בס״ד

רשתות תקשורת מטלה 5

מגישים:

ניסים עטיה 207302027

אליה שלמה 205915663

בס״ד

**sniffer**:

sniffer או רחרחן בעברית הוא סוג של תוכנה הלוכד ומנתח תעבורת רשת.

ניתן להשתמש בו כדי לנתר ולפתור בעיות בתקשורת ברשת, לזהות פרצות אבטחה ולאסוף מידע לניתוח רשת.

Sniffers פועלים על ידי יירוט ופירוש פאקטות נתונים המועברים ברשת. ניתן להשתמש בהם כדי לנתח

תנועה נכנסת ויוצאת וניתן להגדיר אותם כדי לסנן ולהציג סוגים ספציפיים של תנועה.

**חלק א:**

בקוד שלי השתמשתי בספריית pcap היא ספריה בתחום במחשוב ללכידת תעבורה ברשת.

**main:**

הגדרתי מצביע לסוג הרשת(dev) ו-string שיחזיק לי את השגיאות (אם יתרחשו).

לאחר מכן בעזרת פונקציית pcap\_lokupdev הגדרתי באופן אוטומטי את סוג הרשת ואותה שמרתי במשתנה dev ובדקתי אם קרתה שגיאה ואם קרתה אחת הדפסתי הודעה מתאימה.

לאחר מכאן השתמשתי בפונקציה pcap\_open\_live כדי לפתוח את התוכנה להסנפה וגם פה בדקתי אם קרתה שגיאה ואם קרתה אחת הדפסתי הודעה מתאימה למשתמש.

לאחר מכן הגדרתי סינון לפאקטות שהתוכנה תסניף בכדי שהמשתמש יקבל רק פאקטותת מסוג tcp

בעזרת הפונקציות pcap\_cpmpile , pcap\_setfilter וגם פה בדקתי אם קרה שגיאה ואם קרתה אחת הדפסתי הודעה למשתמש.

לאחר כל זה השתמשתי בפונקציה pcap\_loop אשר מאזינה לרשת ואם היא רואה פאקטה מתאימה היא שולחת אותה לפונקציה packet\_handler.

לאחר מכאן סגרתי את התוכנה .

**packet\_handle:**

פונקציה זו מתחילה בזה שהיא פותחת קובץ txt עם תעודת הזהות של שני השותפים אך אם יש קובץ כזה היא תוסיף לכתוב מהנקודה בה הוא נגמר.

לאחר מכאן הגדרתי את התחיליות של שכבת הip ושל הtcp לאחר מכאן שמתי בתוך התחיליות האלה את ההסטים המתאימים בכדי שכל תחילית תצביע על ההתחלה שהמתאימה לשכבה שלה.

לאחר מכאן בדקתי האם הפאקטה שהתקבלה היא מסוג psh כי אני רוצה להציג פאקטות עם data.

אחרי כל זה הפונקציה כותבת לקובץ את כל השדאות הדרושות במטלה ואחרי שהיא סיימה לכתוב לקובץ היא סוגרת אותו.

**answer:**

sniffer דורשת הרשאת root על מנת לתפוס תעבורת רשת הסיבה לכך היא שהיא צריכה לגשת לממשק הרשת ולקרוא את נתוני החבילות שמועברים ברשת, ללא הרשאה זו התוכנית לא תוכל לגשת לנתונים הדרושים ולא תוכל לפעול כראוי.

סיבה נוספת לכך שהתוכנית דורשת הרשאת root היא בכלל שהיא צריכה להיות מסוגלת לעקוף אמצעי אבטחה כגון חומת אש ובקרות גישה שקיימים כדי להגן על המערכת.

ללא הרשאה זו התוכנית תיחסם מגישה לנתונים הדרושים ולא תוכל ללכוד את התעבורת רשת שהיא צריכה.

