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

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 ping, 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 ping
  - `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*:

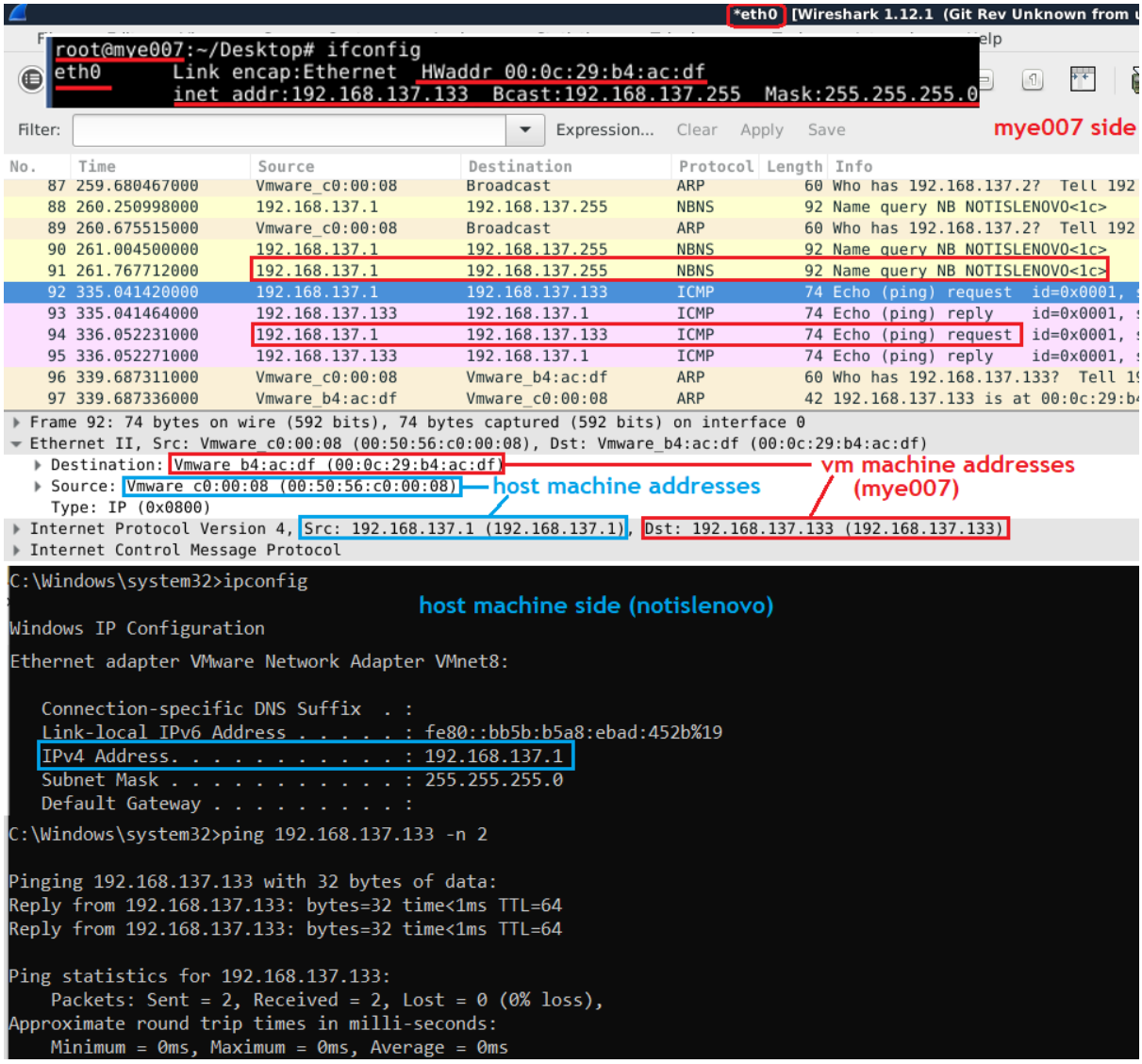
```
root@c2:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.57  netmask 255.255.255.0  broadcast 192.168.122.255
        inet6 fe80::216:3eff:fe9f:1f32  prefixlen 64  scopeid 0x20<link>
        ether 00:16:3e:9f:1f:32  txqueuelen 1000  (Ethernet)
        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 χρησιμοποιησω την εντολή *ping* σε συνδυασμό με το Wireshark (έχοντας ενεργοποιημένο το *promiscuous mode* σε όλα τα interfaces).

