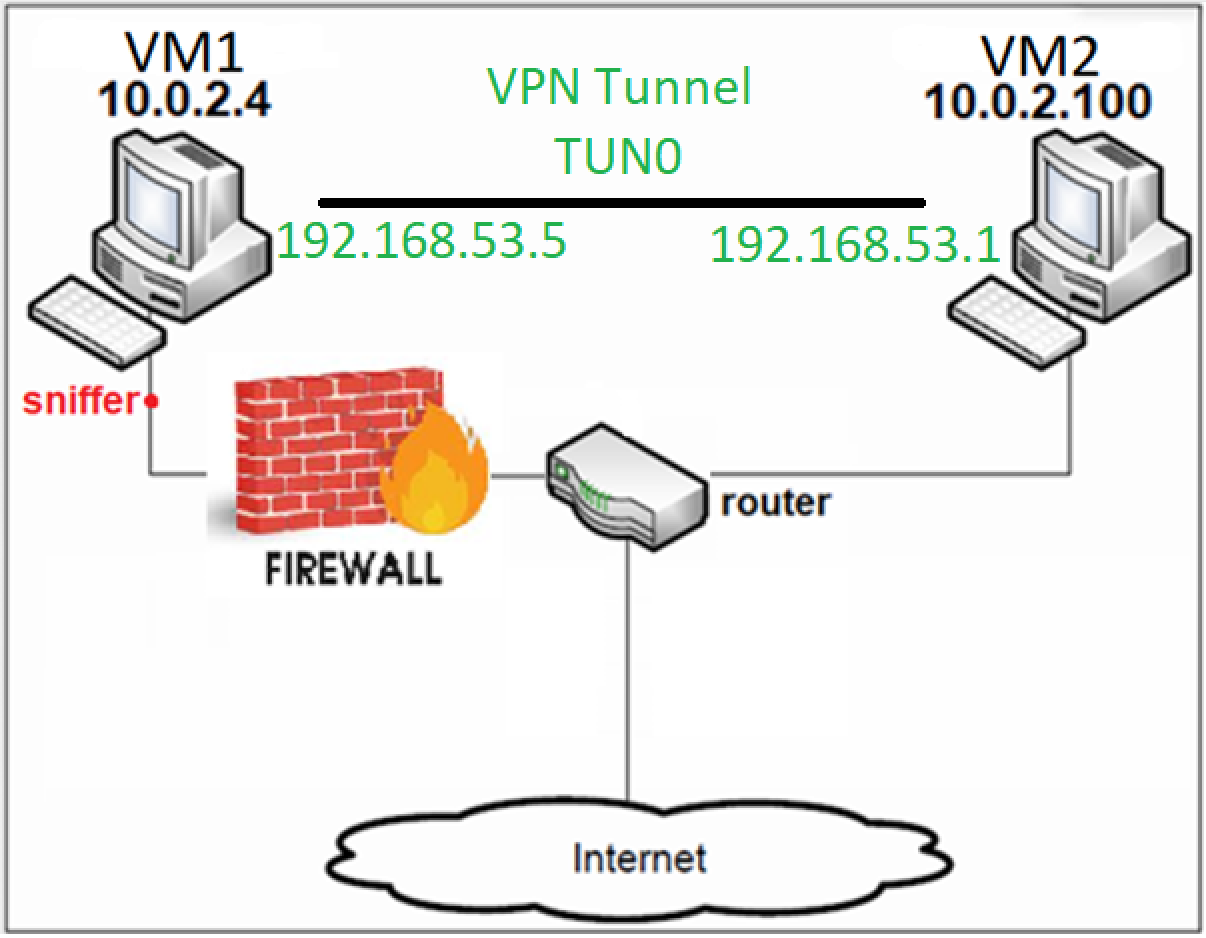
# **A06 – Linux Firewall Evasion Lab**

## כתובות IP לכל מחשב

|  |  |  |
| --- | --- | --- |
| Name | IP | TUN0 |
| VM1 | 10.0.2.4 | 192.168.53.5 |
| VM2 | 10.0.2.100 | 192.168.53.1 |

