עבודת גמר 5 יח"ל

נושא העבודה : ניטור ובקרת רשת מקומית

שם תלמיד : רון גוטהייט

ת.ז תלמיד : 318939816

שם בית ספר ועיר : קריית החינוך ע"ש עמוס דה-שליט, רחובות

שם המנחה : ערן בינט

מועד הגשה : 9.3.2018

תוכן עניינים

1. מבוא

1.1. נושא העבודה

נושא הפרויקט עליו בחרתי לעבוד הוא Network Analyzer. במסגרת זו המערכת תעסוק בניתור הרשת המקומית של המחשב, ובבקרת הפעילות המתבצעת במחשבי הרשת השונים.
המערכת כאמור מתעסקת בתחום הרשת המקומית, רשת מחשבים מקומית המחוברת ע"י גוף מרכזי המקשר בין הרשת לעצמה, ובין הרשת לעולם החיצוני. בחרתי לממש פונקציות רבות בפרויקט בעזרת הכלי – Scapy כלי המאפשר שליחה וקבלה פקטות הנוצרות באופן ידני ע"י המשתמש.
בעזרת Scapy אני מבצע לדוגמה Fingerprinting על המחשבים השונים ברשת .
המערכת מאפשרת למשתמש לראות את המחשבים השונים ברשת, לצפות בפרטים של כל מחשב, ולנתר אותם.

מטרות מרכזיות

המערכת תאפשר למשתמש לראות את המחשבים השונים ברשת (מחשבים נייחים, ניידים ומכשירים סלולריים), פרטים על כל מחשב כגון כתובות לוגיות ופיזיות, מערכת הפעלה, פרוטוקולים שרצים על המחשב וכו'. כמו כן, תהיה אפשרות לנתר מחשבים אלו. לדוגמה – להגדיר תהליכים בלתי חוקיים אשר ברגע הרצתם על מחשב ברשת, יקבל מנהל הרשת (admin) התראה על כך.

1.2. רציונל

המוטיבציה שלי לפיתוח הפרויקט נבעה מרצוני ליצור כלי ידידותי למשתמש וקל לשימוש אשר יאפשר הבנה עמוקה של הרשת המקומית של המחשב שלו, ולאפשר לו לשלוט במידת האפשר על הנעשה ברשת המקומית.

באמצעות הפרויקט אני מקווה להגיע להבנה עמוקה יותר על נושא הרשת המקומית, ובאופן כללי על רשתות, ולהעמיק את הידע שלי בתחום מערכות ההפעלה. בנוסף, אני מקווה שאוכל להרחיב את יכולות התכנות וההתעסקות במחשבים ותחום הסייבר בפרט בעקבות הכנת הפרויקט.

1.3. קישור לחומר הנלמד

הפרויקט עוסק במגוון תחומים אשר למדתי במסגרת שיעורי הסייבר. תחום מרכזי אשר מתבטא בפרויקט הוא תחום התקשורת והרשתות. בשיעורים למדנו רבות על התחום, ובנוסף למדתי המון למידה עצמית בבית, והבנת החומר לעומק יותר על מנת להגיע לרמה הנדרשת בפרויקט שלי.

התקשורת בין מחשבים מתבצעת באמצעות Sockets ברמת ה TCP/IP רמה 4 של מודל 7 השכבות. בנוסף, באמצעות הכלי scapy אני שולח ומאזין לפקטות בעיקר ברמת פרוטוקול ה ARP.

תחום נוסף בו הפרויקט נוגע הוא תחום מערכות ההפעלה – על מנת להשיג מידע מסוים, כגון תהליכים רצים במחשב, השתמשתי בקוד אשר נוגע בתחום זה.

2. תיאוריה

2.1. תיאוריה

רשת מחשבים מקומית או LAN (Local Area Network) היא רשת שבוניה מקבוצת מחשבים קשורים ע"י קו תקשורת משותף או קישור אלחוטי לשרת. הרשת מתפרסת על פני שטח גאוגרפי מוגבל, בדרך כלל בתוך בניין אחד או בניינים סמוכים. הרשת מחוברת לנתב (ראוטר), המקשר בין הרשת המקומית לעולם החיצוני.

מבחינת מודל 7 השכבות, הרשת המקומית היא עד רמה 2. מודל ה OSI (Open System Interconnection), או מודל 7 השכבות, הוא דגם המציג את הפעולות השונות הדרשות על מנת להעביר נתונים ברשת תקשורת, ואת הסדר שלהם. הוא בעצם מחלק כל פיסת מידע אשר נשלחת ברשת תקשורת ל7 רמות שונות, וכל שכבה מייצגת רמת תקשורת אחרת. פרוטוקול ARP, בתרגום חופשי – פרוטוקול תרגום כתובות, הוא פרוטוקול אשר נמצא בין רמה 2 לרמה 3 במודל ה OSI והפרוטוקול מתמקד ברשת המקומית. בעזרת הפרוטוקול ניתן לעשות טבלה המקשרת בין כתובת לוגית (IP) לבין הכתובת הפיזית של המחשב (MAC). התהליך מתחיל בשליחת פקטת בקשת ARP (ARP request), אשר נשלחת בהודעת תפוצה למחשבים ברשת. התפוצה מתבצעת ע"י שימת כתובת FF:FF:FF:FF:FF בכתובת היעד הפיזית, וכך המחשבים ברשת יודעים שאכן מדבור בפקטת תפוצה. כאשר המחשבים מקבלים את הפקטה, הם מחזירים פקטה מתאימה למפיץ ההודעה, עם הכתובת הלוגית והפיזית שלהם במקומות המתאימים. לאחר הגעת הפקטה למחשב המקור, הוא ממלא את הפרטים בזיכרונו, וכך יכול למפות את הרשת המקומית שלו. למשתמש יש אף גישה לטבלה ועל מנת לצפות בה ניתן לרשום ב cmd את הפקודה "arp -a".