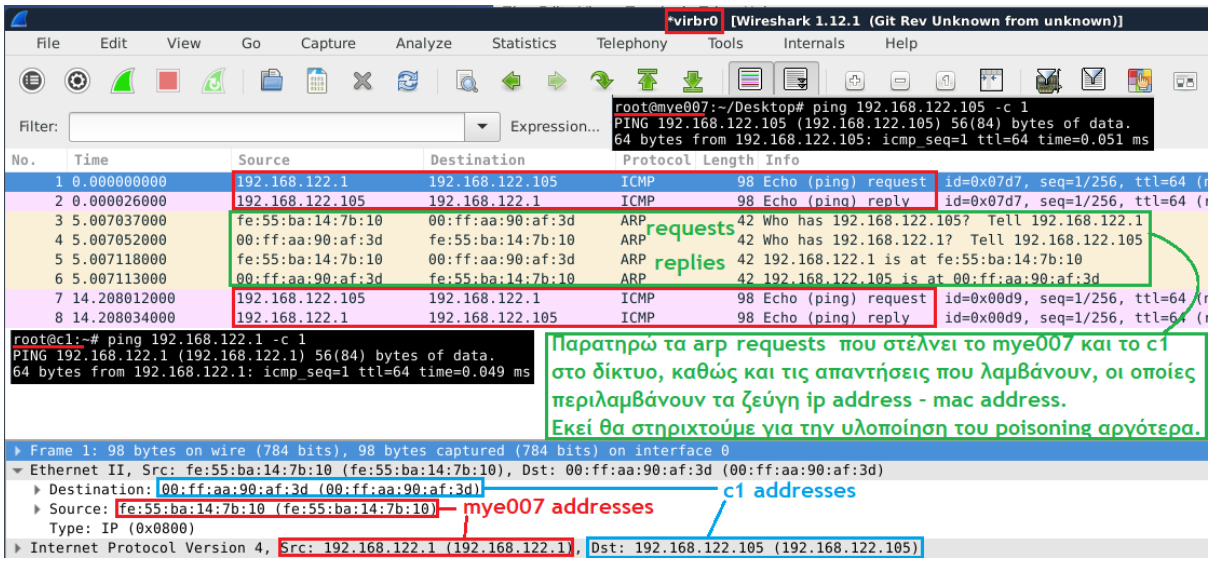
- `eth0`:
Χρησιμοποιώντας το Wireshark για να παρακολουθήσω την διεπαφή `eth0`, παρατηρώ την επικοινωνία του *mye007* με το host machine
όταν το host machine στέλνει δύο πακέτα ping requests στο *mye007*.



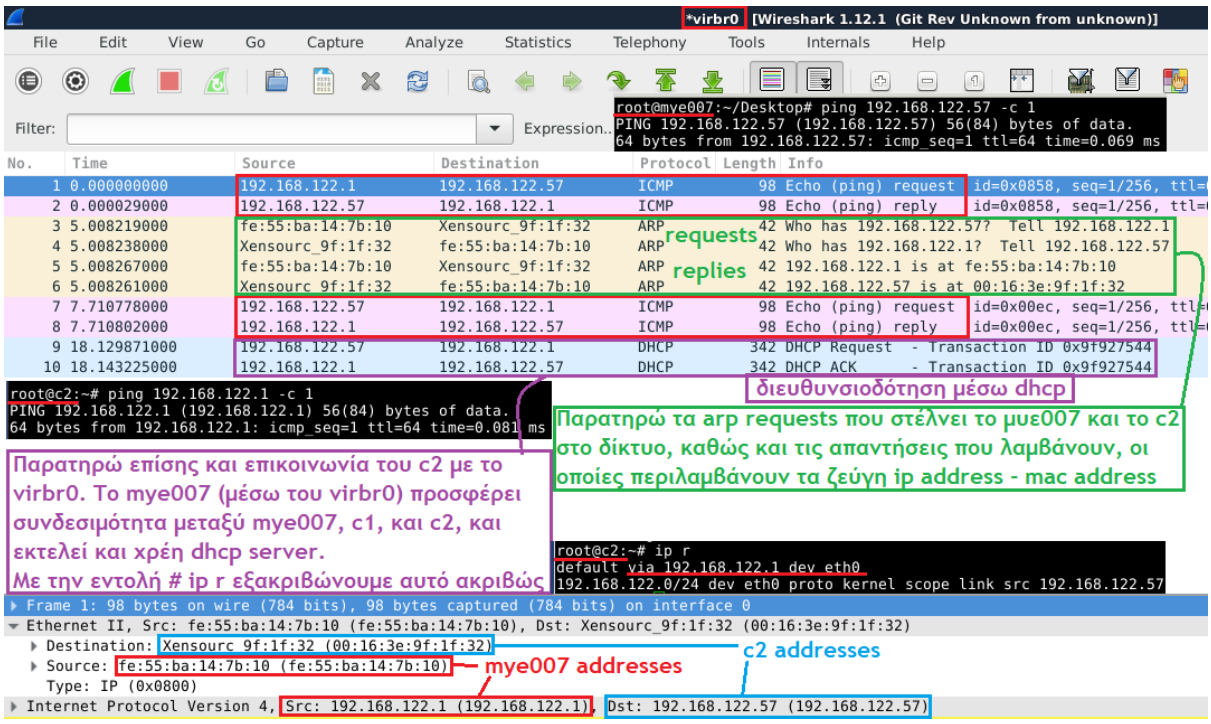
- `virbr0`:

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:

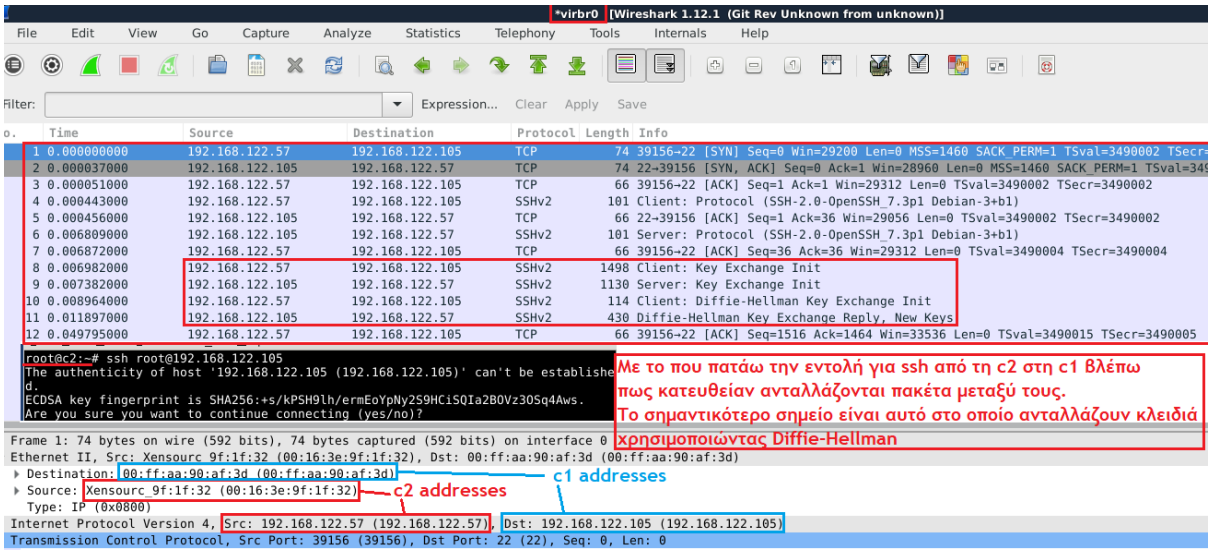


2. ping μεταξύ *mye007* – μηχανής *c2* και αντίστροφα:

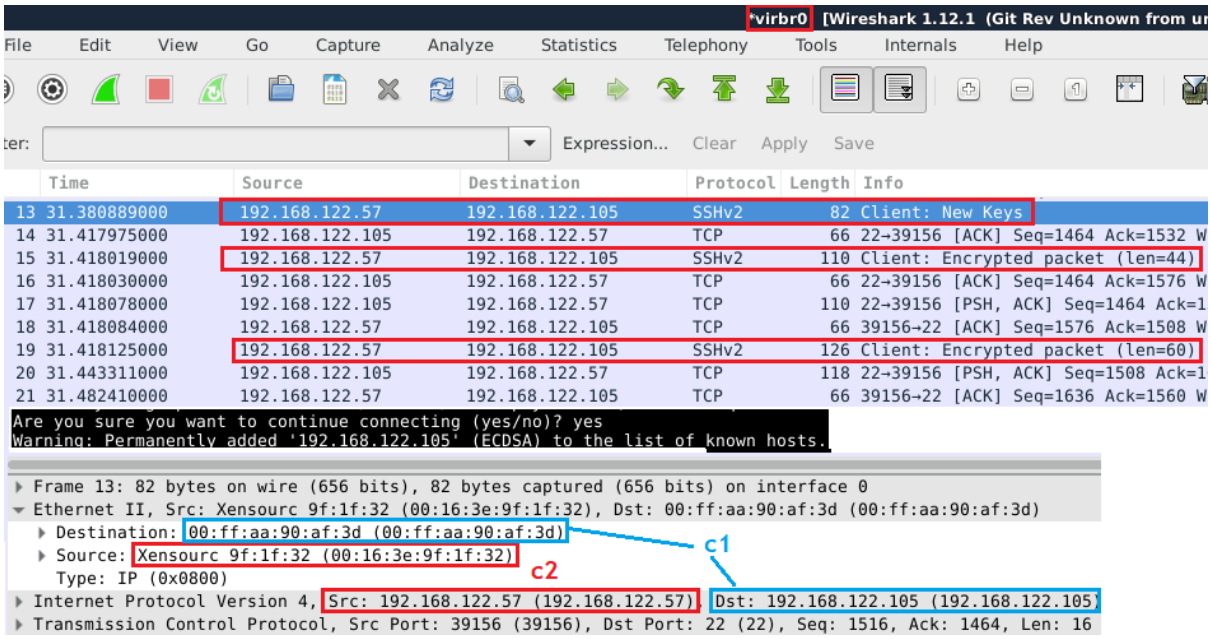


3. ssh από την μηχανή *c2* στη μηχανή *c1*:

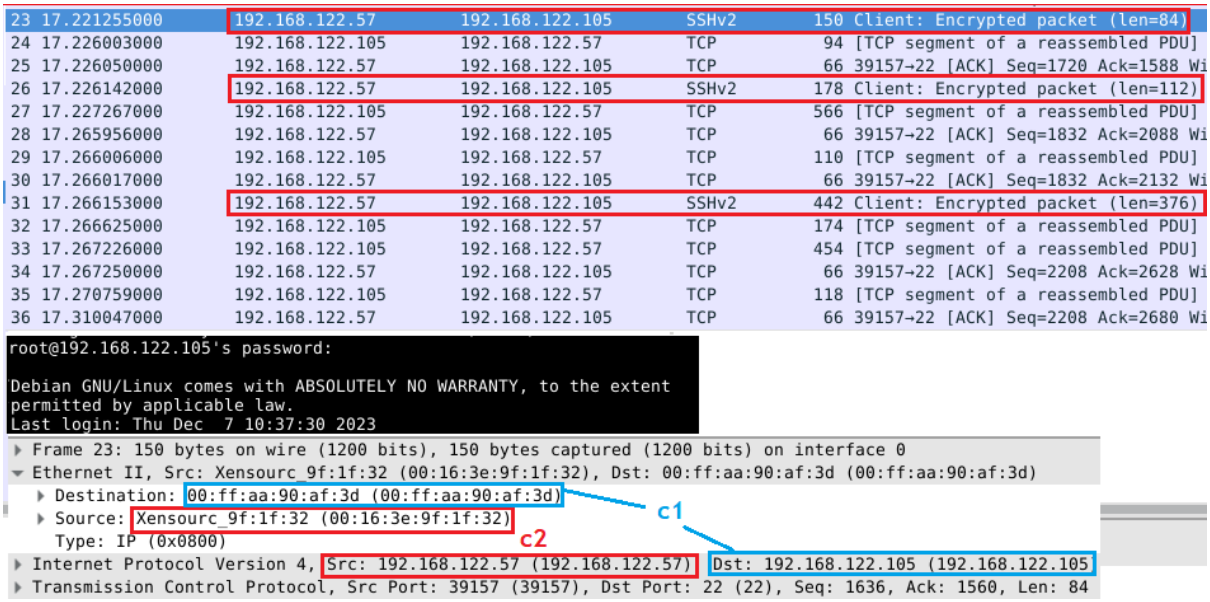
Πριν προσθέσει η *c2* την *c1* στη λίστα known hosts:



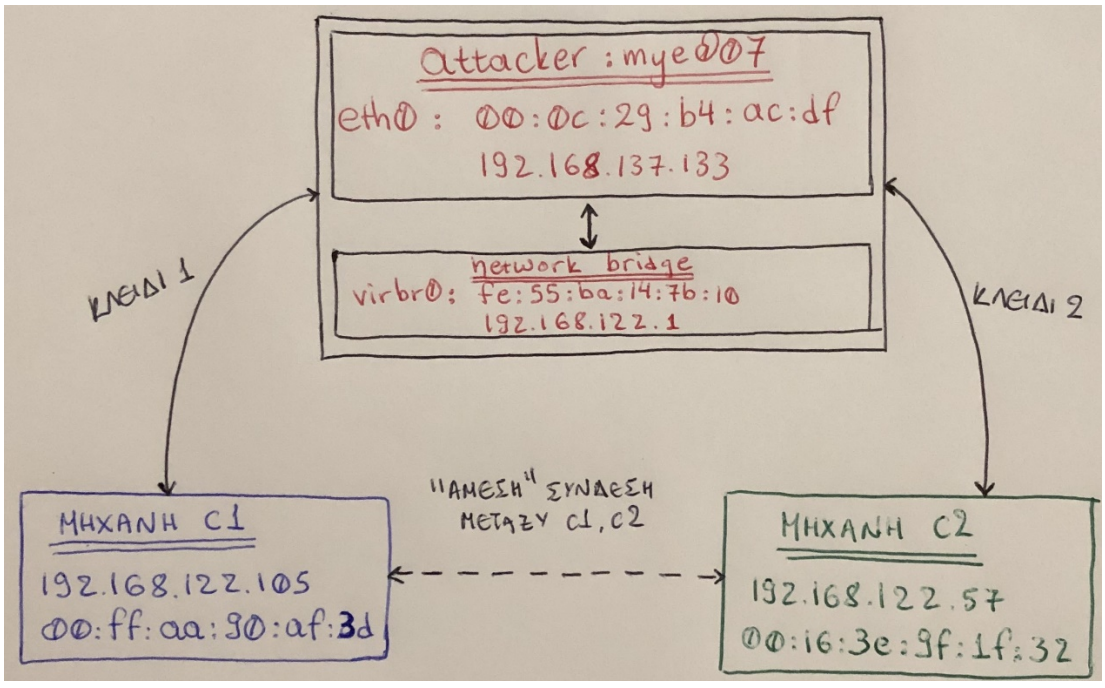
Αφού προσθέσει η *c2* την *c1* στη λίστα known hosts:



Αφού η *c2* πληκτρολογήσει τον κωδικό της *c1*:



Εν τέλει έχω επιβεβαιώσει την εξής συνδεσμολογία:

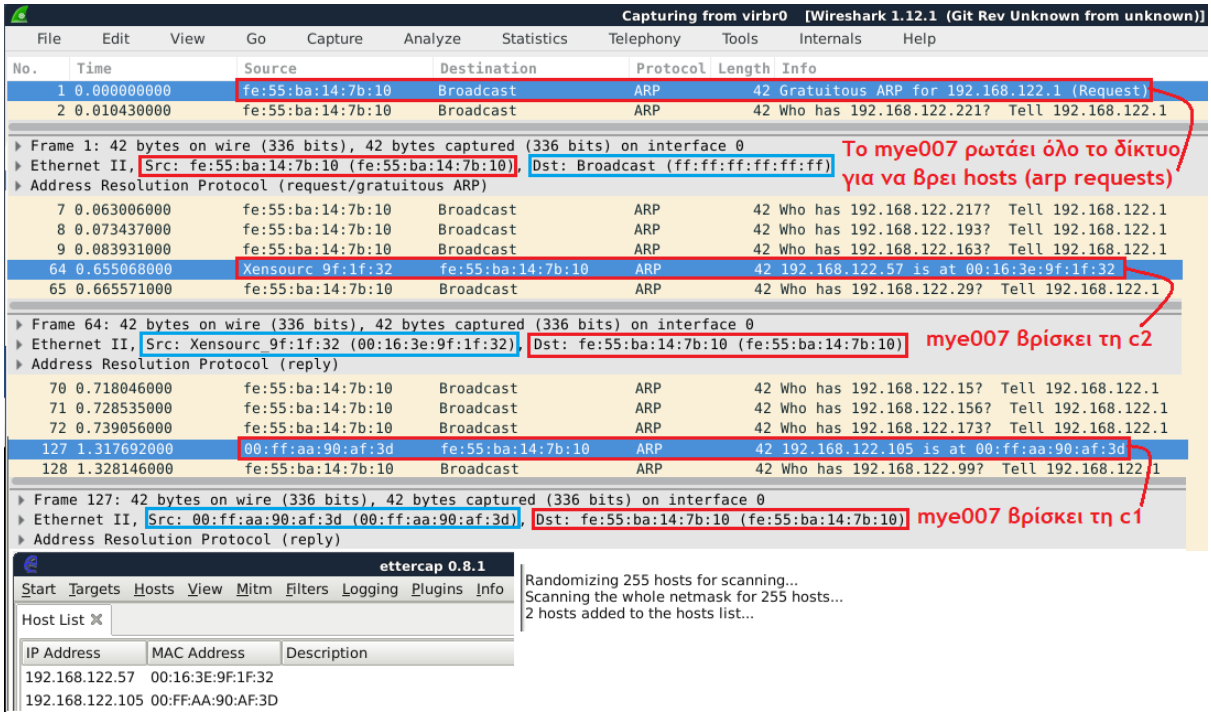


Όπως έχω γράψει και παραπάνω, οι μηχανές *c1* και *c2* μπορούν να δουν και τις δύο διευθύνσεις του επιτιθέμενου. Για τους σκοπούς της άσκησης εγώ αναφέρομαι στο *mye007* με τη διεύθυνσή του αυτή η οποία ανήκει στο ίδιο δίκτυο με τις *c1* και *c2*, δηλαδή τη `virbr0 - 192.168.122.1`.

Ettercap

Ακολουθώντας την εκφώνηση, εκκινώ το Ettercap:
- Κάνω unified sniffing πάνω από τη διεπαφή `virbr0`
- Scan for hosts

