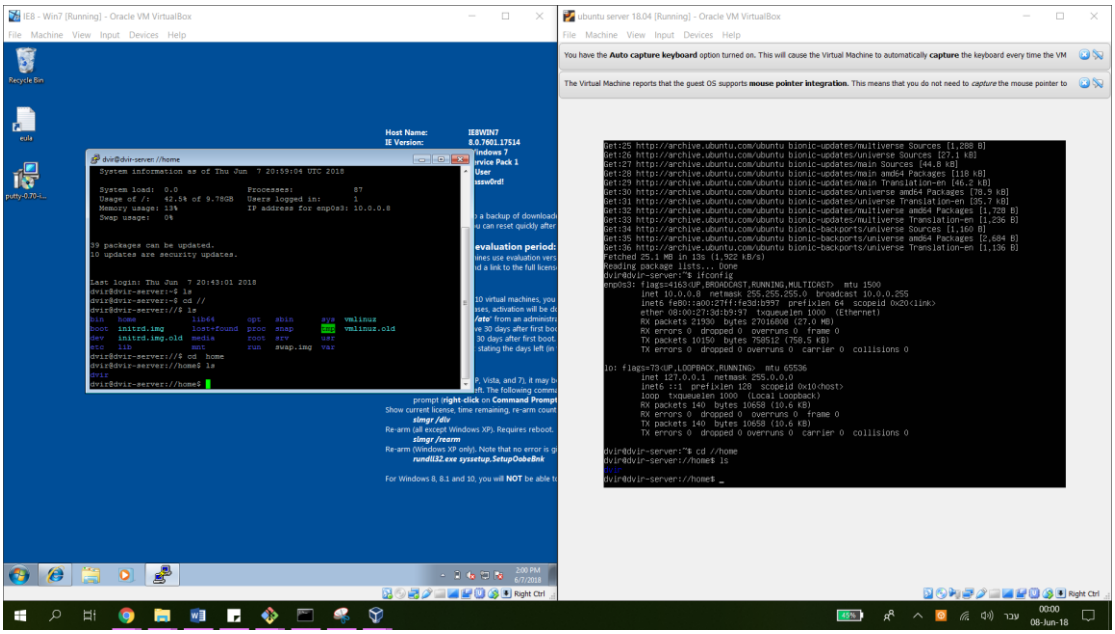
דביר ברזילי

פתרון למטלה 4 – מעבדת סייבר הגנה

Phase 1

לאחר התקנת Virtual Box עם מכונה וירטואלית של Ubuntu Server הגדרתי את הגדרות הרשת של המכונה להיות על מצב Bridged והתחברתי דרך Putty ב - Windows למכונה וירטואלית זו. בתמונה הבאה ניתן לראות שיצרתי תקייה בשם Dvir על המכונה ע"י חיבור דרך פורט 22.

