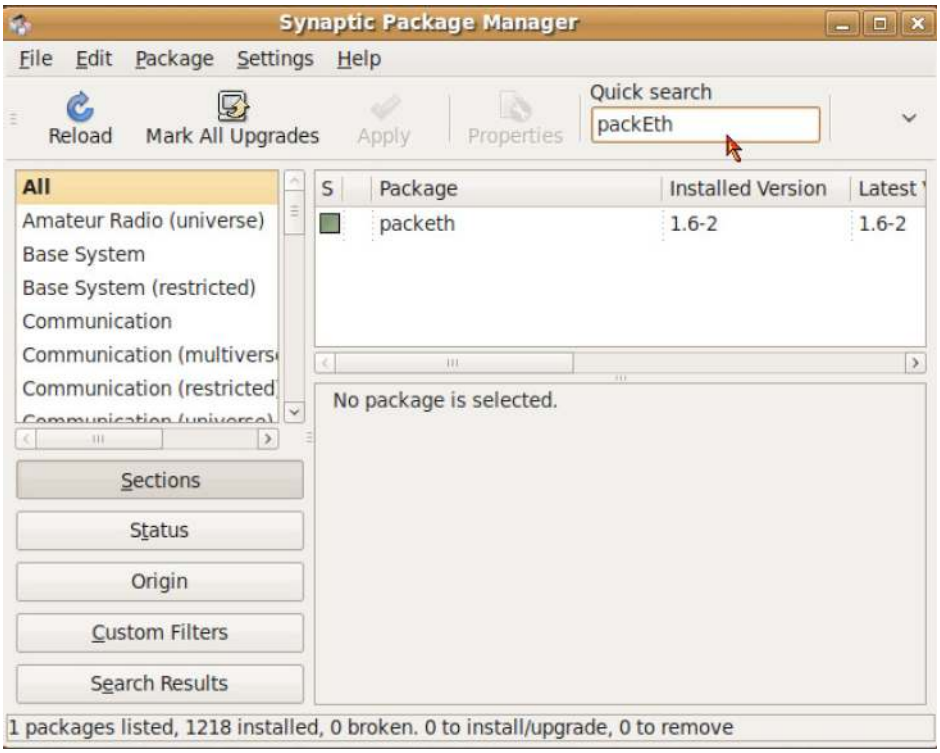
Figure 59.

packETH

packETH Installation via Synaptic Package Manager

Follow the same process as outlined in Figures 53 to 62 to install the packETH (Figure 82) application. The below figure depicts the currently installed application—as denoted by the green box next to the application.

Figure 60.



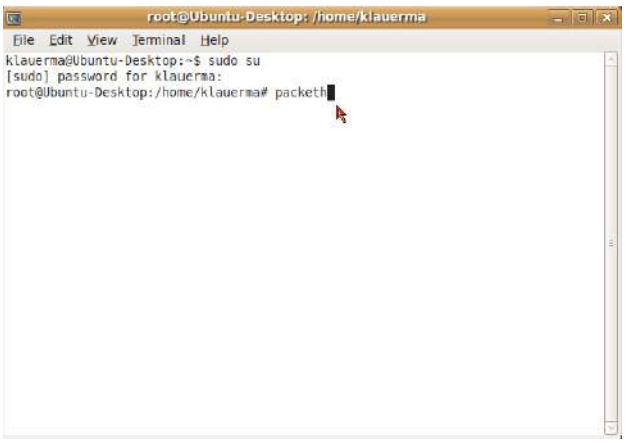
Launching packETH

packETH must be launched from a terminal window. Click on “Applications>Accessories>Terminal “ to open a terminal window.

Figure 61.

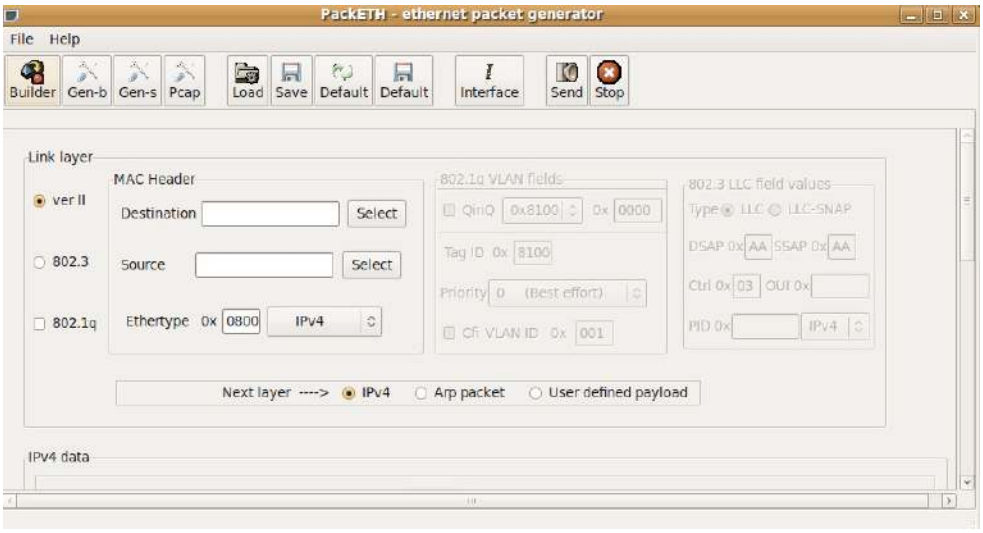
Give the terminal window root privilege by typing the command “**sudo su**” at the command prompt. You will be required to provide a root password. Once you have root privilege, type “**packeth**” at the command prompt and hit enter.