בס״ד

**דוגמת הרצה:**

בכדי להריץ את התוכנה נפתח טרמינל על התקיה של הקבצים ונכתוב את הפקודה make לאחר מכאן יתווספו קבצי הרצה של התוכניות וכדי להריץ את התוכנה של הsniffer נכתוב את הפקודה

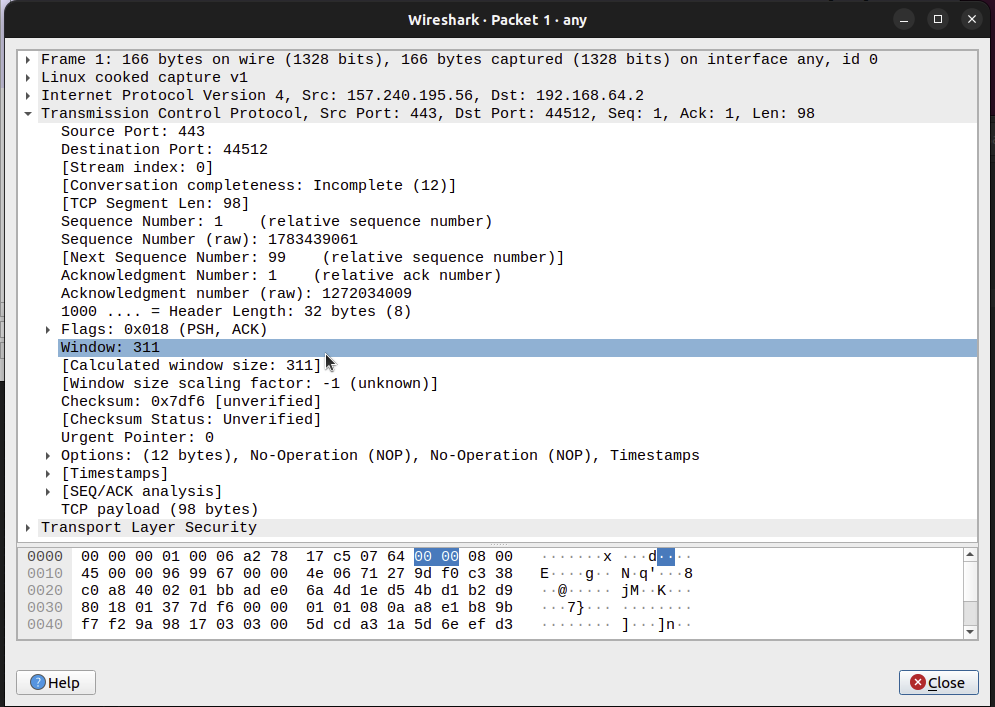
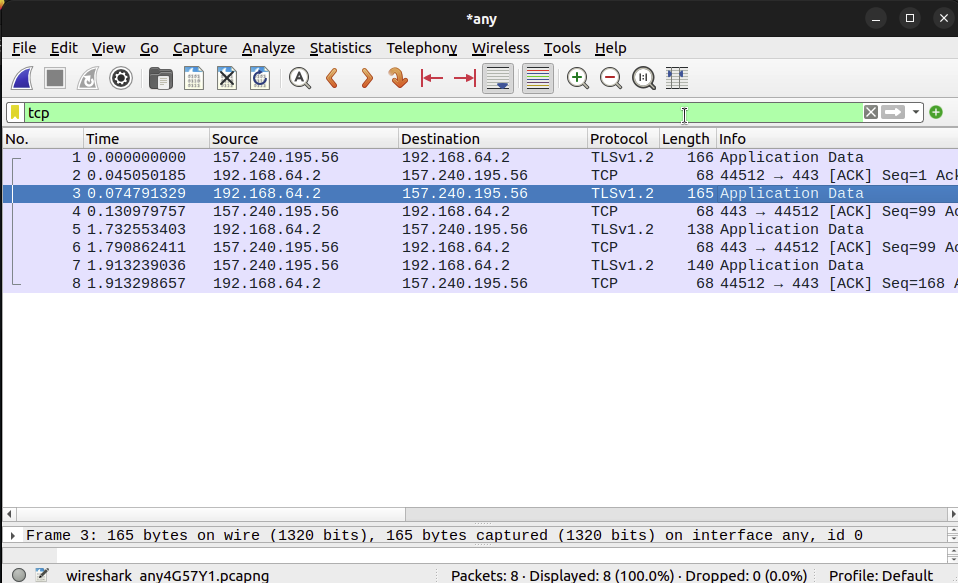
sudo ./sniffer לאחר מכאן נכניס לטרמינל את הקוד של הroot