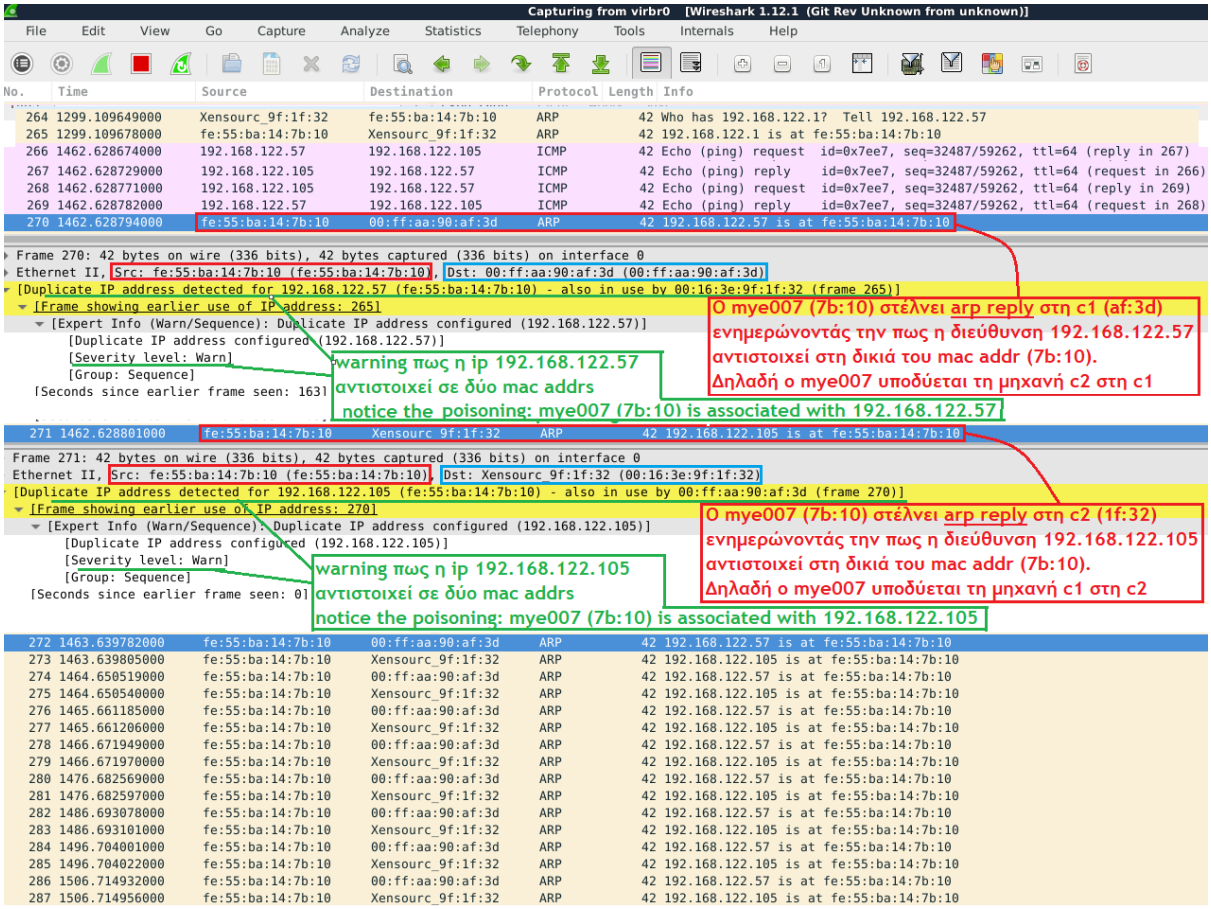
Παρατηρώ το παρακάτω traffic στο Wireshark:



- 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:



Παρατηρώ:

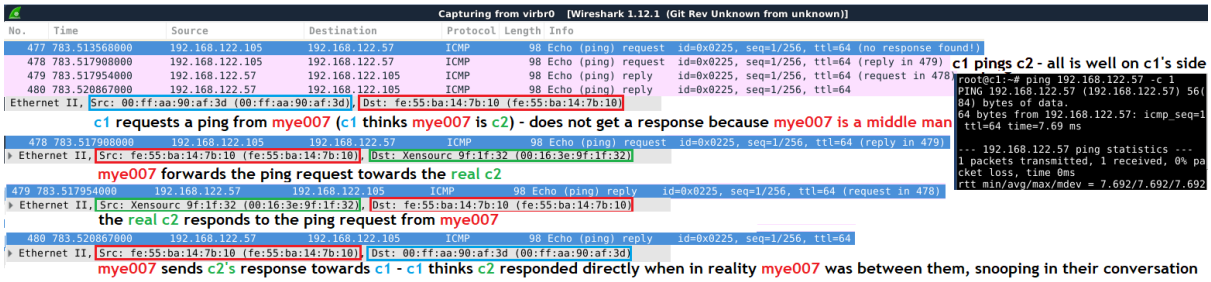
- Ο *mye007* "spammάρει" τις μηχανές *c1* και *c2* με arp replies κάνοντας announce τον εαυτό του ως *c2* και *c1* αντίστοιχα. Δηλαδή:

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 αφορα το assignment των ip