## Firewall

התמונה מטה מייצגת את מבנה הרשת



Task 1: VM Setup

### מבוא:

#### תיאור

הגדרת המכונות הוירטואליות ברשת NAT NETWORK

#### מטרה

ליצור רשת המכילה 2 מכונות וירטואליות שעל אחת מהן ירוץ הfirewall

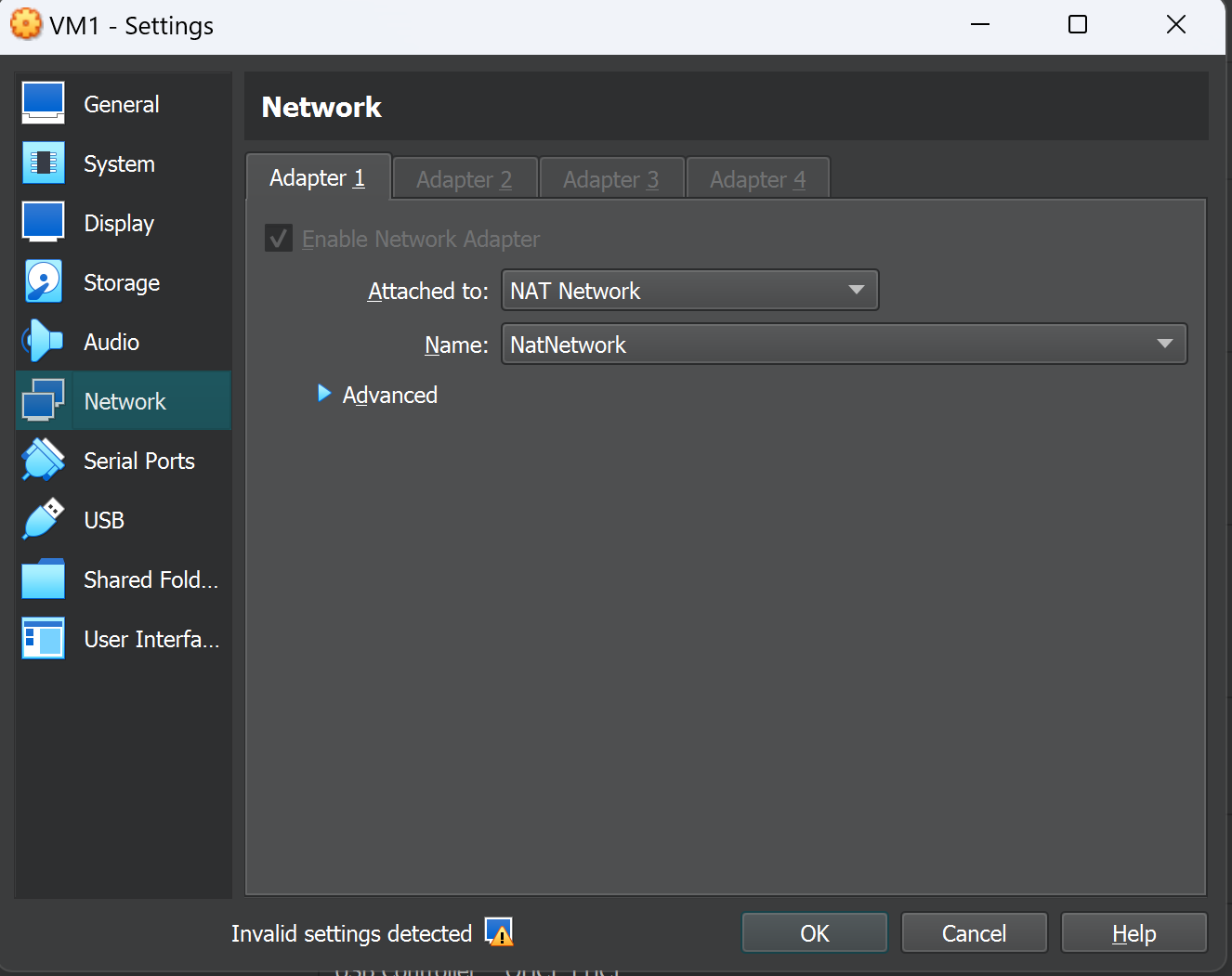
#### תוצאה מצופה

2 מכונות וירטואליות עובדות

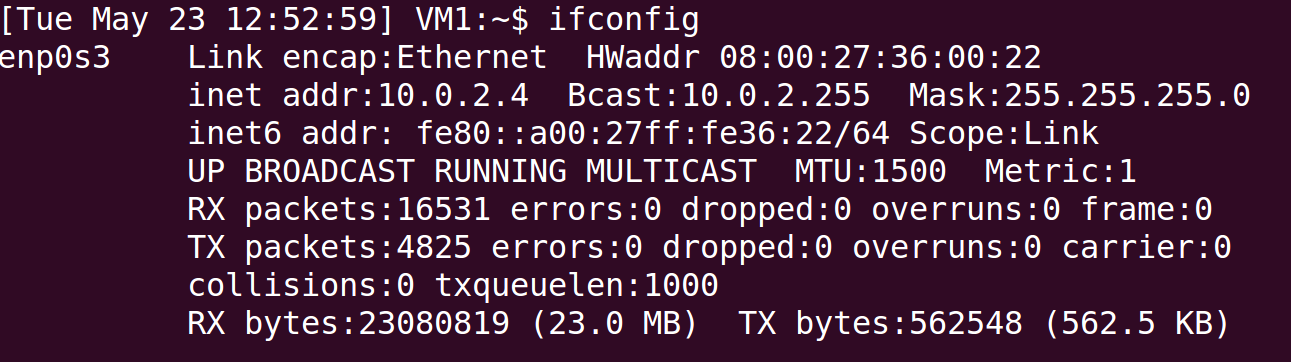
#### ביצוע המשימה

נראה שיש לנו 2 מכונות וירטואליות VM1 ו VM2.

כעת נראה את VM1:



ניתן לראות שיצרנו מכונה וירטואלית המחוברת לאינטרנט.



כעת נראה את VM2:

תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

ניתן לראות שיצרנו מכונה וירטואלית המחוברת לאינטרנט.

#### סיכום המשימה

הצלחנו לבצע את המשימה, ניתן לראות שיצרנו 2 מכונות וירטואליות המחוברות לאינטרנט עם כתובות IP שונות.

התוצאות התאימו למצופה ולא נתקלנו בבעיות במהלך ביצוע המשימה.

Task 2: Set up Firewall

### מבוא:

#### תיאור

הפעלת חומת אש, וקינפוג חוקים חדשים המונעים גישה לאתר sdarot.tw

#### מטרה

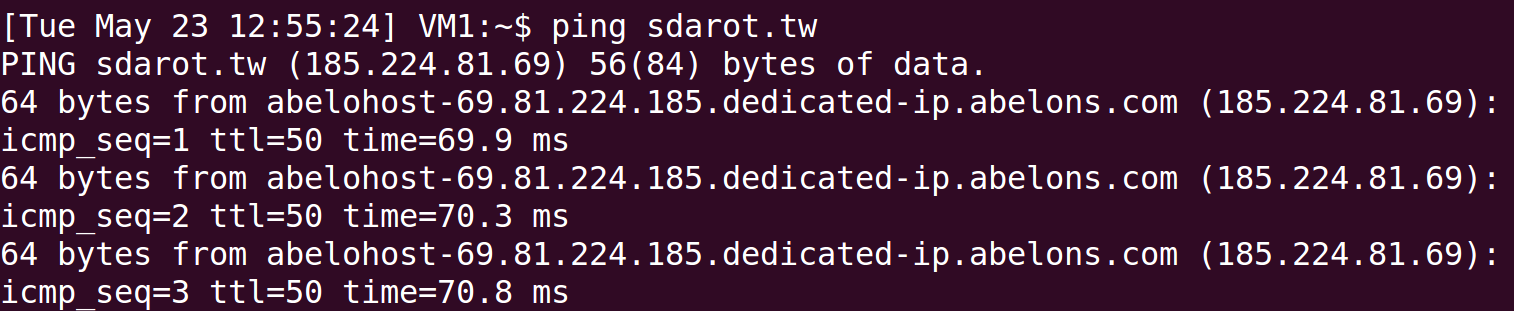
לחסום גישה מVM1 לשליחת פאקטות לשרתים של sdarot

#### תוצאה מצופה

כישלון בעת ניסיון גישה לשרתים של sdarot

#### ביצוע המשימה

תחילה נרצה לבדוק שאנחנו אכן מצליחים לגשת לאתר סדרות באמצעות פקודת PING לדומיין של סדרות:



ניתן לראות כי ניתן לגשת לאתר והping עבר בהצלחה.

כעת נרצה לבדוק את כל כתובות הדומיין הקשורים לאתר sdarot:

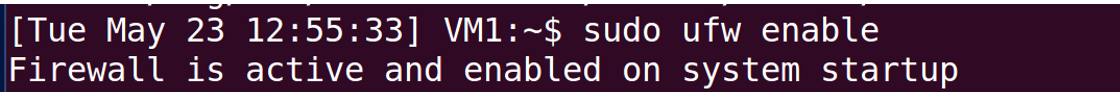
תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

נראה כי לסדרות יש 3 כתובות IP שנצטרך לחסום.