(פתחתי דף של אתר אינטרנט ולכן יהיו תנועות ברשת).

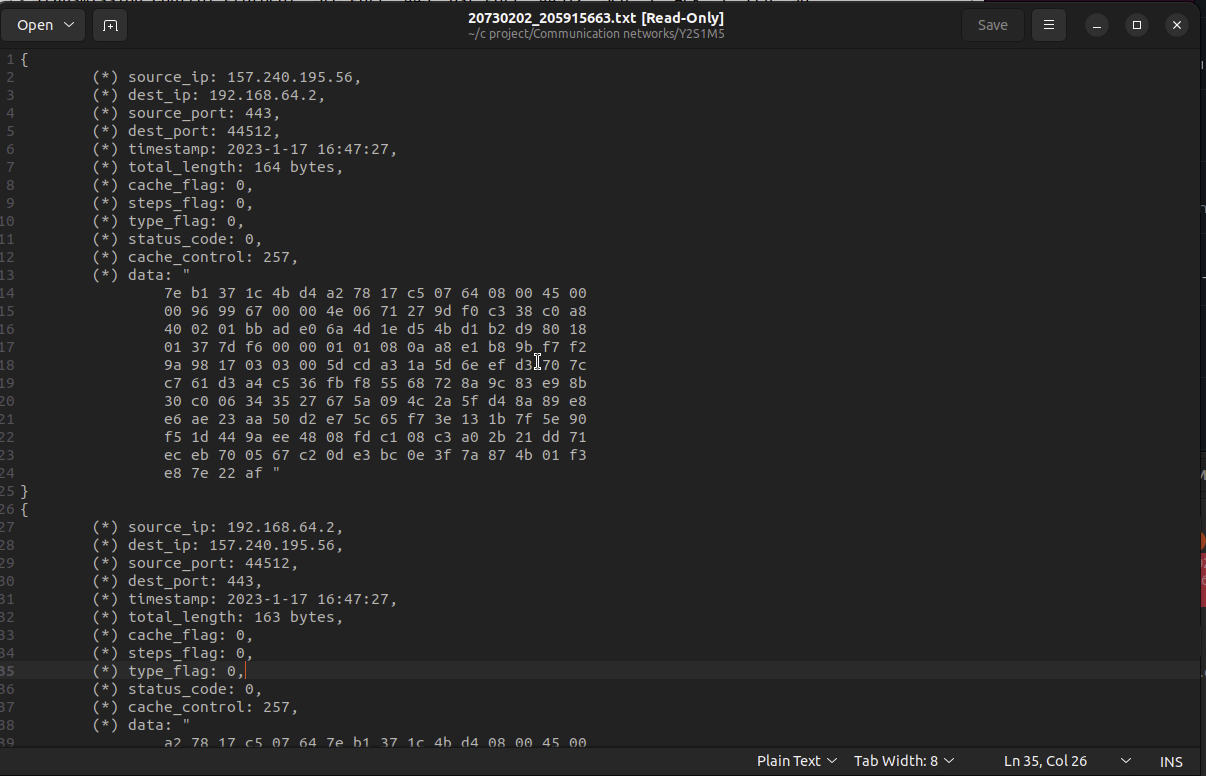
נוכל לראות בווירשארק שנשלחה ברשת פאקטה מסוג tcp מ-ip מכתובת 157.240.195.56 לכתובת 192.168.64.2 (מסומן בעיגול)אם נכנס לפרוט של הפריים הזה ולפרוט בשכבת התעבורה של הפריים הזה נוכל לראות שפורט המקור הוא 443 ופורט היעד הוא 44512(מסומן בעיגול) ועוד.