addresses. Βέβαια τώρα γνωρίζω πως κάποιος προσπαθεί να υποδυθεί κάποιον άλλον.

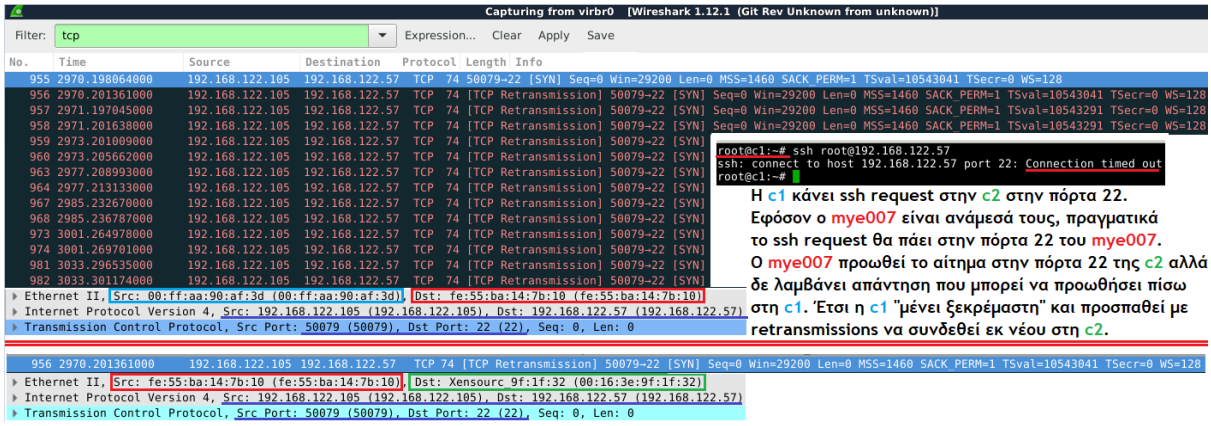
Θα επιχειρήσω τις εξής ενέργειες ενόσω κοιτάζω το traffic στο Wireshark:
- Να κάνω ping μεταξύ *c1* και *c2*
- Να κάνω ssh μεταξύ *c1* και *c2*

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



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`):



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.