בעזרת ARP ופרוטוקולים נוספים ניתן לבצע פעולות רבות, ולהבין המון אודות המחשבים ברשת. אחת הדוגמאות לשימוש בפרוטוקולים על מנת ללמוד עוד על הרשת המקומית היא Fingerprinting.

Fingerprinting היא שיטה המאפשרת למשתמש למצוא מידע על מחשב מסוים, ולזהותו. המידע האפשרי להשיג נע בין כתובות של המחשב, מידע על החומרה שלו, מערכת ההפעלה של המחשב ועוד מגוון דברים. את המידע אוספים באמצעות האזנה לפקטות שהמחשב שולח ברשת וניתוח המידע בהם ע"י פרמטרים קבועים מראש. ניתן להבחין בין 2 סוגים של Fingerprinting - אקטיבי ופאסיבי.

בגישה האקטיבית מחשב המקור שולח למחשב היעד פקטות ומידע מסוים, ובוחן את התגובה שלו בהתאם למידע הנשלח. לעומת זאת, בגישה הפאסיבית מחשב המקור אינו שולח למחשב היעד פקטות מסוימות, אלא רק מאזין לתעבורת הרשת וממתין לפקטות ממחשב היעד, שאותן הוא מנתח. לגישה הפאסיבית יש החיסרון שלעיתים יכול לעבור פרקי זמן משמעותיים בין שליחת פקטות מתאימות ממחשב היעד, והדבר עלול לארוך זמן רב שרובו הוא בעצם זמן מת בו המחשב מאזין לתעבורת הרשת ומחכה למידע ממחשב היעד. אולם, אם הפעולה מתבצעת למטרות זדון, מטרת בעל מחשב המקור היא שלא יעלו עליו, ולכן בגישה הפאסיבית כמעט ואין דרך לעלות על כך שהוא מאזין ומנתח את התשובות שלו.

2.2. מוצרים קיימים

לינק למוצר דומה: Nmap - https://nmap.org/

Nmap היא תכונת סריקת רשת, המשמשת לגילוי מתחמים ושירותים ברשת מחשבים. Nmap פועל על ידי שיטת ה Active Fingerprinting – שליחת פקטה בעלות מבנה מסוים אל המחשב הרצוי, וניתוח התגובה המתקבלת ממנו. בנוסף ליכולות הסריקה, יש למערכת גם יכולות תקיפה. המערכת עובדת ע"י כתיבת קוד וביצועו במערכת. המערכת מסוגלת למפות מערכות הפעלה של מחשבים ברשת, זיהוי שירותים, ובלבול האויב ע"י שינוי פרטים מסוימים בפקטה שאותה שולחת ובכך ללמוד עוד על המחשב. התוכנה היא תכנה חזקה מאוד, ואחד היתרונות הבולטים שלה היא שהאפשרויות בתכנה הן רבות ומגוונות מכיוון שהמשתמש כותב קוד אותה התכנה תריץ, בנוסף לכך שהיא באה עם המון אופציות מובנות ואף אפשרויות להרחבות נוספות.

ייחודיות המוצר שלי על המוצר הזה, היא יכולת המערכת לנתר את המחשבים ברשת, בנוסף לסריקה שלהם. בנוסף, המערכת שלי היא מאוד פשוטה לשימוש, ובניגוד לNmap שדורש ידע מקדים על מנת להשתמש בה, במערכת שלי לא נדרש ידע מקדים, והשימוש בה פשוט וידידותי למשתמש.

3. תוצר סופי

3.1. תיאור הפרויקט

כאשר מנהל הרשת (המשתמש) מריץ את ה Admin – החלון הגרפי אשר בעצם משרת אותו, הוא יקבל תמונה של כל המחשבים הפועלים ברשת – אלו הנמצאים ב LAN, ואלו שבם הותקנו מראש Client אשר מאפשרים למקסם את יכולות השליטה והבקרה שלו.

כאשר יצפה בנתוני מחשב מסוים, מנהל הרשת יוכל לראות פרטים מגוונים עליו – כתובות, מערכת הפעלה, פרוטוקולים שרצים על המחשב, ובאיזה Domain רשת גולש כרגע אדם הנמצא ברשת המקומית. בנוסף, מנהל הרשת יוכל לקבוע לו אפליקציות בלתי חוקיות שבעת כניסה לאפליקציות אלו, יקבל התראה על כך.

המוצר מתוכנן בצורה כך שהניווט בתוכנה יהיה פשוט והגרפיקה ידידותית למשתמש ומאפשר הבנה מעמיקה של הנתונים המוצגים.

3.2. דרישות ואילוצי פתרון

למערכת ישנם מספר אילוצים.

ראשית, נדרשת התקנה של הספרייה Scapy במחשב עליו רץ השרת. זאת מכיוון שהשרת מבצע את כל שליחת ההודעות הדורשות לעיתים את הספרייה.