לאחר מכן, נשתמש בufw (נותן גישה פשוטה וידידותית יותר למשתמש להגדיר חוקים בחומת האש):

תחילה נפעיל את חומת האש



נראה שחומת האש הופעלה.

ואז נוסיף את החוקים שחוסמים את כתובות הIP

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

ניתן לראות שהחוקים נוספו בהצלחה.

ולבסוף נבדוק את טבלת חומת האש המעודכנת:

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

ניתן לראות שחומת האש מופעלת עם החוקים שהוגדרו לפני כן בהצלחה.

זה עתה, נרצה לבדוק אם חומת האש חוסמת את הגישה אל sdarot:

תמונה שמכילה טקסט, גופן, צילום מסך

התיאור נוצר באופן אוטומטי

ניתן לראות כי החסימה בוצעה בהצלחה ולא ניתן לגשת אל השרתים של sdarot.

#### סיכום המשימה

הצלחנו לבצע את המשימה.

ניתן לראות שהצלחנו ליצור חוקים אשר חוסמים גישה אל כתובות הIP של sdarot ולאחר יצירת החוקים לא היה ניתן לגשת ולבצע שליחת ping.

התוצאות התאימו למצופה מאחר והצלחנו לבצע חסימת גישה לשרתי sdarot ולא נתקלנו בבעיות במהלך ביצוע המשימה.

Task 3: Bypassing Firewall using VPN

### מבוא:

#### תיאור

יצירת חיבור VPN לצורך מעקף חוקי חומת האש.

#### מטרה

נרצה ליצור חיבור VPN בין הלקוח לשרת ולהעביר את הpackets דרך החיבור הנ"ל וע"י כך לעקוף את החסימה המוגדרת בחומת האש לשרתים ספציפיים.

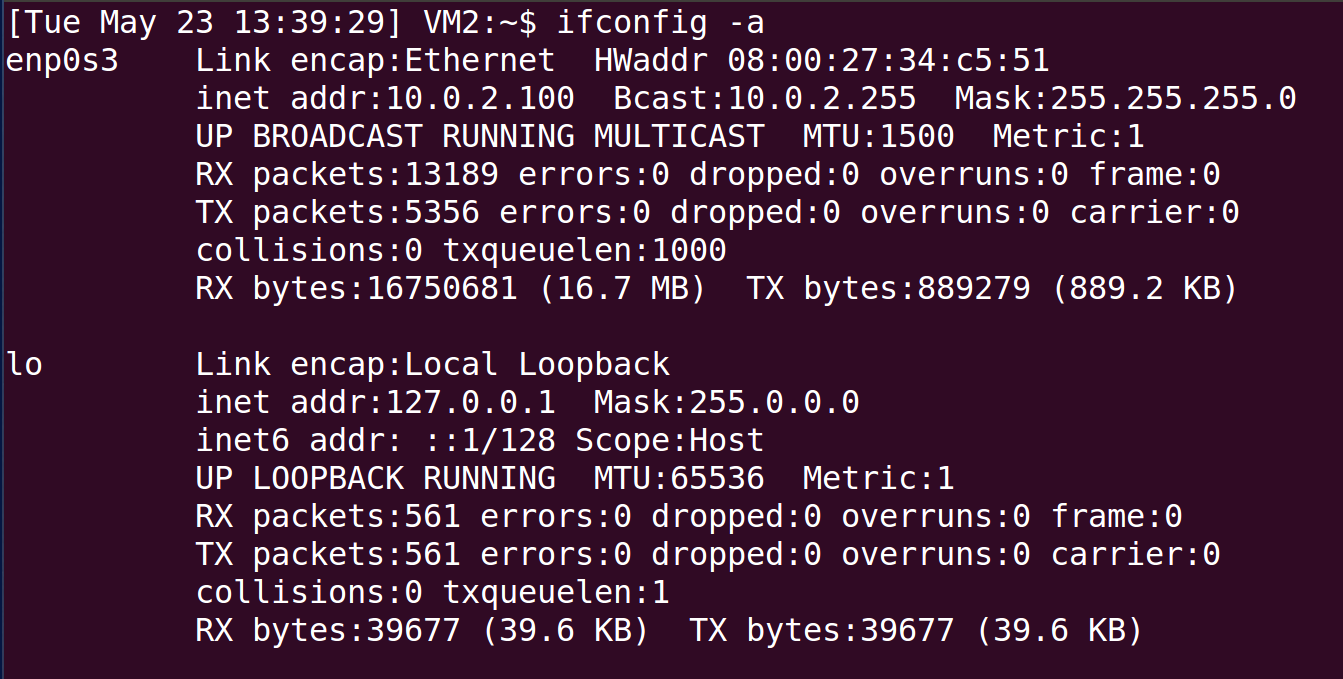
#### תוצאה מצופה

נצליח להעביר packets בין המחשב לשרת ולגשת אל אתר סדרות דרך חיבור ה- VPN שיצרנו, למרות ההגבלות שהגדרנו בחומת האש.

#### ביצוע המשימה

לשם יצירת חיבור הVPN, נצטרך לבצע את השלבים הבאים:

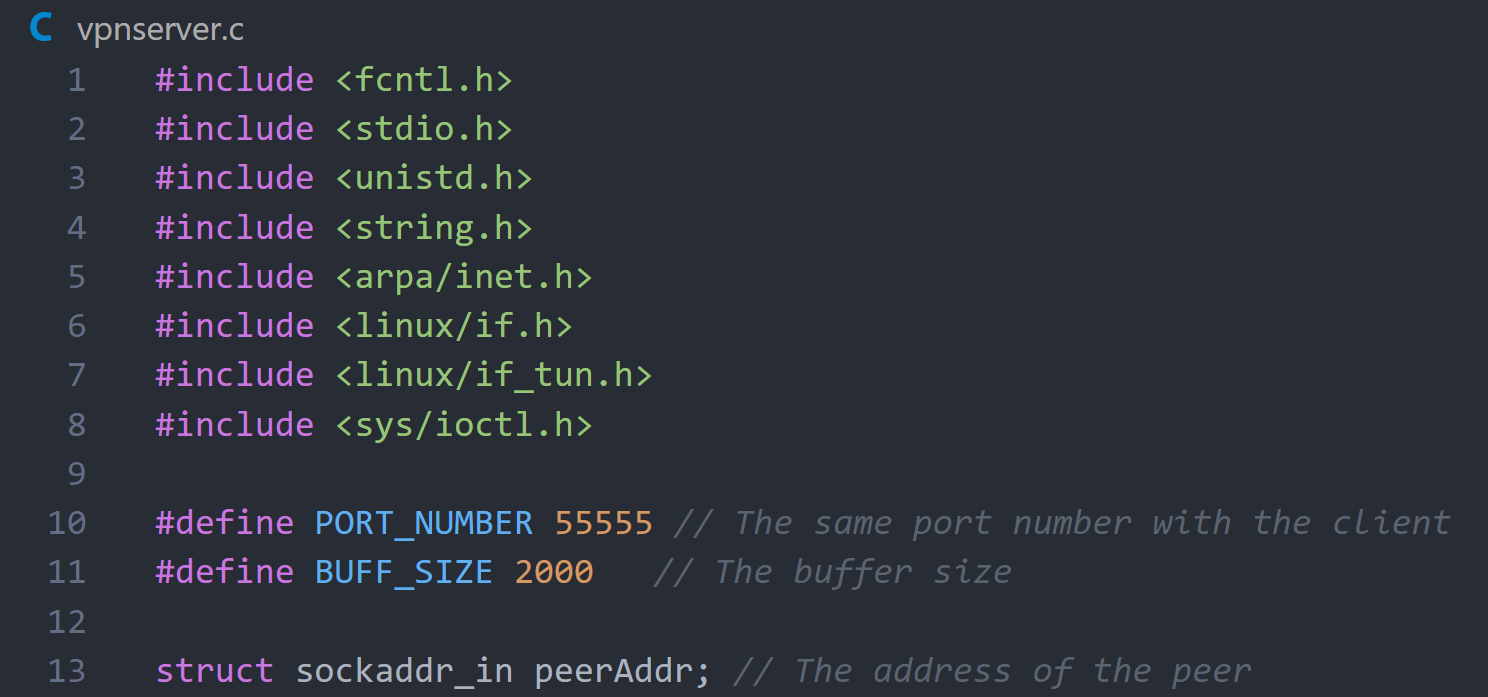
Step 1: Run VPN Server

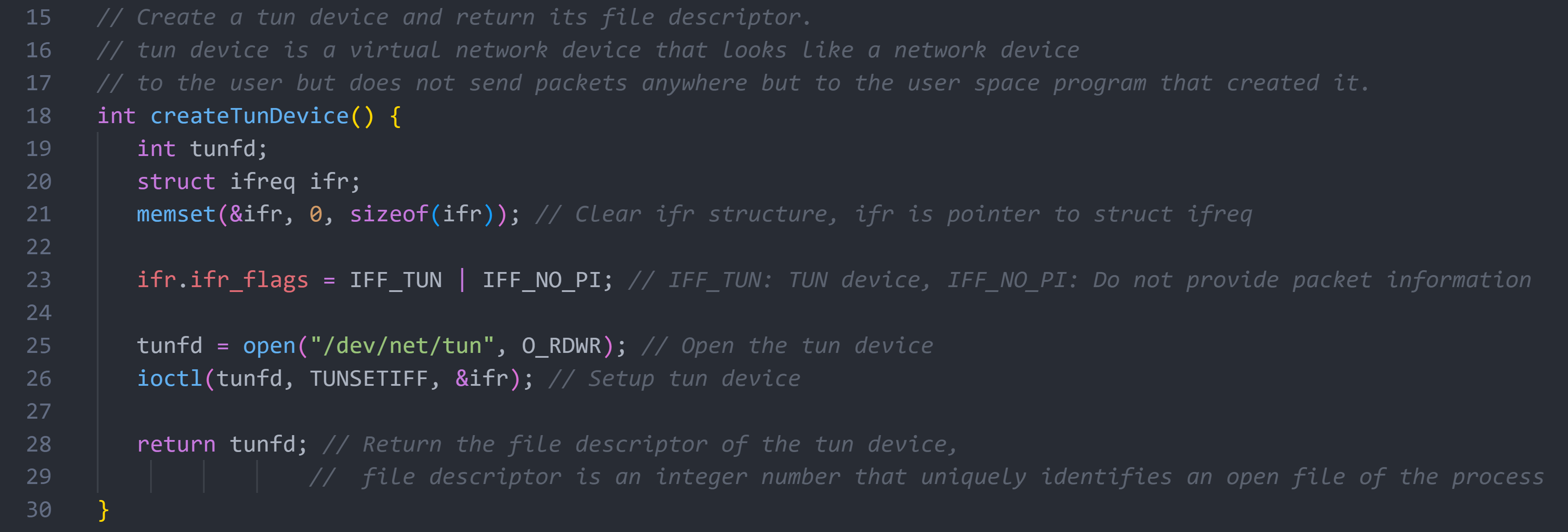
תחילה, נבדוק איזה interfaces קיימים לנו ברשת של מחשב VM2:

ניתן לראות ממשק אחד בשם enp0s3.

במעבדה קיבלנו קבצים מצורפים לטובת יצירת חיבור VPN בין השרת ללקוח בשפת C.

Vpnserver.c

בפורט 55555 נאזין לpackets בפרוטוקול UDP.

הbuff\_size זה הגודל של הבאפר לקריאה ושליחה של נתונים.peerAddr – מבנה שמכיל את הכתובת של הלקוח. הפונקציה אחראית ליצירת מכשיר רשת וירטואלי בשם tun ומחזיר המיקום של הקובץ קונפיגורציה של הinterface.

Tun – interface הפועל בשכבת הרשת במודל הOSI, משמש למטרות ניתוב packets ומאפשר שליחה וקבלה של packets ברשת באופן וירטואלי מתהליכים הרצים במחשב (מדמה רכיב רשת) המתחבר לשרת VPN ומעביר את הpackets ליעד.

Ifr – מבנה המוגדר בקרנל של לינוקס ומשמש לקינפוג ממשקי רשת. תמונה שמכילה טקסט, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטיהפונקציה יוצרת socket בפרוטוקול UDP ומחכה לחיבור. תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

הפונקציה מופעלת כאשר יש מידע בבאפר המוכנים לקריאת מהtun, ושולחת אותו ללקוח באמצעות הsocket שיצרנו. תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטיהפונקציה מופעלת כאשר הנתונים מוכנים לקריאה מה UDP SOCKET שהתקבלו מהלקוח ומעביר אותם לtun. תמונה שמכילה טקסט, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטי

הפונקציה הראשית אשר יוצרת SOCKET וinterface ומחכה למעבר נתונים מאחד הצדדים.