ואכן נפתח קובץ ובו נוכל לראות שאכן נקלטה פאקטה זו נוכל לראות זאת לפי כל הערכים שיש בקובץ ולראות שהם זהים לדוגמא כמו שציינו שראינו בווירשארק שהפאקטה הראשונה היא מכתובת 157.240.195.56 לכתובת 192.168.64.2 וכן הפורט מקור הוא 443 ופורט היעד הוא 44512 וכן נוכל לראות בקובץ שזה תואם(מסומן במלבן).

נוכל לראות גם שגם הפאקטה השניה מתאימה בדיוק לפלט שיצא לתוך הקובץ מהפונקציה.



בס״ד



**Spoofer:**

spoofer הוא סוג של תוכנה או כלי המשמש לזיוף של פאקטות. הוא משמש בדרך כלל לביצוע פעילויות זדוניות כגון התקפות, שהיא טכניקה המשמשת להשגת גישה לא מורשית למחשב או לרשת על ידי התחפשות למארח מהימן. ניתן להשתמש ב-spoofer גם כדי להסתיר את המקור של חבילה ולגרום לה להיראות כאילו הגיעה ממקור אחר. זה יכול לשמש למטרות שונות כמו התחמקות מגילוי, עקיפת אמצעי אבטחה ועוד.

**חלק ב:**

**main:**

יצרתי משתנה ans שיחזיק לי את התשובה של המשתמש.

ביקשתי מהמשתמש להכניס 1 אם הוא רוצה לעשות spoof לudp packet או להכניס 2 אם הוא רוצה לעשות spoof ל- tcp pact או להכניס 3 אם הוא רוצה לעשות spoof לicmp packet ואם הוא רוצה לצאת הוא צריך להכניס 0.

אם המשתמש הכניס 1 יתבצע הפונקציה spoof\_UDP ואם אכן ישלח הוא יקבל הודעה על כך.

אם המשתמש יכניס 2 תתבצע הפונקציה spoof\_TCP ואם אכן ישלח הוא יקבל הודעה על כך.

אם המשתמש יכניס 3 תתבצע הפונקציה spoof\_ICMP ואם אכן ישלח הוא יקבל הודעה על כך.

בס״ד

לאחר מכאן ביקשתי מהמשתמש שוב להכניס את הבחירה שלו וחזרתי על זה בלולאת while אם המשתמש הכניס 0 הוא יקבל הודעה שהתוכנית הסתיימה.

**spoof\_udp:**

יצרתי מערך בשם buffer שהוא ייצג את הפאקטה שלי, ואיפסתי אותו.

יצרתי את התחיליות הנדרשות והסטתי אותם כדי שיצביעו על ההתחלה של השכבה שלהן.

יצרתי string שהוא masage של הפאקטה הזאת והוספתי אותה למקום שלה בפאקטה

לאחר מכאן הכנסתי ערכים לפאקטה כגון שהsource port הוא 1234 ושה-destination pord הוא 9090 ואת האורך של הupdheader ושהchaecksum הוא 0.

ושהפרוטוקול יהיה פרוטוקול של udp עם ערך 17.

וכן עוד ועוד דברים בשדות המתאים להם.

לאחר מכן שלחתי את התחילית של הip לפונקציית send\_raw\_ip\_packet.

**spoof\_tcp:**

יצרתי מערך בשם buffer שהוא ייצג את הפאקטה שלי, ואיפסתי אותו.

יצרתי את התחיליות הנדרשות והסטתי אותם כדי שיצביעו על ההתחלה של השכבה שלהן.