בנוסף, על מנת לנצל את הפונקציונאליות המלאה של המערכת, יש להתקין לקוח בכל אחד ממחשבי הרשת המקומית. בנוסף, צריכים להתאים את גרסת ה python במחשב השרת לגרסת ה windows (windows 10 – python 2.7, אחרת, python 2.6), מכיוון ש Scapy ל python ל windows תלוי בגרסת ה windows שמותקן על המחשב.

תוכנות ה Client, Admin וה Server רצות אך ורק על מערכת הפעלה מסוג Windows 7 ומעלה.

3.3. אלגוריתמים עיקריים

- מיפוי מערכת ההפעלה של מחשב מרחוק – ע"י Active Fingerprinting:

ע"י יצירת פקטות מסוימות כגון: ICMP Echo Request with code 0, ICMP Timestamp Broadcast, שליחתן למחשב היעד, וניתוח התגובות המתקבלות על ידו על פי פרמטרים קבועים מראש (לדוג' TTL בפקטת התגובה), המערכת מסוגלת לזהות את מערכת ההפעלה של המחשב. הדבר מתאפשר מכיוון שכל מערכת הפעלה מגיבה באופן שונה ממערכת הפעלה אחרת להודעות מסוימות. שיטה זו נקראת גם Fingerprinting. לצורך העניין, windows תגיב לפקטת ICMP Echo Request with code 0 עם פקטת תגובה בעלת קוד 0, אך מערכות ההפעלה האחרות יגיבו עם קוד שאינו שווה לאפס.

- מציאת מחשבים ברשת- ע"י הגדרת "טבלת ARP משלי":

בהתאם לכתובת ה IP של המחשב, וה mask שלו ניתן למצוא את כתובת הרשת. כך ניתן למצוא את כל המחשבים הפוטנציאלים היכולים להימצא ברשת. המערכת שולחת לכל המחשבים הפוטנציאלים ברשת פקטת ARP request ותוך כדי מאזינה לתגובות. כך, ניתן לדעת שמחשב אשר הגיב לבקשה, מחובר לרשת ולקחת מהתגובה את כתובת ה mac שלו. הדבר נותן יתרון מכיוון שפעמים רבות טבלת ה ARP אינה אמינה, וכך אין תלות על מידע מהטבלה.

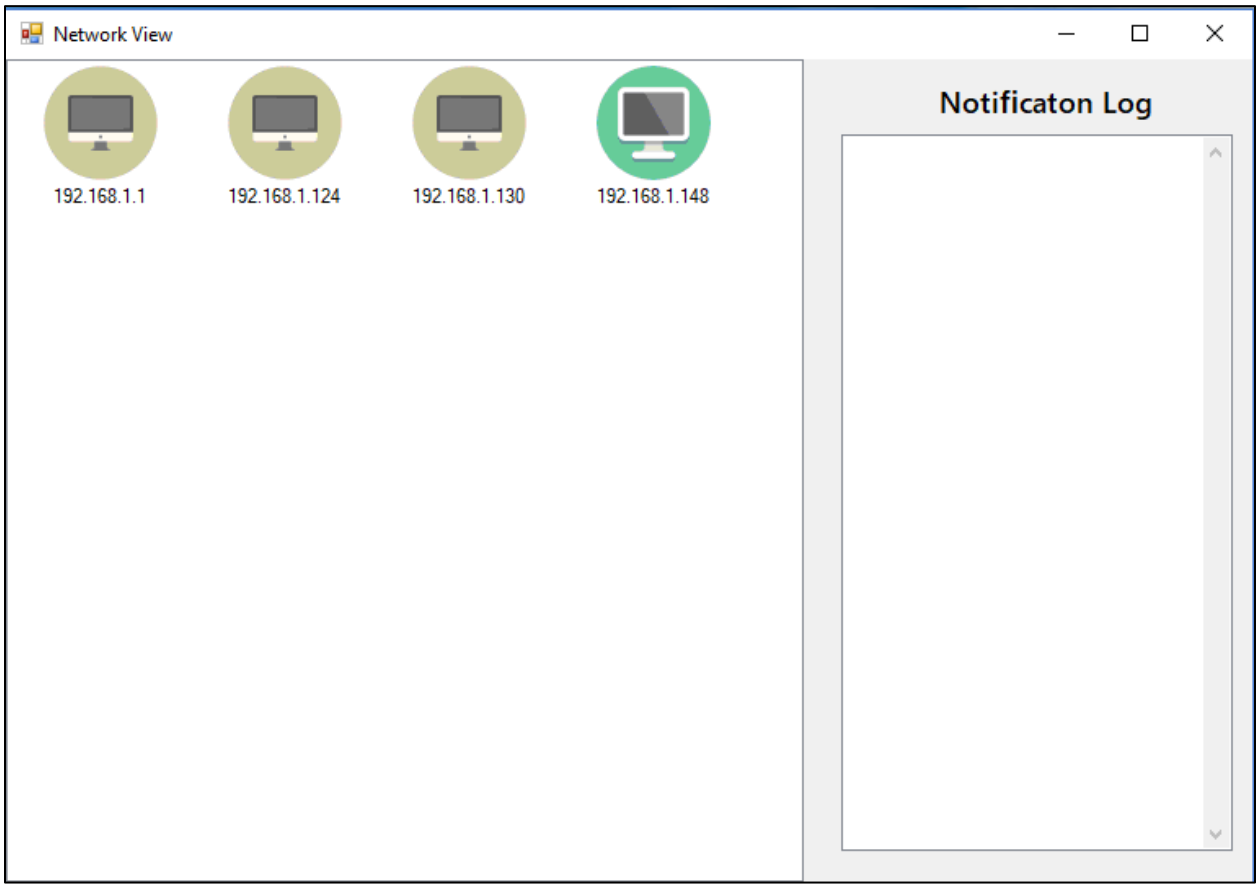
- קביעת תהליכים לא חוקיים (Black List) – ושמירתם במסד נתונים על הסרבר + קבלת התראה ל admin:

בעת קביעתו של ה admin על תהליכים בלתי חוקיים במחשב מסוים ברשת, אשר מותקן עליו לקוח, התוכנה תשלח לשרת הודעה מתאימה שם הוא ישמור אותה במבנה נתונים עם הפקודה הבלתי חוקית, וכתובת המחשב, ויעביר אותה ללקוח הרלוונטי. בעת ביצוע פעולה בלתי חוקית במחשב הנ"ל ישלח השרת ללקוח הודעה מתאימה, והוא יעביר לadmin התראה בנושא. ל admin יש אופציה לקבוע אפליקציות בלתי חוקיות, ו domain אסורים ללקוח לגשת אליהם.

3.4. ממשק משתמש

להלן מסכי המערכת העיקריים (כולם לקוחים מתוך ה Admin):

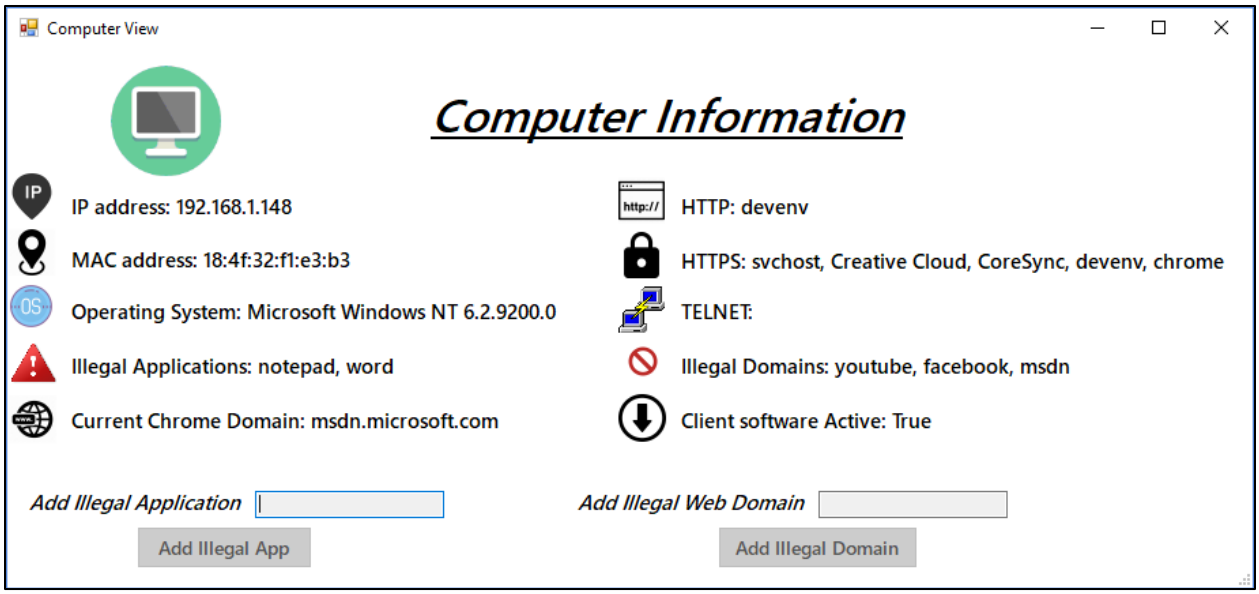
מסך ראשי – ראיית כל המחשבים ברשת:



זהו המסך ההתחלתי. ניתן לראות את כל המחשבים ברשת, המחולקים ל2 סוגים:

- מחשב עם תכנת client מותקנת עליו מראש – בצבע ירוק
- מחשב ללא תכנת client מותקנת עליו מראש – בצבע חום

מסך משני – ראיית מחשב ספציפי והפרטים עליו:

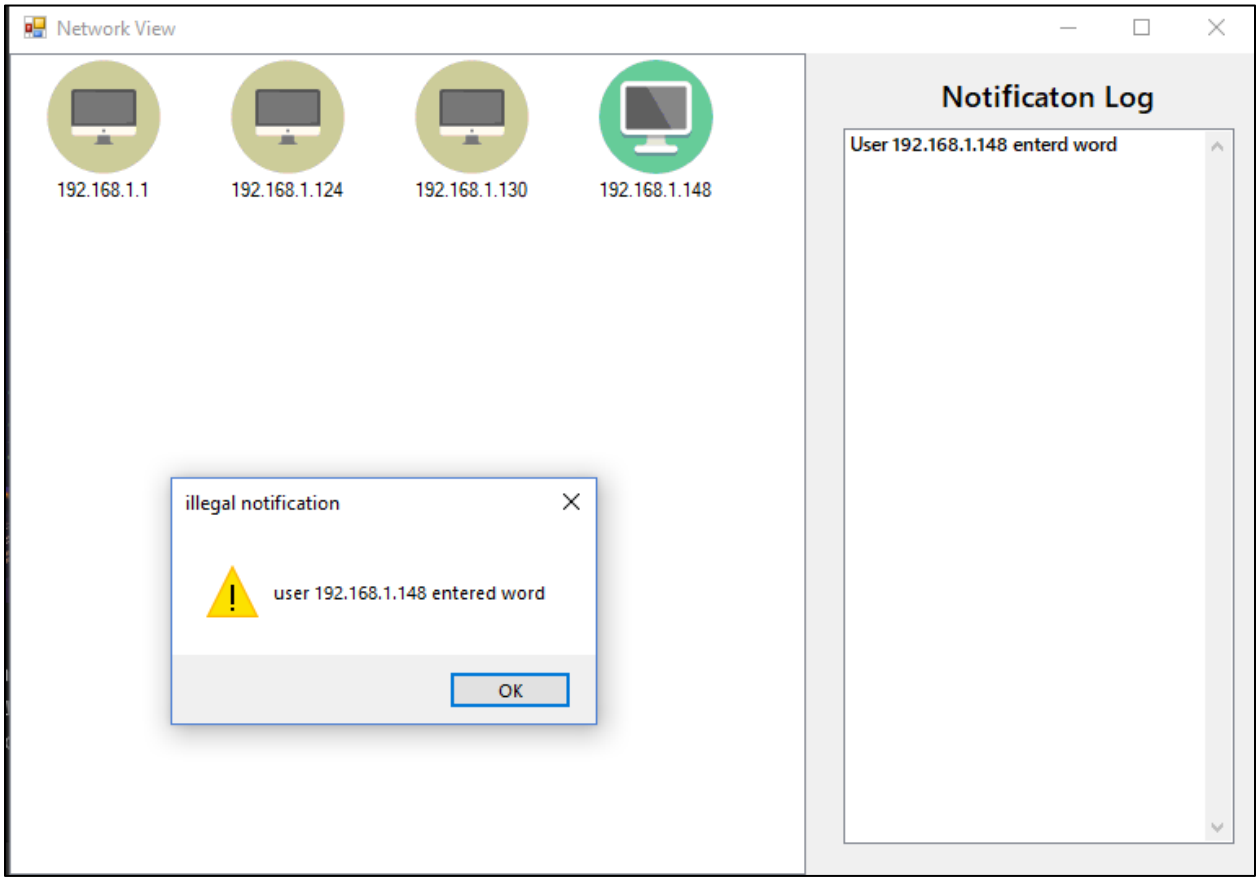


כאשר מקליקים במסך הראשי על מחשב מסויים, מגיעים למסך זה המאפשר למשתמש לראות המידע על המחשב, ולקבוע את האפליקציות/domain הלא חוקיים למחשב הזה.

הסבר על כל שדה:

- Ip address: The computer's IP address (logical address)
- MAC address: the computer's MAC address (physical address)
- Operating System: The computer's Operating System
- Illegal Applications: A list of Illegal Applications, defined by the admin, in the "Add Illegal Application" text box below
- Browser Domain: The computer's current Chrome's Tab Domain.
- HTTP: A list of Processes running on the computer, using HTTP protocol
- HTTPS: A list of Processes running on the computer, using HTTPS protocol
- Telnet: A list of Processes running on the computer, using Telnet protocol
- Illegal Domains: A list of Illegal Domains, defined by the admin, in the "Add Illegal Web Domain" text box below
- Client software Active: Does the computer have a client software pre-installed and is it currently active and running.

מסך בעת קבלת התראה:



בעת קבלת התראה על שימוש באפליקציה/Domain לא חוקי, שהוגדר ע"י ה admin, תתקבל התראה מסוג "Pop-Up" שתוצג למשתמש, בנוסף לעדכון הרשימה בצד ימין של המסך הראשי תחת הכותרת "Notification Log".

3.5. Use Cases עיקריים

	Admin	Server	Client
סריקת הרשת		מבצע את סריקת הרשת - שולח בקשות ARP למחשבים ברשת.	
			מגיב על הבקשות לשרת.
		שומר את המידע במבנה נתונים פנימי מתאים.	
	בכל מספר רגעים, שולח לשרת בקשה לשליחת מצב הרשת.		
		שולח ל admin את מצב הרשת.	
	קבלת המידע, והצגתו בצורה אינטראקטיבית למשתמש.		
קביעת הגדרות	שולח הודעה מתאימה לשרת בנוגע לקביעת התהליך הבלתי חוקי.		
		מקבל את המידע, שומר במבנה נתונים ומעביר ללקוח..	

			שומר את המידע במבנה נתונים משלו, ובמידה ומבצע פעולה זו שולח לסרבר הודעה מתאימה.
		מקבל התראה, ומעביר לAdmin.	
	מציג את ההתראה למשתמש.		
מציאת מערכת ההפעלה של המחשב ע"י AF		בעת קבלת תגובה מפקטת ARP והוספה למבנה הנתונים המתאימה, מתחיל עם כל מחשב ספציפית את תהליך התקשורת.	
			תקשורת רלוונטית בין שרת ללקוח.
		מציאת מערכת הפעלה והוספה של המערכת למבנה הנתונים הרלוונטי.	

4. תהליך כתיבת הפרויקט

4.1. תהליך הפרויקט

בתחילת תהליך כתיבת הפרויקט כתבתי שרת ולקוח בסיסיים, ללא מסך Admin. כל המידע היה מוצג בצורת console על הקליינט והשרת ללא הצגה גרפית. מרבית העבודה בתחילה הייתה בצד השרת בתחום ה Fingerprinting. מכיוון שהמידע באינטרנט מוגבל ונורא כללי בנושא זה, הייתי צריך לבצע מחקר עצמי רב, תוך התייעצות עם מומחים בתחום – שי, הטכנאי בבית הספר שלנו ותלמידים משנים עברו אשר התעסקו בתחומים דומים לשלי.