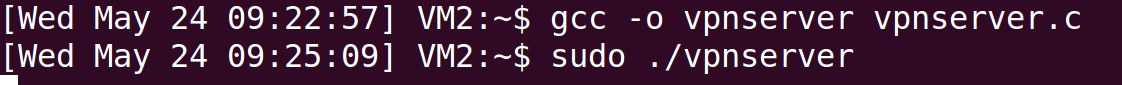
FD\_ZERO מנקה את המצביע לקבצי הקונפיגורציה.

FD\_SET מוסיף קובץ קונפיגורציה למצביע.

הפונקציה select היא פונקציה חוסמת אשר מחכה לקבלת או שליחת נתונים.

בהתאם לקבלת הנתונים מאחד הצדדים, נקראת הפונקציה המתאימה שמעבירה את המידע מה tun לsocket או מהsocket לtun.

כעת, נכנס אל VM2 שהוא השרת VPN שלנו, ונרצה לקמפל את הקוד בקובץ vpnserver.c ולהריץ אותו:



נשים לב שלאחר הקימפול קיבלנו קובץ הרצה:

תמונה שמכילה טקסט, צילום מסך, לוגו, גופן

התיאור נוצר באופן אוטומטי

לאחר מכן נריץ את הפקודה הבאה:

הפקודה הנ"ל משייכת את הIP 192.168.53.1 עם הsubnet /24 אל הinterface שהגדרנו שהוא tun0 והפרמטר up מפעיל את ה interface.

כעת, נבדוק אילו interfaces קיימים בVM2:

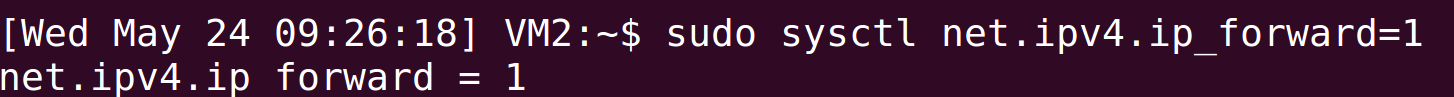
תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי



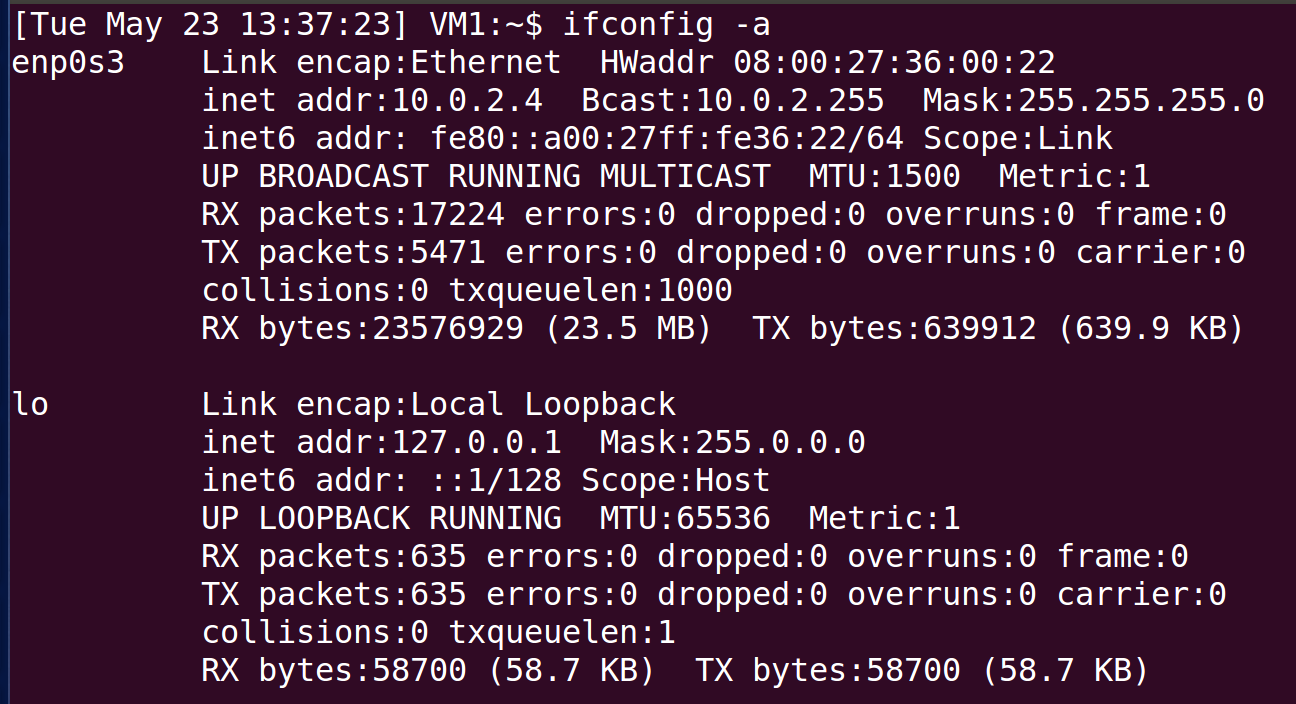
ניתן לראות כי נוצר לנו interface חדש בשם tun0 וה IP שלו הוא 192.168.53.1 כפי שהגדרנו לו.

כדי שVM2 יפעל כשרת, נצטרך להפוך אותו למעין gateway שמעביר את הpackets ולכן נפעיל לו את האופציה להעביר packets דרכו באמצעות הפקודה הבאה:



כעת VM2 יוכל להעביר packets דרך הצינור VPN שניצור בהמשך לרשת החיצונית ולהפך.

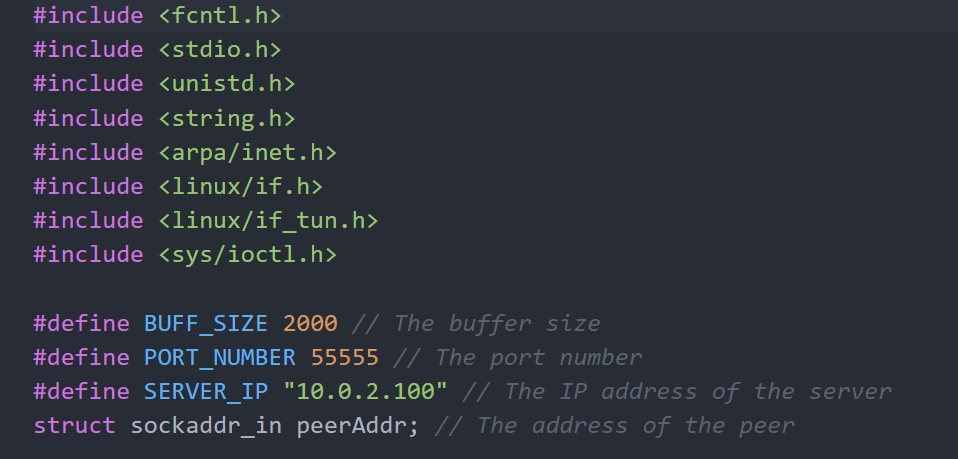
Step 2: Run VPN Client

תחילה, נבדוק איזה interfaces קיימים לנו ברשת של מחשב VM1:

ניתן לראות ממשק אחד בשם enp0s3.