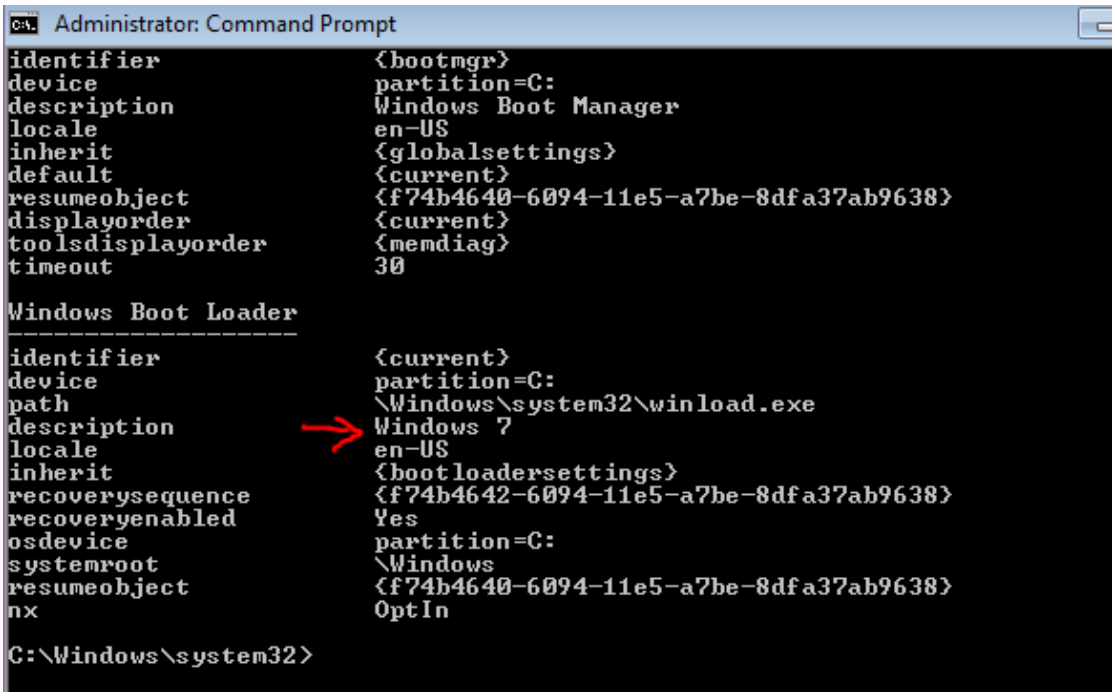


דרכים נוספות לשוטט בקבצים של מחשב אחר הן דרך הפרוטוקולים FTP ו-Telnet.

נבחן את ההבדלים בין הגדרות הרשת השונות במכונה וירטואלית:

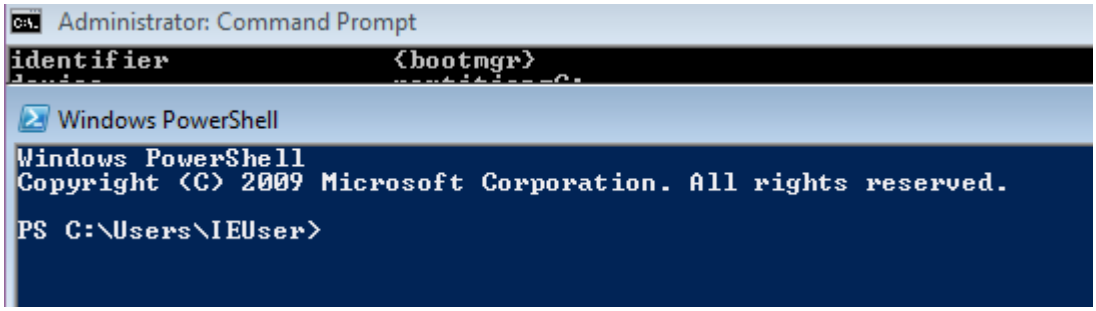
- Host Only – משמעות הגדרה זו היא שהמכונה תקבל כתובת IP אך הכתובת תהיה נגישה רק למכונות שרצות על ה- VM. מחשבים אחרים לא יוכלו לתקשר עם המכונה שמוגדרת במצב זה.
- Bridged – במצב זה המכונה תתפקד כמחשב נוסף בתוך הרשת שלנו אך תהיה נגישה בכתובת הIP שלה לכולם.
- NAT – למכונה הוירטואלית תהיה כתובת IP משלה כמחשב נוסף בתוך הרשת ותוכל לגשת לאינטרנט. אולם מחוץ לרשת לא נוכל לגשת ישירות לכתובת הIP של המחשב.