לאחר שסיימתי זאת, עברתי לפתח את הלקוח, תוך כדי יצירת Admin בסיסי. הלקוח ידע להוציא את מרבית הפונקציונאליות שלו כבר מלכתחילה, והקושי היה בלשלב את כל הפונקציות ל"בלוק" אחד גדול שיודע לתקשר אחד עם השני ופועל בסנכרון תואם. ברגע שהיה לי שרת ולקוח מספיקים התחלתי לפתח בעיקר מבחינה גרפית את ה Admin.

לבסוף, נשאר לשלב את שלוש הישויות למערכת אשר פועלת בתיאום וביעילות רבה.

בתהליך בנוסף ללמידה מעמיקה ומשמעותית על החומר שהפרויקט עוסק בו, למדתי המון על כתיבת פרויקט. יש שוני רב בין כתיבת תוכנה קטנה, לבין כתיבת מערכת גדולה ומורכבת. אני חש כי כתיבת עבודה זו תקנה לי ידע שימושי לעתידי.

4.2. אתגרים ואופציות שונות למימוש

במהלך כתיבת הפרויקט נתקלתי במספר אתגרים:

- OS Fingerprinting – נושא זה הוא נושא ללא יותר מדי מידע ספציפי באינטרנט, ולכן נאלצתי לבצע למידה ומחקר עצמי רב בנושא, בנוסף לניסוי וטעייה עד שמצאתי את הפתרון הנכון והמדויק ביותר.
- ייצור טבלת ARP עצמית – במהלך הפרויקט נתקלתי בבעיה מעניינת. משום מה לא כל המחשבים במעבדה תמיד הופיעו בטבלת ה ARP ולכן הייתי צריך לייצר נתונים אלו באופן עצמי. זאת ע"י חיקוי פעולת הטבלה, ושליחת פקטות ARP מסוג "who-has" (פקטות בקשה), ומילוי הנתונים במבנה נתונים במערכת.
  בתחילה, חשבתי על דרכים אחרות לבצע זאת. האופציה הראשונה שעלתה, היתה לגרום

לגרום לטבלה להתעדכן.

המחשב, בכל פעם שנדלק, ובכל כמה זמן, מעדכן את טבלת ה ARP שלו. הוא עושה זאת ע"י שליחת פקטות ARP לכתובת Broadcast ברשת וכל המחשבים עונים. חשבתי לעצמי כי אוכל לעשות זאת גם, והטבלה תתעדכן.

כאשר שלחתי פקטת Arp Broadcast Request גיליתי 2 דברים:

- למרות שהגיעו תשובות ממחשבים, הטבלה לא התעדכנה. אני סבור כי הדבר נובע מכך שמערכת ההפעלה מעדכנת את טבלת הARP שלה רק כאשר היא זו ששולחת את הפקטות בקשה, ולא כאשר פקטות מגיעות אל המחשב כתשובה למשתמש לדוגמה.
- בנוסף, גיליתי כי במעבדה בבית הספר לא כל המחשבים הגיבו לפקטת Broadcast. היו מחשבים שמסיבה כלשהי לא הגיבו לפקטת Broadcast, אך כאשר נשלחה אליהם הודעה עם כתובת ספציפית, הם כן ענו. הדבר כנראה נובע ממבנה הרשת בבית הספר, שמורכב ממספר Switchים, דבר המקשה כנראה על הפקטות להגיע לכלל המחשבים ברשת בפורמט של Broadcast.

לכן הפתרון הסופי, הוא שליחת פקטות ARP בעצמי, לכל מחשב "פוטנציאלי" ברשת (ב mask 255.255.255.0 על כתובת השרת), ותוך כדי הסנפה (האזנה) לתעבורת הרשת, ויצירת דמי – טבלת ARP משלי, אשר תשמש את המערכת לאורך כל זמן הריצה שלה.

5. מרכיבי פתרון

5.1. תיחום הפרויקט

הפרויקט עוסק בתחומים הבאים:

- תקשורת - שרת ,לקוח, תעבורה בעיקר ברמת IP וב ARP.
- תצוגה - שימוש בתצוגת winforms בצד ה admin.
- מבנה נתונים - שימוש ברשימות מקושרות ומערכים על מנת לשמור את המידע הנחוץ על המחשבים ברשת.
- ארכיטקטורת קוד- שימוש רב בפונקציות ומחלקות על מנת להקל על הגדרת המכונות השונות ברשת וכיוצ"ב.
- תיעוד- תיעוד אקטיבי בקוד, ושמירת גרסאות השונות בקוד. תיעוד כתוב (ספר פרויקט)

5.2. סביבת העבודה (טכנולוגיה)

שפות התכנות:

קוד השרת ייכתב ב python. זאת מכיוון ש Scapy עובד רק ב python והוא נחוץ לשרת.

את תוכנת ה admin והלקוח ארשום ב c# עם GUI של winforms בצד ה admin. ה GUI של c# נוח מאוד, פשוט לכתיבה ובעל רמה גבוהה יחסית.