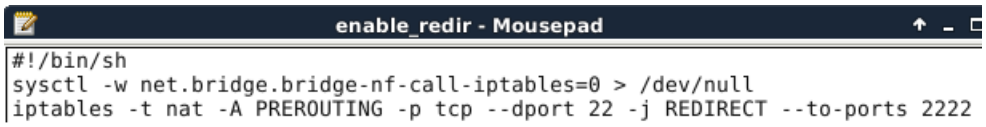
Συνεπώς, ο *mye007* προωθεί το original ssh request στην πόρτα 22 της *c2* αλλά δε λαμβάνει απάντηση που μπορεί να προωθήσει πίσω στη *c1*.

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”.

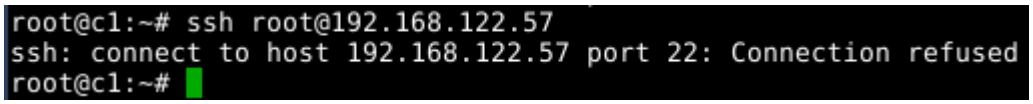
Finalizing our attack

enable_redir

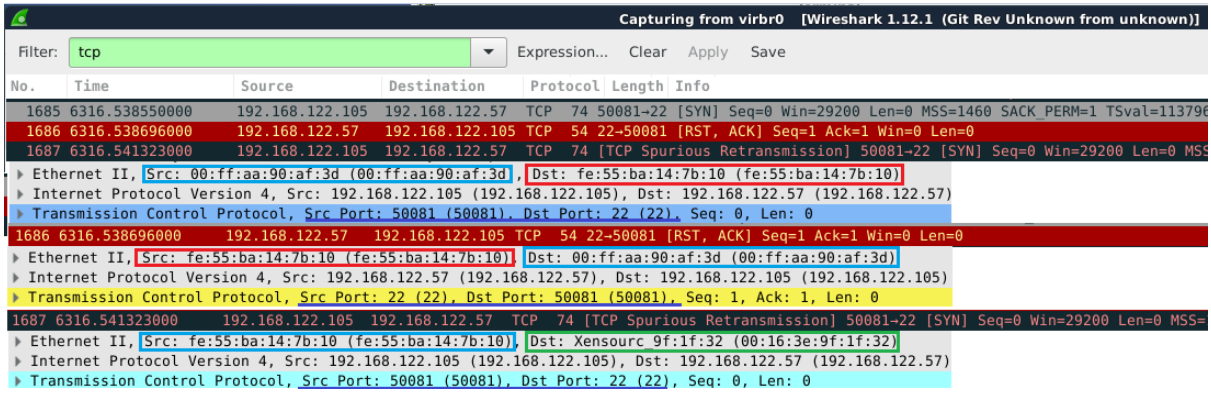
Ακολουθώντας την εκφώνηση, εκτελώ `#enable_redir` για να ανακατευθύνω το traffic που φτάνει στην πόρτα 22 του *mye007* στην πόρτα 2222 (εκεί όπου μετά θα βάλω το mitm-ssh να ακούει):



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



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.



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:jvlbd6ZRY9mhkkOwa0sjACQplxDI+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:

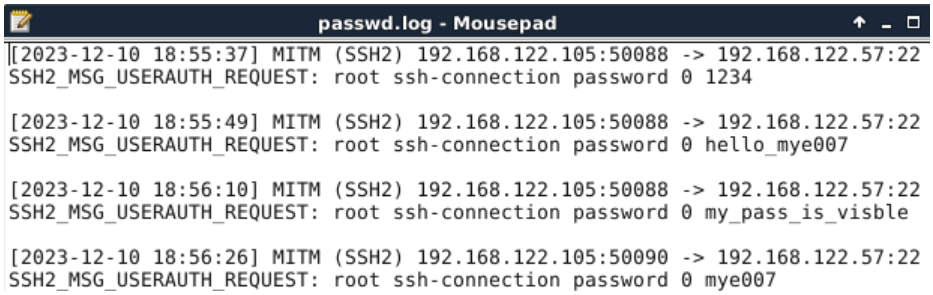
```
root@c1:~# ssh root@192.168.122.57
The authenticity of host '192.168.122.57 (192.168.122.57)' can't be established.
RSA key fingerprint is SHA256:jvlbd6ZRY9mhkkOwa0sjACQplxDI+VrzzqnPIGwUejk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.122.57' (RSA) 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:

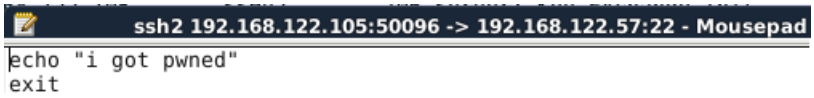
```
root@mye007:~/Desktop# mitm-ssh -v -n -p 2222
Using static route to 255.255.255.255:22
SSH MITM Server listening on 0.0.0.0 port 2222.