במעבדה קיבלנו קבצים מצורפים לטובת יצירת חיבור VPN בין השרת ללקוח בשפת C.

בקוד בקובץ vpnclient.c שינינו את הIP שהוגדר לשרת לIP של השרת VPN שלנו שהוא 10.0.2.100:

כאן הוספנו גם את הIP של השרת VPN שלנו שהוא 10.0.2.100. תמונה שמכילה טקסט, צילום מסך

התיאור נוצר באופן אוטומטי הפונקציה אחראית ליצירת מכשיר רשת וירטואלי בשם tun ומחזיר המיקום של הקובץ קונפיגורציה של הinterface.



Tun – interface הפועל בשכבת הרשת במודל הOSI, משמש למטרות ניתוב packets ומאפשר שליחה וקבלה של packets ברשת באופן וירטואלי מתהליכים הרצים במחשב (מדמה רכיב רשת) המתחבר לשרת VPN ומעביר את הpackets ליעד.

Ifr – מבנה המוגדר בקרנל של לינוקס ומשמש לקינפוג ממשקי רשת. תמונה שמכילה טקסט, צילום מסך, תוכנה

התיאור נוצר באופן אוטומטיהפונקציה מחברת את הלקוח אל הsocket שנוצר בפרוטקול UDP אצל השרת VPN ושולחת הודעת Hello המסמנת יצירת חיבור מוצלח.תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי הפונקציה מופעלת כאשר יש מידע בבאפר המוכנים לקריאת מהtun, ושולחת אותו ללקוח באמצעות הsocket שיצרנו. תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי הפונקציה מופעלת כאשר הנתונים מוכנים לקריאה מה UDP SOCKET שהתקבלו מהלקוח ומעביר אותם לtun. תמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

הפונקציה הראשית יוצרת interface ברשת הלקוח וחיבור לsocket בשרת VPN.

FD\_ZERO מנקה את המצביע לקבצי הקונפיגורציה.

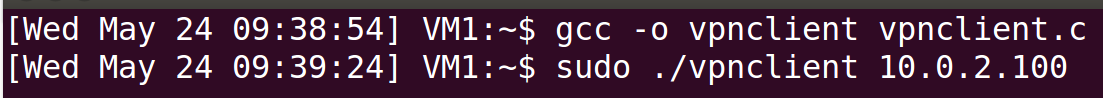
FD\_SET מוסיף קובץ קונפיגורציה למצביע.

Fdset הוא המבנה שמכיל את קבצי הקונפיגורציה של הממשק והsocket.

הפונקציה select היא פונקציה חוסמת אשר מחכה לקבלת או שליחת נתונים.

בהתאם לקבלת הנתונים מאחד הצדדים, נקראת הפונקציה המתאימה שמעבירה את המידע מה tun לsocket או מהsocket לtun.

נכנס אל VM1 שהוא הלקוח שלנו, ונרצה לקמפל את הקוד בקובץ vpnclient.c ולהריץ אותו:



נשים לב שלאחר הקימפול קיבלנו קובץ הרצה:

תמונה שמכילה טקסט, צילום מסך, לוגו, גופן

התיאור נוצר באופן אוטומטי

לאחר מכן נריץ את הפקודה הבאה:



הפקודה הנ"ל משייכת את הIP 192.168.53.5 עם הsubnet /24 אל הinterface שהגדרנו שהוא tun0 והפרמטר up מפעיל את ה interface.

נבדוק שאכן הגדרנו כנדרש וניתן לראות את tun0 ברשימת הממשקים:

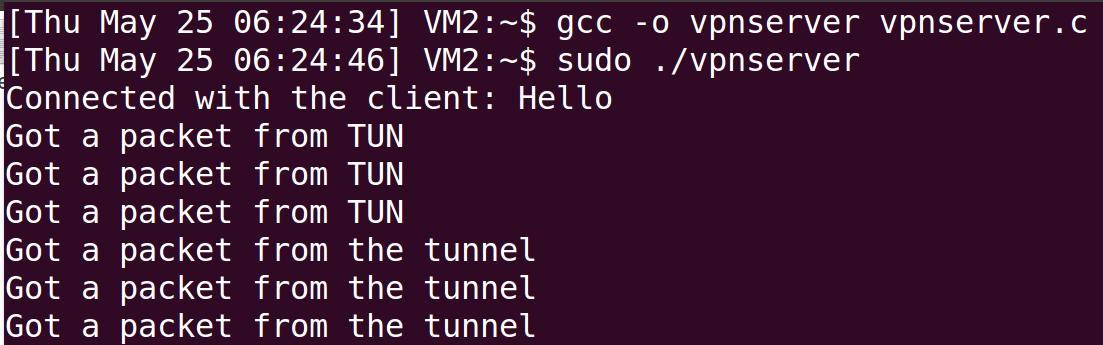
תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי



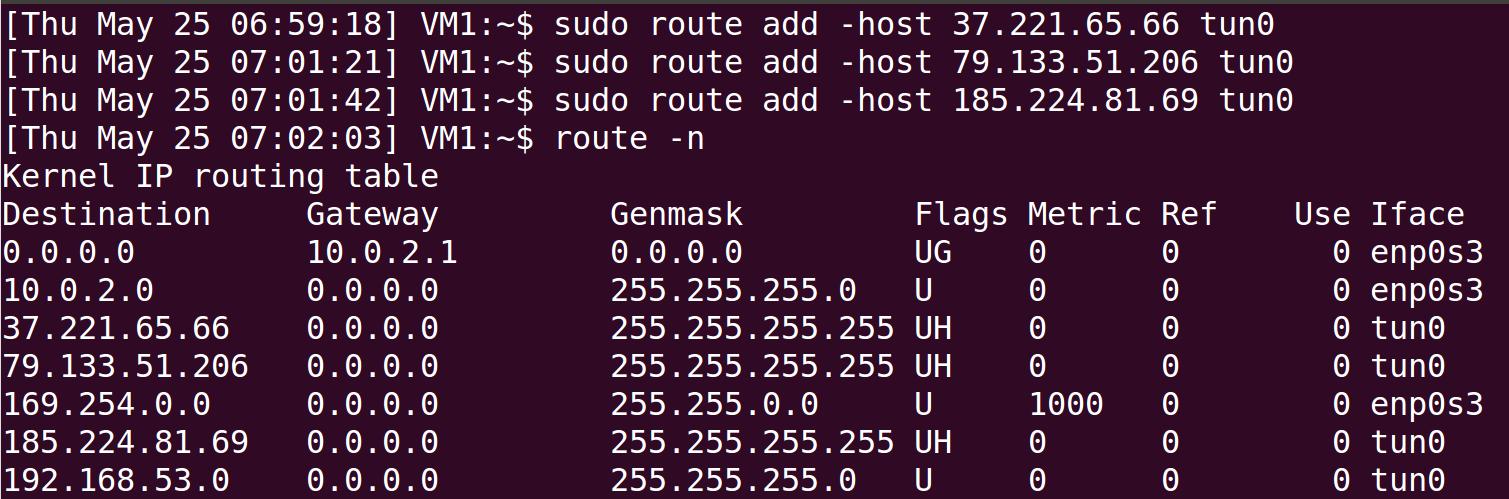
נראה כי התווסף בהצלחה ממשק חדש בשם tun0 עם הip 192.168.53.5.

כעת נבדוק אם החיבור נוצר:



ניתן לראות שהחיבור נוצר בהצלחה בין הVM1 ו VM2 בכך שקיבלנו הודעות פלט מתאימות בטרמינל שמאשרות שנוצר חיבור וירטואלי בtun.

Step 3: Set Up Routing on Client and Server VMs

בשלב נרצה לנתב מחדש בטבלת הניתוב של מחשב VM1 את ה packets לשרתי סדרות שיעברו דרך הממשק הוירטואלי tun0 שבנינו ולבסוף נבדוק שהתווסף בהצלחה לטבלה:ניתן לראות בטבלת הניתוב שהתווספו בהצלחה 3 כתובות IP חדשות המציינות את שרתי סדרות.

-n – מציין שנקבל את כתובות הIP במספר במקום הדומיין.

Step 4: Set Up NAT on Server VM

ננקה את טבלאות החוקים של חומת האש במחשב VM2 כדי להבטיח שלא יהיו לנו חוקים מוגדרים שעלולים להוות חסימה עם הפקודה הבאה:



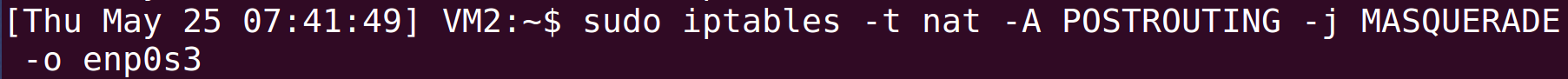


בפקודה הבאה ננקה גם את כל החוקים בטבלת הNAT:



-F – flush מציין ניקוי של הטבלה בזיכרון.

ביצענו זאת כדי לנקות את הנתונים המוגדרים על המחשבים שהעבירו packets ברשת.

בפקודה הבאה נגדיר בVM2 בטבלת הNAT חוק חדש: 

החוק הנ"ל מגדיר שכל הpackets אשר יוצאות לרשת החיצונית יזוהו (התחזות) כאילו עברו דרך interface enp0s3 וכך הpackets שעוברות דרך tun0 יועברו כביכול דרך enp0s3 אל הרשת החיצונית.

ביצענו את כל השלבים הנדרשים לטובת הקמת חיבור VPN TUNNEL בין VM1 וVM2 וכעת נרצה להדגים שניתן לגשת אל שרתי סדרות.

בפקודה הבאה נשלח ping אל שרת סדרות: תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטיניתן לראות כי הפינג עבר בהצלחה וקיבלנו גם echo reply.

נרצה לוודא זאת גם באמצעות wireshark שאכן הצלחנו להעביר את הpackets דרך הVPN TUNNEL שיצרנו:

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

ניתן לראות שאכן הpackets עוברים דרך הVPN TUNNEL שיצרנו, מאחר והם נשלחו עם הIP של VM1 אשר הוגדר בinterface של tun0 שהוא 192.168.53.5.

בתמונה רואים כי הpackets עוברים בהצלחה מVM1 אל שרת סדרות ובחזרה.

נרצה לראות בטרמינל שאכן מתקבלות הודעות פלט מתאימות על מעבר packets בtunnel.

תמונה שמכילה טקסט, צילום מסך, תפריט, גופן

התיאור נוצר באופן אוטומטי

ניתן לראות כי יש תעבורה בtunnel שיצרנו.

נרצה לראות שגם ניתן לגשת אל אתר סדרות:

תמונה שמכילה טקסט, צילום מסך, תוכנה, דף אינטרנט

התיאור נוצר באופן אוטומטי

ניתן לראות שהדף נטען בהצלחה.

נרצה לראות גם את התעבורה בwireshark: תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

נראה כי יש חיבור TCP אל שרתי סדרות מהIP של מחשב VM1 אשר הוגדר בinterface tun0 כ192.168.53.5.

#### סיכום המשימה

הצלחנו לבצע את המשימה.

תחילה הגדרנו שרת VPN ב VM2 כדי שנוכל להתחבר אליו וליצור חיבור VPN TUNNEL שבעזרתו נוכל לעקוף את הגבלות חומת האש.

לאחר מכן, יצרנו חיבור VPN ב VM1 וחיברנו אותו אל שרת הVPN של VM2.

ניתבנו בVM1 שכל התעבורה שלו תעבור דרך הVPN TUNNEL שבנינו והגדרנו במחשב השרת שכל התעבורה תצא לרשת במקום מinterface tun0 מinterface enp0s3.

לבסוף, ניסינו להעביר packets מVM1 אל שרתי סדרות והראינו שהpackets אכן עברו בהצלחה דרך חיבור הVPN TUNNEL שיצרנו.

גילינו כיצד ליצור חיבור VPN TUNNEL בין 2 מחשבים ולמדנו שבעזרתו ניתן לעקוף חוקי חומת אש שהוגדרו.

התוצאות התאימו למצופה מאחר והצלחנו לעקוף את חוקי חומת האש ולגשת אל שרתי סדרות למרות ההגבלות וזה נעשה בעזרת חיבור הVPN TUNNEL שיצרנו.

לא נתקלנו בבעיות במהלך ביצוע המשימה.