יצרתי string שהוא masage של הפאקטה הזאת והוספתי אותה למקום שלה בפאקטה

לאחר מכאן הכנסתי ערכים לפאקטה כגון שהsource port הוא 12345 ושה-destination pord הוא 80 ושהchaecksum הוא 0, וכן עוד נתונים המתאימים לשדה של הtcpheader.

בשכבת הapp הכנסתי ערכים לשדאות כמו שהפרוטוקול יהיה פרוטוקול tcp ואת האורך של הdata+header ואת אורך הheader, וכן עוד נתונים המתאימים לשדה של הapp.

לאחר מכן שלחתי את התחילית של הip לפונקציית send\_raw\_ip\_packet.

**spoof\_icmp:**

יצרתי מערך בשם buffer שהוא ייצג את הפאקטה שלי, ואיפסתי אותו.

יצרתי את התחיליות הנדרשות והסטתי אותם כדי שיצביעו על ההתחלה של השכבה שלהן.

בשכבת הapp הכנסתי ערכים לשדאות כמו שהפרוטוקול יהיה פרוטוקול icmp ואת האורך של הdata+header ואת אורך הheader ושהsource ip הוא 1.2.3.4 ושה-destination ip הוא 10.0.2.5 ואת האורך של הicmpheader ואת החישוב של הchaecksum בשכבה זו ושהttl הוא 20, וכן עוד נתונים המתאימים לשדה של הapp.

ובשכבת הnetwork שמתי שהtype יהיה מסוג request ואת החישוב של הchaecksum בשכבה זו.

לאחר מכן שלחתי את התחילית של הip לפונקציית send\_raw\_ip\_packet.

send\_raw\_ip\_**packet**:

פונקציה זו פותחת socket המאפשרת לשנות דברים בחבילה ולהשאיר דברים בחבילה הנשלחת.

יצרתי raw socket בעזרת פונקציה socket.

לאחר מכאן נתתי את היכולת לsocket לשנות דברים ולהשאיר דברים בחבילה בעזרת פונקציה setsockpt.

לאחר מכאן סיפקתי את המידע הדרוש על היעד, ושלחתי את החבילה וסגרתי את הקובץ.