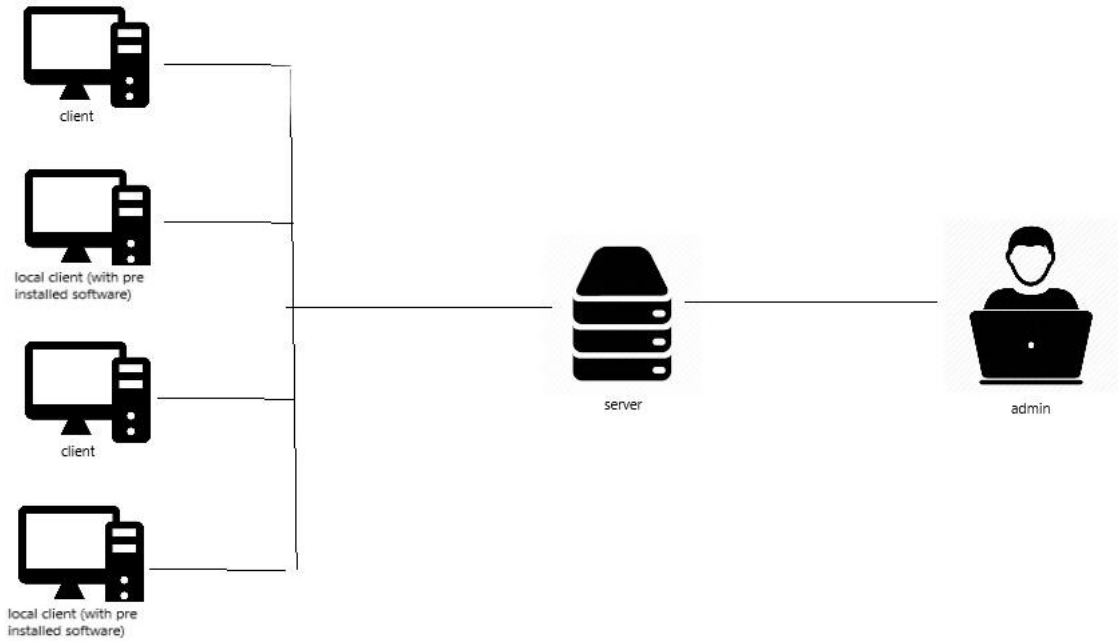
בנוסף, על מנת לבדוק ולעזור לבקר בזמן כתיבת הפרויקט איעזר ב Wireshark, תוכנת "רחרחן", על מנת לצפות בפעילות התוכנה ברשת.

סביבות פיתוח:

Visual studio לשפת c#

Pycharm לשפת python

5.3. מבט טופולוגי



המערכת בנויה מ3 חלקים – שרת, לקוח (אחד שנמצא ברשת המקומית וללא לקוח מותקן מראש, ואחד שמותקן עליו לקוח מראש) וAdmin.

5.4. מבט מודולרי

להלן החלקים העיקריים בפרוייקט:

שרת:

- Computer – מייצג אובייקט של מחשב קליינט (מקומי או לא). במחלקה הראשית כל המחשבים המחוברים מיוצגים כרשימת עצמים מסוג Computer.
- Main – המחלקה הראשית של השרת. בתוכה נמצאים כל הפעולות המפעילות את השרת ומתקשרות עם הלקוחות וה Admin.

לקוח:

- ClientSock – מטפל ב sockets של המערכת, ומבצע את כל הפעולות הנדרשות בנושא.
- Connection – נוגע לPorts. כל חיבור של Port מסויים במחשב מיוצג כאובייקט מסוג Connection. כך, ניתן לראות את תכונות החיבור המסויים כגון כתובת יעד, פורט יעד, פרוטוקול, שם התוכנה.
- Program – המחלקה הראשית של הלקוח. בתוכה נמצאים כל הפעולות המפעילות את הלקוח ומתקשרות עם השרת.

Admin:

- ClientSock – מטפל ב sockets של המערכת, ומבצע את כל הפעולות הנדרשות בנושא.
- Form1 – מייצג את מסך הפתיחה של ה Admin, מסך בו רואים את כל הקליינטים בתצוגה.
- Form2 – מייצג את המסך השני של ה Admin, מסך של כניסה לקליינט ספציפי.
- LoginForm – מייצג את המסך ההתחלתי של ה Admin, מסך ההתחברות לשרת. לאחר מילוי הפרטים, מועברים למסך הראשי – Form1.

5.5. פירוט מודלים עיקריים

להלן הפעולות המרכזיות והחשובות ביישום הפרויקט

:Server

Class	Function	Input / Output	Description
Main	get_network_computers	Input: None Output: None	Sends a “who has” Arp packets to all potential computers in the network
	arp_sniff	Input: None Output: None	While the Arp packets are sent, the function sniffs for responses from the computers. It sniffs only for ARP packets using a filter.
	analyze_packet	Input: Packet Output: None	Whenever a matching Arp packet is received, the function analyzes the packet and adds the sending computer to the database.
	admin_handler	Input: socket, address Output: None	When an Admin connects, this function handles it in a different thread. It communicates with it and sends him the relevant information.
	normal_client_handler	Input: socket, address Output: None	When a client connects, this function handles it in a different thread. It communicates with it and receives from him the relevant information.
Computer	OS_FingerPrinting	Input: None Output: prints the computer’s OS	When a computer is detected via Arp packets, the system identifies using this function its operating system with Active FingerPrinting