**סיכום כללי למעבדה**

ראשית נדרשנו ליצור 2 מכונות וירטואליות המחוברות לאינטרנט עם כתובות IP שונות.

שנית, יצרנו חוקים אשר חוסמים גישה אל כתובות הIP של sdarot ולאחר יצירת החוקים לא היה ניתן לגשת אל שרתי סדרות והדגמנו זאת ע"י שליחת ping.

לסיום, רצינו לעקוף את החוקים שהגדרנו בעזרת יצירת חיבור VPN TUNNEL ועשינו זאת באופן הבא:

תחילה הגדרנו שרת VPN ב VM2 כדי שנוכל להתחבר אליו וליצור חיבור VPN TUNNEL שבעזרתו נוכל לעקוף את הגבלות חומת האש.

לאחר מכן, יצרנו חיבור VPN ב VM1 וחיברנו אותו אל שרת הVPN של VM2.

ניתבנו בVM1 שכל התעבורה שלו תעבור דרך הVPN TUNNEL שבנינו והגדרנו במחשב השרת שכל התעבורה תצא לרשת במקום מinterface tun0 מinterface enp0s3.

לבסוף, ניסינו להעביר packets מVM1 אל שרתי סדרות והראינו שהpackets אכן עברו בהצלחה דרך חיבור הVPN TUNNEL שיצרנו.

לכן ניתן לראות כי הצלחנו לבצע את המשימות במעבדה ולהקים חיבור VPN שעזר לנו לעקוף את החסימות שיצרנו בחומת האש.

**משהו חדשני – APP-ID:**

תמונה שמכילה טקסט, צילום מסך, תוכנה, מספר

התיאור נוצר באופן אוטומטי

App-ID של Palo Alto Networks היא תוכנה המאפשרת לנו לזהות ולשלוט באפליקציות שונות שמעבירות מידע על גבי הרשת שלנו, כולל יישומי VPN.

איך זה עובד ומה אנחנו יכולים לעשות כדי להתגונן מפני ניסיונות עקיפה בעזרת VPN:

זיהוי יישומים: App-ID משתמש בטכניקות שונות כדי לזהות יישומים, כולל התאמת חתימות, פענוח פרוטוקול וניתוח התנהגותי. בעת ניתוח packets זה יכול לזהות במדויק מגוון רחב של יישומים ותבניות שהוגדרו מראש ע"י צוות המחקר כמו פורטים, מבנה וכו' והשוואתם מול מאגר מידע של חתימות נוספות ובדיקה אם יש התאמה, כולל פרוטוקולי VPN פופולריים ואפליקציות VPN.

זיהוי וחסימה של VPN: App-ID יכול לזהות פרוטוקולי VPN כגון OpenVPN, IPSec, PPTP ואחרים. לאחר זיהוי, אתה יכול ליצור מדיניות מעודכנת לחומת האש כדי לחסום או להגביל תעבורת VPN בהתבסס על יישומים או קטגוריות ספציפיות של יישומים. זה מונע יצירת חיבורי VPN או מאפשר רק שימוש מורשה ב-VPN אשר מוגדרים מראש על ידינו.

פענוח SSL/TLS: VPNs מסוימים משתמשים בהצפנת SSL/TLS כדי להסתיר את התעבורה שלהם. חומות האש של Palo Alto Networks מציעות יכולות פענוח SSL/TLS, המאפשרות לנו לבדוק תעבורת VPN מוצפנת ולהחיל App-ID כדי לזהות את אפליקציית ה-VPN שבשימוש.

חתימות מותאמות אישית: אם צצה אפליקציית VPN חדשה או לא מוכרת, אנחנו יכולים ליצור חתימות מותאמות אישית כדי לזהות ולחסום אותה. Palo Alto Networks מספקת גמישות בהגדרת חתימות מותאמות אישית על סמך דפוסים, התנהגויות או תכונות ספציפיות של אפליקציית ה-VPN.

סינון כתובות אתרים: בנוסף לשליטה ברמת היישום, חומות האש של Palo Alto Networks מציעה יכולות סינון כתובות אתרים. אנחנו יכולים למנף קטגוריות סינון כתובות אתרים כדי לחסום גישה לאתרים ידועים של ספקי שירותי VPN, ולמנוע ממשתמשים להוריד ולהשתמש ביישומי VPN.

שילוב User ID: חומות האש של Palo Alto Networks יכולות להשתלב עם טכנולוגיות User ID, כגון Active Directory, כדי לשייך תעבורת רשת למשתמשים ספציפיים. זה מאפשר לנו ליצור מדיניות המכוונת למשתמשים בודדים או לקבוצות, מה שמשפר עוד יותר את השליטה שלנו על השימוש ב-VPN.

כדי להתגונן מפני ניסיונות עקיפת VPN עלינו לבצע את השלבים הבאים:

א. הפעל App-ID: ודא ש-App-ID מופעל בחומת האש של Palo Alto Networks ומוגדרת לזהות ולשלוט בתעבורת VPN.

ב. עדכן חתימות יישומים: עדכן באופן קבוע את מסד הנתונים של חתימות האפליקציה כך שיכלול את יישומי ה-VPN והגרסאות העדכניות ביותר.

ג. צור מדיניות App-ID: הגדר מדיניות חומת אש המאפשרת או חוסמת באופן מפורש יישומי VPN בהתבסס על הדרישות של הארגון שלנו. יש לשקול יצירת מדיניות נפרדת לקבוצות משתמשים או לרשתות שונות.

ד. אפשר פענוח SSL/TLS: אם רלוונטי, אפשר פענוח SSL/TLS כדי לבדוק תעבורת VPN מוצפנת ולהחיל מדיניות App-ID.

ה. אפשר סינון כתובות אתרים: השתמש בסינון כתובות אתרים כדי לחסום גישה לאתרים הקשורים לספקי שירותי VPN.

ו. מעקב וביקורת: מעקב רציף אחר תעבורת הרשת שלנו, סקירת יומנים וביצוע ביקורות כדי לזהות ניסיונות שימוש ב-VPN או הפרות מדיניות.

ז. חינוך משתמשים: למד את העובדים על מדיניות השימוש ב-VPN של החברה, הסיבות מאחוריה והסיכונים הפוטנציאליים הקשורים לעקוף חומת האש. חיזוק החשיבות של עמידה במדיניות האבטחה.

חשוב לעדכן באופן קבוע את תוכנת חומת האש ומסדי נתונים של חתימות כדי להבטיח זיהוי ומניעה יעילים של יישומי VPN וטכניקות חדשות.

עקוב אחר טכנולוגיות VPN מתפתחות ועדכן את מנגנוני ההגנה שלך בהתאם.