בס״ד

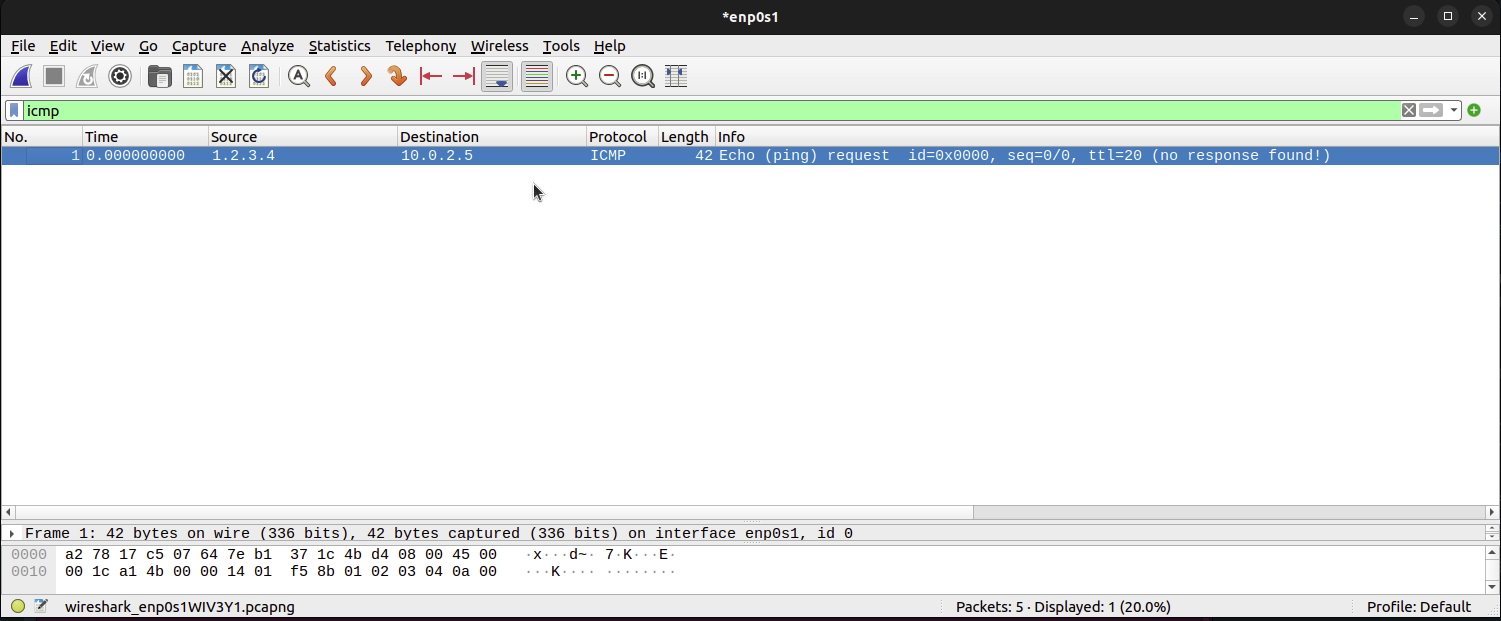
**דוגמאת הרצה:**

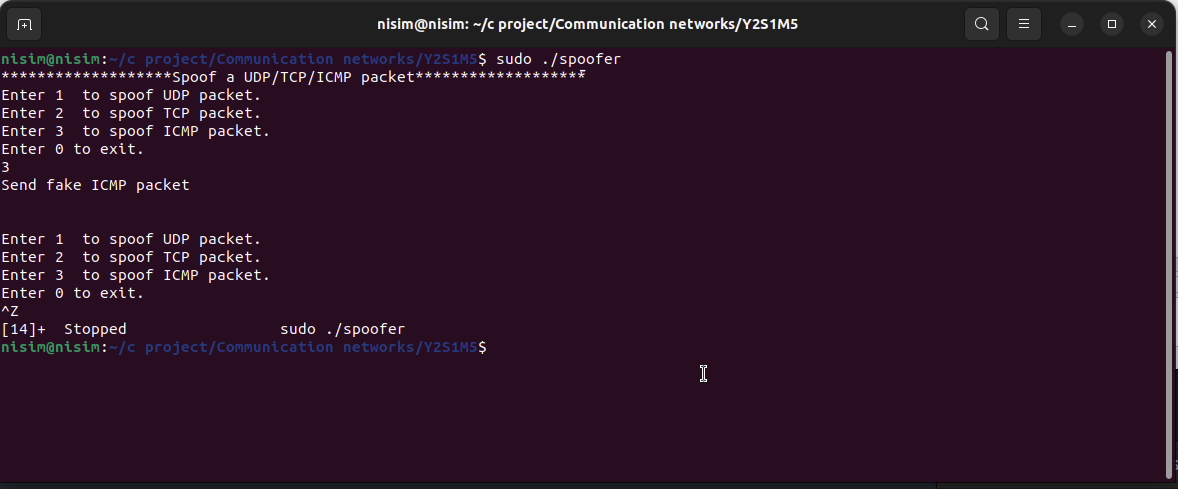
בכדי להריץ את התוכנה נפתח טרמינל על התקיה של הקבצים ונכתוב את הפקודה make לאחר מכאן יתווספו קבצי הרצה של התוכניות וכדי להריץ את התוכנה של הspoofer נכתוב את הפקודה

sudo ./spoofer לאחר מכאן נכניס לטרמינל את הקוד של הroot.