Phase 2



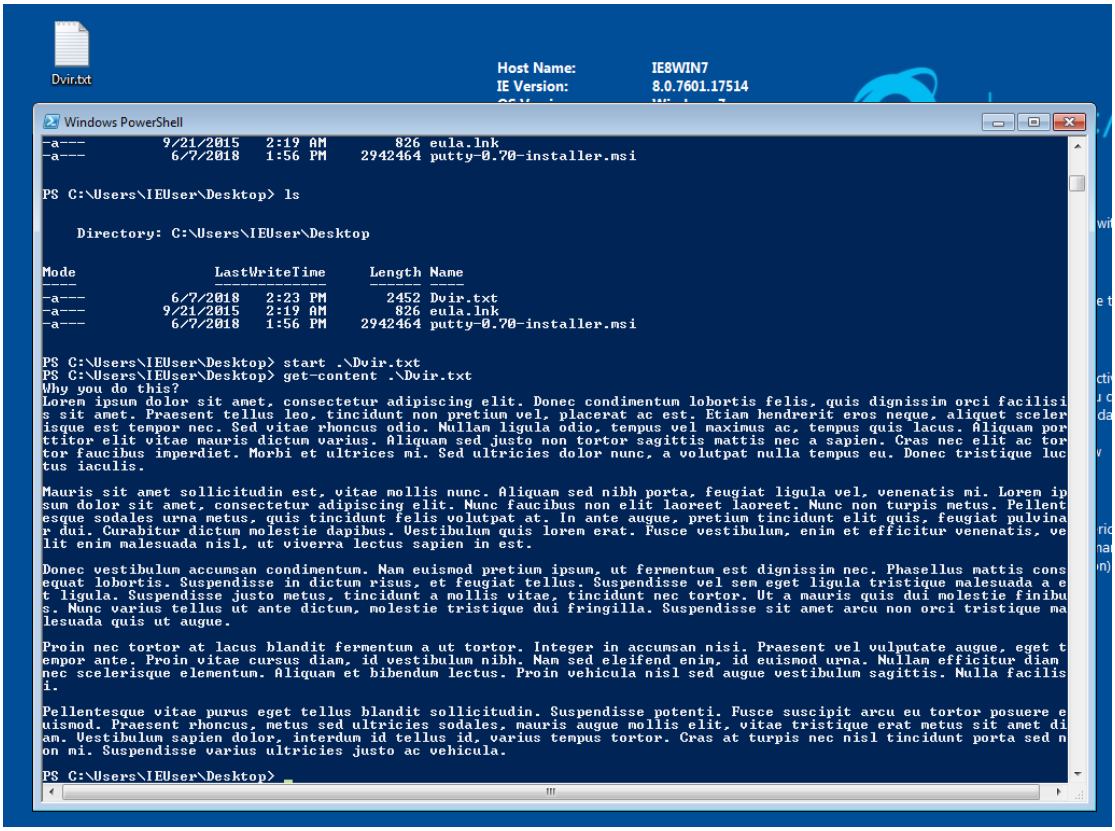
לפי התיאור ניתן לראות שהשם והגרסה של הboot manager הוא Windows 7.

כפי שניתן לראות בתמונה, אני השתמשתי ב-Command Prompt Shell. אולם קיימים Shells נוספים.
לדוגמא – Windows PowerShell:



דביר ברזילי

יצרתי קובץ Dvir.txt שמכילים חמש פסקאות שהעתקתי מהאתר Lorem ipsum וקראתי את התוכן שלו דרך ה-PowerShell בעזרת הפקודה get-content .\Dvir.txt



<u>תיעוד של תהליך המחיקה של ה-kernel הנוכחי והתקנת kernel חדש:</u>

עדכנתי את כל המאגרים של המערכת:

sudo apt update

ושדרגתי את כל החבילות:

sudo apt upgrade –y

לאחר מכן נעשה ריבוט למערכת כדי לוודא שכל השינויים יכנסו לתוקפם בעזרת הפקודה

sudo reboot

וידאתי שלא קיימים עדכונים נוספים שעליי לעשות בעזרת הפקודה

sudo apt list –upgradeable

ומכאן המשכתי לשדרוג ה-kernel שלי לאחד חדש.

בתמונה הבאה ניתן לראות מהי הגרסה של ה-kernel הנוכחי:

```
root@dvir-server:/home/dvir# uname -msr
Linux 4.15.0-22-generic x86_64
```

אחרי זה נכנסתי לאתר http://kernel.ubuntu.com/~kernel-ppa/mainline/ ונכנסתי לגרסה העדכנית ביותר שיש – 4.17 והורדתי את הקבצים הנדרשים בעזרת הפקודות:

דביר ברזילי

wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.17/linux-headers-4.17.0-041700_4.17.0-041700.201806041953_all.deb

wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.17/linux-headers-4.17.0-041700-generic_4.17.0-041700.201806041953_amd64.deb

wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.17/linux-image-unsigned-4.17.0-041700-generic_4.17.0-041700.201806041953_amd64.deb

wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.17/linux-modules-4.17.0-041700-generic_4.17.0-041700.201806041953_amd64.deb

```
root@dvir-server:/home/dvir# ls
linux-headers-4.17.0-041700_4.17.0-041700.201806041953_all.deb
linux-headers-4.17.0-041700-generic_4.17.0-041700.201806041953_amd64.deb
linux-image-unsigned-4.17.0-041700-generic_4.17.0-041700.201806041953_amd64.deb
linux-modules-4.17.0-041700-generic_4.17.0-041700.201806041953_amd64.deb
root@dvir-server:/home/dvir#
```