Figure 62.



This is a partial view of the default window that appears for packETH upon launching. You will notice that this packet generator supports ver II, 802.3 and 802.1q frames.

Figure 63.

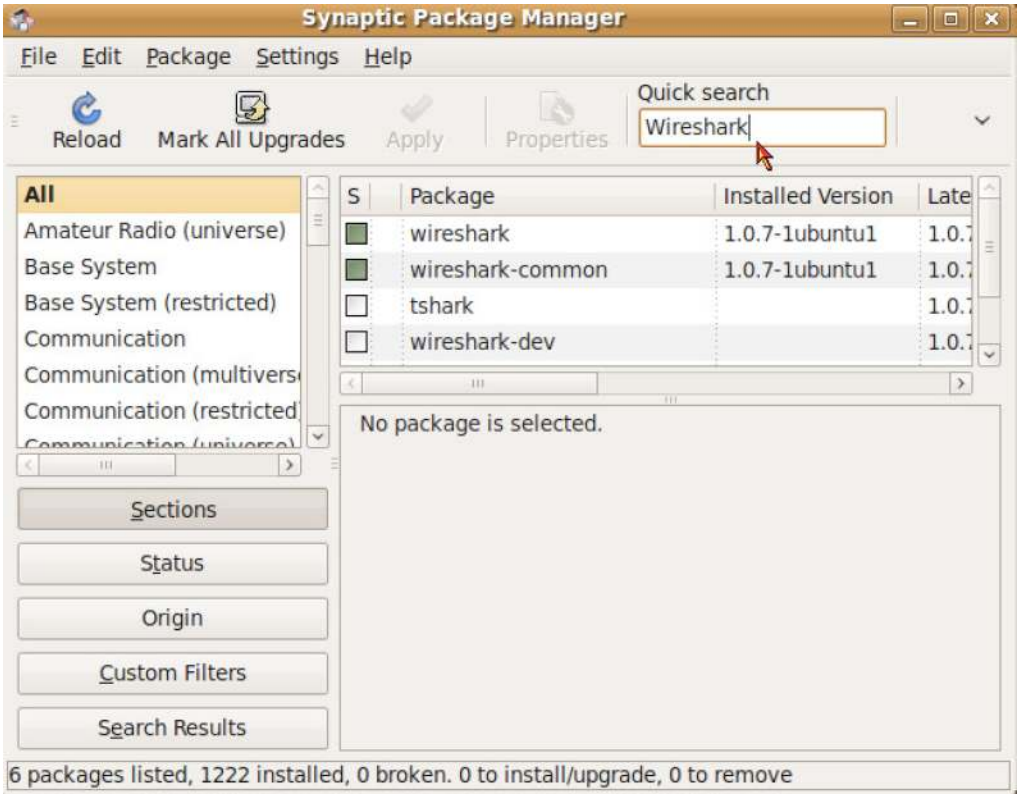


Wireshark

Wireshark Installation via Synaptic Package Manager

Follow the same process as outlined in Figures 53 to 62 to install the Wireshark (Figure 86) application. The below figure depicts the currently installed application—as denoted by the green box next to the application. You will note that the Wireshark application is dependent on "**wireshark-common**." The Synaptic Package Manager will identify the dependency and prompt you to also install "**wireshark-common**."

Figure 64.

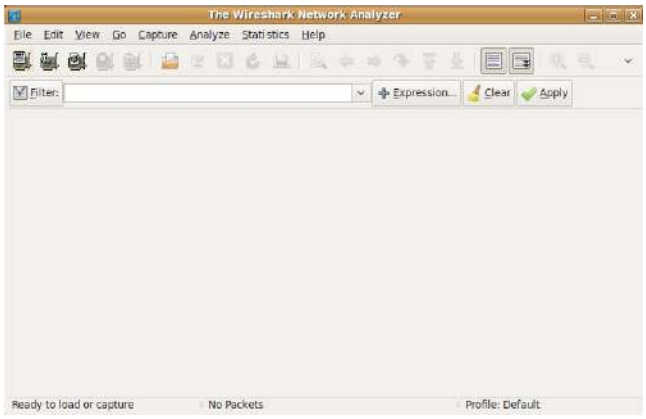


Launching Wireshark

Wireshark can be launched from "**Applications>Internet>Wireshark (as root)**" by a single click or you can add the launcher to the desktop by right clicking on wireshark and then select "**Add this launcher to desktop**." The Wireshark icon can be double clicked from the Desktop to launch the application. Launched application is depicted in Figure 88.

Figure 65.

Figure 66.



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Printed in USA C11-603836-00 06/10