נוכל לראות בטרמינל שלאחר הכנסת הפקודה הוא מוכן לעשות spoof לפאקטות מסוג udp tcp icmp ומחכה להחלטת המשתמש ברגע שהכנסתי לו את הספרה 3 ולחצתי אנטר נשלחה לי הודעה שנשלחה פאקטה מסוג icmp כמו שרציתי(כי הכנסתי 3) ושוב התוכנה מחכה לבחירה שלי.

נוכל לראות בווירשארק עם סינון לicmp שאכן נשלחה חבילה אחת של icmp(מסומן בחץ) וכל השדות שלה הן כמו השדות שאנחנו הכנסו בקוד לדוגמא שהחבילה היא מכתובת 1.2.3.4 לכתובת 10.0.2.5 (מסומן במלבן) ושסוג ההודעה הזאת היא request(מסומן בחץ) ושהttl שלה הוא 20 מסומן בחץ(20)

****

****

**answer 1 :**

לא, שדה ה-IP packet length חייב להכיל את האורך האמיתי של החבילה בבתים, כולל התחילית. הגדרתו לערך סתמי תגרום לחבילה להיחשב כבלתי חוקית ולהפיל אותה על ידי כל הנתבים בדרך.

בס״ד

**answer 2:**

בעת שימוש בתכנות raw socket ,מערכות הפעלה מסוימות יחשבו ויגדירו באופן אוטומטי את checksum עבור תחילית ה-IP, בעוד שאחרות עשויות להשאיר בידי המשתמש לחשב ולקבוע את סכום הבדיקה.

אם מערכת ההפעלה לא מחשבת את סכום הבדיקה עבור כותרת ה-IP, אזי זה יהיה תלוי במשתמש לעשות זאת על מנת שהמנה תועבר בצורה נכונה.

**חלק ג:**

בחלק זה נתבקשנו ליצור תוכנה אשר תסניף פקאטות icmp מסוג request, ולשלוח פקאטות icmp זדוניות מוסג replay.

זדוניות הכוונה שהן לא יוצאות מכתובת הip dest של פקאטת הrequest, אלא ממחשב צד שלישי אשר מאזין ומחליף בין הsource ip לdest ip , ומדמה כאילו הפקאטה שהוא שולח יצאה מהdest ip של פקאטת הrequest.

**main**:

בפונקציה זו הגדרנו משתנה בשם dev שהוא יהיה שם הרשת עליו יאזין התוקף. הגדרנו string בשם errbuf שהוא יכיל לי את השגיאות אם קרו.

בעזרת פונקציית pcap\_open\_live פתחנו את התוכנה להסנפה של פקאטות על הרשת.

בדקנו אם קרתה שגיאה, ואם קרתה הדפסנו הודעה בהתאם.

הגדרנו משתנה filter\_esp שהוא יהיה הסטרינג icmp בכדי לסנן פאקטות icmp.

לאחר מכן הגדרתי סינון לפאקטות שהתוכנה תסניף בכדי שהמשתמש יקבל רק פאקטותת מסוג icmp

בעזרת הפונקציות בעזרת הפונקציות pcap\_cpmpile , pcap\_setfilter וגם פה בדקתי אם קרה שגיאה ואם קרתה אחת הדפסתי הודעה למשתמש.

לאחר כל אלו בעזרת פונקציית pcap\_loop קיבלנו פקאטות מהרשת המתאימה ומהסינונים המתאימים וכל פקאטה שקיבלנו נשלחה לפונקציית spoof\_replay, לאחר מכן סגרתי מכשיר ההסנפה והחזרתי 0.

**spoof**\_rplay:

בפונקציה זו הגדרתי מצביעים מסוג ethheader ipheadar וicmpheader בשמות eth,ip\_p וicmp\_p בהתאמה, והסטתנו אותם למקומות הרלוונטיים מבחינת התחיליות של כל שכבה.

בדקנו אם הפקאטה שתפסנו היא מסוג ip והאם היא מסוג icmp והאם היא מסוג request. אם כל אלו מתקיימים אני מדפיס הודעה שאכן נתפסה פקאטה, ואת כתובות ip\_source ו ip\_dest שלה.