כעת, בעזרת הפקודה *dpkg -i *.deb אנו נתקין את ארבעת החבילות שהורדנו.*

בסיום ההתקנה נריץ את הפקודות:

sudo update-grub
sudo reboot

אחרי כל זה, ניתן לראות שעדכנו את ה-kernel שלנו לגרסה החדשה ביותר 4.17.

```
dvir@dvir-server:~$ uname -msr
Linux 4.17.0-041700-generic x86_64
```

<u>לאחר שעברנו לגרסה העדכנית ביותר אנו יכולים למחוק את הגרסה הישנה שבה השתמשנו לפני העדכון, גרסה 4.15:</u>

ע"מ להסיר את הגרסה הישנה נשתמש בשורת הפקודה של **byobu** , שבעזרתה נוכל לבצע את התהליך בצורה בטוחה בלי למחוק בטעותאת הגרסה שבה אנחנו משתמשים כרגע.

נתקין את byobu:

sudo apt install byobu

אחרי שאנו מסיימים להתקין, נבדוק איזה גרסאות ישנות יש לנו על המחשב בעזרת הפקודה

dpkg -l | grep linux-image

```
root@dvir-server:/home/dvir# dpkg -l | grep linux-image
ii  linux-image-4.15.0-20-generic              4.15.0-20.21
  amd64        Signed kernel image generic
ii  linux-image-4.15.0-22-generic              4.15.0-22.24
  amd64        Signed kernel image generic
ii  linux-image-generic                        4.15.0.22.23
  amd64        Generic Linux kernel image
ii  linux-image-unsigned-4.17.0-041700-generic 4.17.0-041700.201806041953
  amd64        Linux kernel image for version 4.17.0 on 64 bit x86 SMP
```

בתמונה זו נוכל לראות שיש גם את הגרסה הישנה שהייתה לנו 4.15, וגם את הגרסה החדשה שהורדנו 4.17.

אם נריץ את הפקודה

דביר ברזילי

sudo purge-old-kernels

אנו נמחק את כל הגרסאות הישנות שיש ברשותנו, מלבד ה-2 האחרונות (כך מוגדר להיות כברירת מחדל, כדי להיות בטוחים שאם הkernel החדש נכשל אזי יהיו לנו 2 גרסאות נוספות לחזור אליהן).

--

בחלק זה נעזרתי במדריך המפורט: https://www.howtoforge.com/tutorial/how-to-upgrade-linux-kernel-in-ubuntu-1604-server/

Phase 3

יצרתי משתמש חדש בשם ariel:

```
root@dvir-server://home# ls
ariel  dvir
```

כמו כן, יצרתי משתמש נוסף בשם root-1 באופן ידני ע"י עריכת הקובץ passwd:

```
root-1:x:1002:1006:root-1,root-1,,:/home/root-1:/bin/bash
```

ולאחר מכן הוספתי סיסמה בעזרת הפקודה passwd root-1 , ניתן לראות בתמונה הבאה שהתווספה שורה עם הסיסמה המוצפנת בסוף הקובץ shadow:

```
dvir:$6$c9OmBILJRQ/kggY9$jTXDiZ0yxYEYZy9HSgUWW8pDLAfgeJvJTcN3ea80/IZ04QR1peSLdCfe1TXlhV4jLzEW60yFr171F454y.EDK0:17680:0:99999:7:::
ariel:$6$BwedWlEu$hdXXBQqHFdQzQUqDZxrUiF3yLDuzKHChEVzkm.EFZExQ4KeJ.vu.ywCQ21enJ.C9gasPOJnuKmRGbel510Vwj/:17690:0:99999:7:::
root-1:$6$MXMaE5xl$KeNaDwn1SbnBSHjmKRBJl3vRrT0unEb74qkDQ9NIvs3/LyyYBJq0T53jDPw9wRSqt/EONIlzwTxVsqXd.q3rP0:17690::::::
root@dvir-server://etc#
```

הרשאת sudo למשתמש ariel:

נריץ בהרשאת ה-root הנוכחית את הפקודה sudo visudo ונערוך את הקובץ תחת הכותרת User privilege specification כך שגם למשתמש ariel יהיו הרשאות:

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
ariel    ALL=(ALL:ALL) ALL
```

ואחרי ששמרנו אנו יכולים להתחבר למשתמש ariel ולהשתמש בsudo:

```
ariel@dvir-server:~$ sudo apt-get update
[sudo] password for ariel:
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [83.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [119 k
B]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [7
9.8 kB]
Fetched 440 kB in 2s (213 kB/s)
Reading package lists... 19%
```

כדי לאפשר למשתמש root-1 להשתמש בפקודות sudo ללא צורך בהזנת סיסמה עלינו להזין שוב את הפקודה sudo visudo ולהוסיף את השורה הבאה (נעשה את זה מהמשתמש ariel, בשביל הקטע):

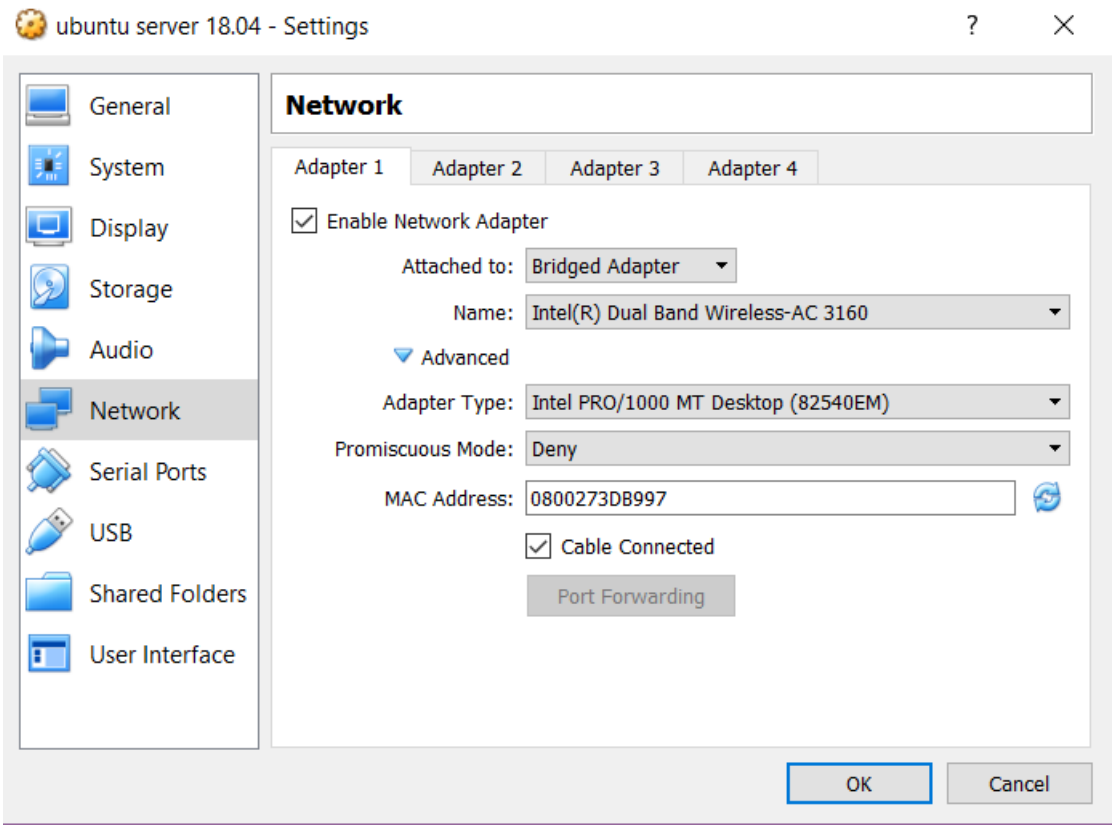
```
# User privilege specification
root    ALL=(ALL:ALL) ALL
ariel    ALL=(ALL:ALL) ALL
root-1    ALL=(ALL) NOPASSWD: ALL
```

לאחר שמירת השינויים, נוכל להתחבר למשתמש root-1 ולהשתמש בפקודות sudo ללא צורך בהזנת סיסמה:

```
root-1@dvir-server:/$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease
Fetched 83.2 kB in 1s (102 kB/s)
```

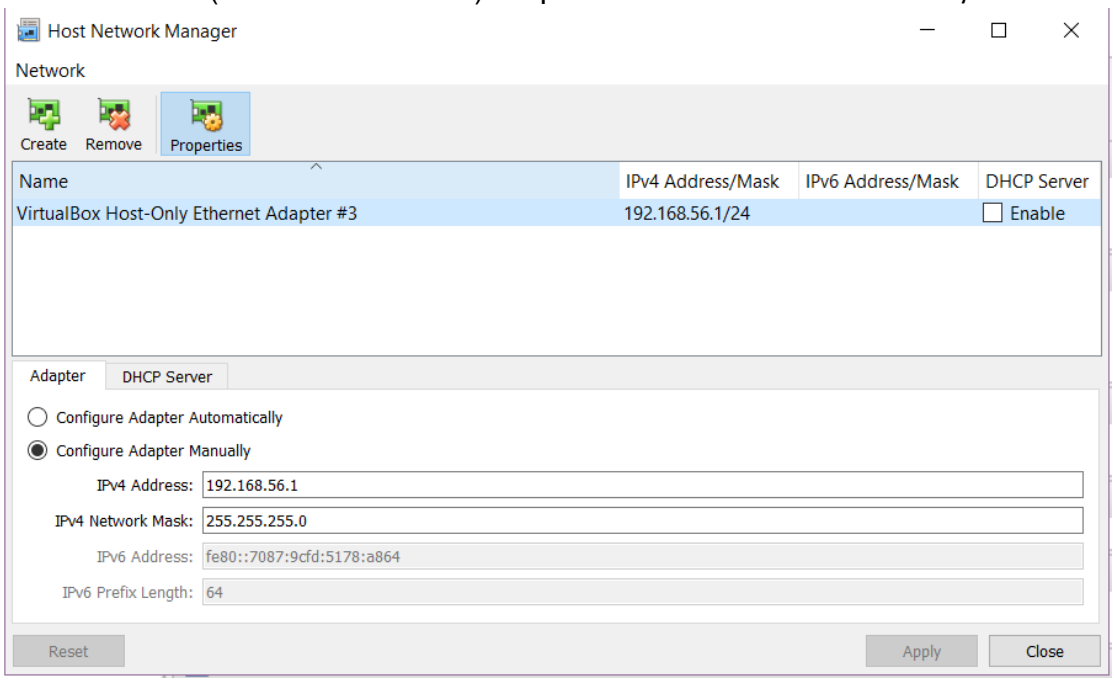
Phase 4

ניתן לשנות את כתובת הMAC כפי שניתן לראות בתמונה:

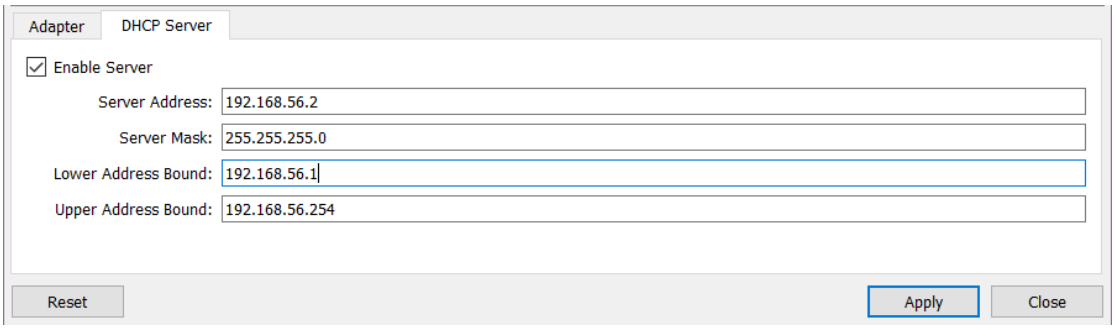


לאחר השינוי קיבלתי כתובת MAC חדשה: B3A88F080027

ובמצב רשת Host-Only נשנה את הכתובת IP באופן ידני (ה-DHCP במצב Enable):



נאפשר לDHCP להעניק למכונות כתובת IP בתחום קבוע, וכך נוכל ההכתובת תוכל להשתנות מפעם לפעם שאנו מתחברים לרשת:



כמו כן נוכל להוסיף ממשקים נוספים לרשת כך שלכל ממשק תהיה כתובת IP משלו וכך לאותה מכונה יהיו כמה כתובות IP.