WARNING: /usr/local/etc/moduli does not exist, using fixed modulus
[MITM] Found real target 192.168.122.57:22 for NAT host 192.168.122.105:50088
[MITM] Routing SSH2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



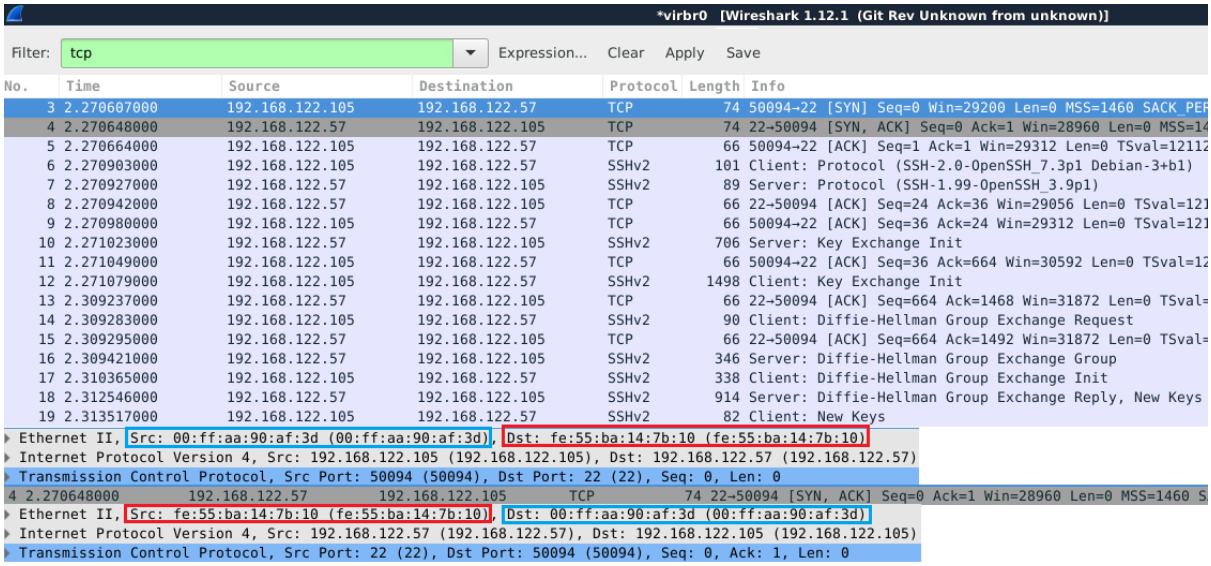
- οτιδήποτε στέλνει η *c1* στη *c2*



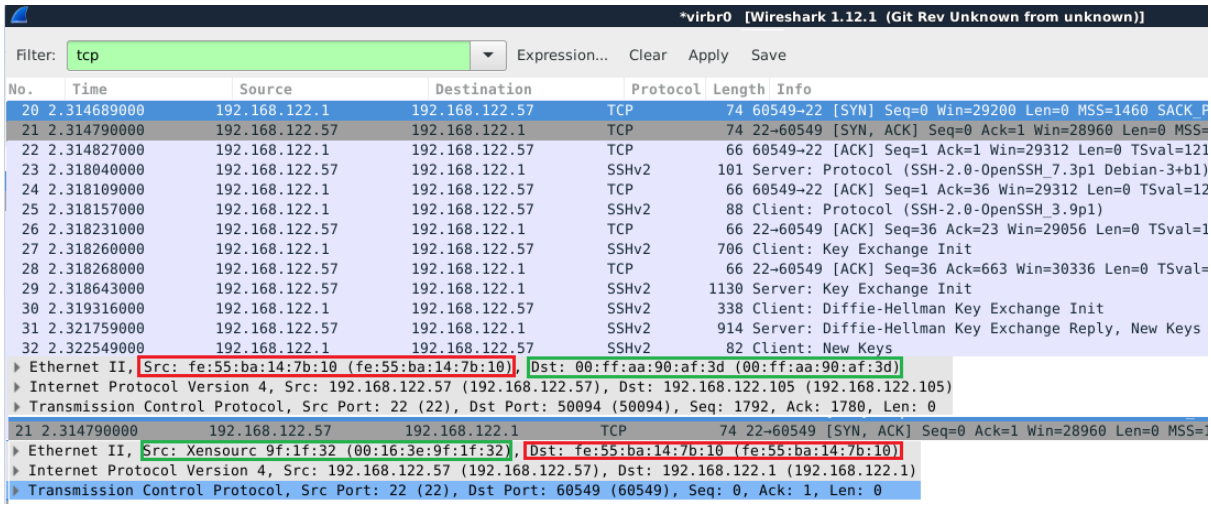
- οτιδήποτε στέλνει η *c2* στη *c1*

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

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



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



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.