לאחר מכן נחליף לפקטה זו את הicmp\_type שלה להיות icmp\_echoreplay את הicmp\_code שלה להיות 0, נאפס לה את הicmp\_checksum ולאחר מכן חישבנו אותו מחדש. עשינו swap בין

בס״ד

הip\_dest ל ip\_source הגדרנו שip\_ttl שווה 120, ip\_flag שווה ל0, וip\_ident שווה ל0, ןip\_ver שווה ל4.

ושלחנו את ip\_p לפונקציית send\_raw\_ip\_packet.

**send\_raw\_ip\_packet:**

פונקציה זו פותחת socket המאפשרת לשנות דברים בחבילה ולהשאיר דברים בחבילה הנשלחת.

יצרתי raw socket בעזרת פונקציה socket.

לאחר מכאן נתתי את היכולת לsocket לשנות דברים ולהשאיר דברים בחבילה בעזרת פונקציה setsockpt.

לאחר מכאן סיפקתי את המידע הדרוש על היעד, ושלחתי את החבילה.

**in\_cksum:**

פונקציה זו מחשבת ומחזירה את check\_sum על ידי כך שהיא מקבלת מצביע לפקאטה ואת אורכה.

**דוגמאת הרצה:**

בכדי להריץ את התוכנה נפתח טרמינל (1) על התקיה של הקבצים ונכתוב את הפקודה make לאחר מכאן יתווספו קבצי הרצה של התוכניות וכדי להריץ את התוכנה של sniff\_and\_spoof נכתוב את הפקודה

sudo ./sniff\_and\_spoof לאחר מכאן נכניס לטרמינל את הקוד של הroot.

כרגע תוכנית זאת בהאזנה.

ריצה ראשונה-

נפתח טרמינל נוסף ונכתוב את הפקודה ping ולאחר מכן את כתובת הip של host b, לאחר מכן נלחץ קונטרול z, וזה יעצור לנו את שליחת ה-ping-ים, נוכל לראות בטרמינל 1 שאכן התוכנה מסניפה icmp packets מכתבות המקור לככתובת הip של host b.

ריצה שניה-

נכתוב את הפקודה ping ולאחר מכן את כתובת הip שלgoogle dns שהוא 8.8.8.8, לאחר כמה רגעים נלחץ קונטרול z, וזה יעצור לנו את שליחת הpingים, נוכל לראות בטרמינל 1 שאכן התוכנה מסניפה icmp packets מכתבות המקור לכתובת הip של google dns ושהיא אכן שולחת icmp packets מזוייפות(מסומן במלבן).

ריצה שלישית-

נכתוב את הפקודה ping ולאחר מכן את כתובת הip שלכתובת שאינה חוקית לדוגמא1.2.3.4, לאחר רגעים אחדים נלחץ קונטרול z, וזה יעצור לנו את שליחת הpingים, נוכל לראות בטרמינל 1 שאכן התוכנה מסניפה icmp packets מכתבות המקור לככתובת 1.2.3.4ושהיא אכן שולחת icmp packets מזוייפת(מסומן במלבן).

נוכל לראות בטרמינל 2 שאכן מתקבלות icmp replay מכתבות 1.2.3.4 למרות שזאת כתובת שאינה חוקית, ולכן נוכל להסיק מכך שזוהי עבודה של התוכנה שלנו(מסומן בחץ).

נ.ב הייתה לנו בעיה בעניין הדוקרים, חקרנו את האינטרנט(הפכנו כל פינה ברשת) בכדי להבין מאיפה נובעת הבעיה ולפתור אותה. דיברנו עם מרצים ומתרגלים בעניין זה ולא קיבלנו מענה. ולכן לא יוכלנו להריץ את הקוד שלנו מהדוקרים, אך כפי שניתן לראות כאן הקוד שלנו עובד.

בס״ד