:Client

Class	Function	Input / Output	Description
ClientSock	Send	Input: the message output: None	the function sends (to the server) the message via sockets
	Recieve	Input: None Output: the received message	the function receives (from the server) the message via sockets
Program	Get_current_chrome_domain	Input: None output: None	the function returns the current domain the user is browsing on chrome, using AutomationElement class.
	running_monitor	Input: the server's socket output: None	the functions loops on the list of illegal apps the admin has set to him, while comparing it to the running processes on the computer. If an illegal app is currently running, the functions sends the server a notification, which is transferred to the Admin.
	Get_Process_port_status	Input: None Output: None	the function returns an array of current ports status, which then on another function is sent properly to the server, and transferred to the admin

:Admin

Class	Function	Input / Output	Description
Form1	Timer_Tick	input: None output: None	every 100 milliseconds, the admin sends the server a request to get a list of connected computers, it then receives the answer and shows it graphically.
	LstView_SelectedIndexChanged	input: None output: None	if the user pushes one of the computers listed on the screen, another Form with information about a specific computer is shown.
Form2	timer_Tick	input: None output: None	every 100 milliseconds, the admin sends the server a request to get the information about the specific client, it then receives the answer and shows it graphically.
	btn_add_illegal_Click	input: None output: None	whenever the user presses the button to add an illegal app to the list, the app is added and sent to the server a message notifying it, which is transferred to the client.

6. תסריטי בדיקה

6.1. דגשים בבדיקה

הדגשים שלי בבדיקה היו:

- חיבור והתנתקות של לקוחות מהשרת ללא באגים.
- מעבר הודעות מהלקוח לשרת, ומהשרת לAdmin בשלמותם.
- שינוי נתונים ב Admin בזמן אמת, ללא באגים או בעיות אחרות.
- בדיקת אמיתות הנתונים הנאספים ע"י הלקוח (אפליקציות רצות, תהליכים רצים, domain) וע"י השרת (המחשבים הנמצאים ברשת והמחשבים עם לקוח מותקן עליהם)

6.2. תסריטי בדיקה עיקריים

- התחברות של Admin / לקוח לשרת.
- התנתקות של Admin מהשרת.
- התנתקות של לקוח מהשרת – בזמן שה Admin צופה בנתונים שלו.
- בדיקת התראות – לקוח נכנס לאפליקציה / Domain שהוגדר לו מראש כבלתי חוקי
- בדיקת אמינות התהליכים הרצים – סגירה / פתיחה של תוכנות ובדיקה האם אכן התעדכנו ב Admin
- התחברות של לקוח, לאחר שנמצא המחשב דרך סריקת הרשת – עדכון הנתונים עליו. לאחר מכן ניתוק שלו בשנית, וחיבור שוב.

7. רפלקציה

7.1. לוח זמנים מוערך לניהול הפרויקט:

נובמבר	תשתית סריקה אקטיבית של הרשת, וצפייה בה.
דצמבר	סריקה חיצונית ומקומית של המחשבים ברשת.
ינואר	ניהול התראות DB וADMIN
פברואר	תצוגה.
מרץ	שיפור הקוד והוספת אפשרויות נוספות לשליטה בקשת. תיקון באגים ושיפור תצוגה.

7.2. אתגרים ותרומה אישית

תהליך הכתיבה היה תהליך מאתגר ומעניין, תהליך ממושך שדורש זמן והשקעה רבה. הכתיבה עזרה לי להתמקד בתוצר הסופי, והפירוק שלו לחלקים קטנים יותר, דבר שעזר לי בהבנתו ובתכנונו בצורה מיטבית ביותר.

במהלך תהליך הכתיבה, הייתי חושב על הפרויקט המון, וכאשר היה עולה לי רעיון חדש, גם אם באותו הרגע לא הייתי עובד על הפרויקט, כתבתי בפתק את הרעיון ומימשתי אותו כאשר עבדתי שוב על הפרויקט. כך, תמיד הייתי משפר את המערכת ופותר בעיות רבות שבתחילה נראו לי בלתי אפשריות.

קביעת לוחות הזמנים, והניסיון לעמוד בהם הכניסה אותי לקצב עבודה, אשר עזר לי לתכנן את הזמן ולייעל אותו.

העמידה בזמנים אינה קלה, אך מאפשרת תכנון נכון ומושכל של הזמן ובכך עוזרת לייעל את תהליך העבודה.

7.3. תובנות

כתיבת הפרויקט העשירה אותי המון בתחום המחשבים. היא עזרה לי לרכוש ניסיון בתכנות, אך בנוסף עזרה לי להכיר יותר לעומק את התחום בפועל ולא רק בתאוריה. הייתי צריך להתמודד עם אתגרים לא פשוטים במהלך הדרך, ולהתגבר עליהם. העבודה בעצמי לימדה אותי להכיר עוד תכונות על עצמי ולגלות שאני מסוגל לעשות דברים גם אם הם נראים בלתי אפשריים בתחילה.

בתחילת הדרך, התלבטתי זמן רב לגבי הנושא עליו אעשה את העבודה. כאשר החלטתי ללכת על כיוון זה, הייתי נורא לא בטוח כי אכן אצליח להגיע לתוצר מרשים מספיק, שעובד, ושמספק את המטרה שלו. כעת אני יכול לומר, כי אני מרוצה מאוד מהתוצר הסופי, וגאה בעצמי על ההתמדה והעבודה שהשקעתי במהלך הכתיבה.

8. הוראות התקנה ותפעול

8.1. תצורה ודרישות קדם

למחשב הAdmin:

- נדרש net framework.

מחשב הClient:

- נדרש net framework.
- מומלץ על ריצה ב10 Windows

מחשב הServer:

- נדרשים ספריות Python – scapy
- התאמה של מערכת ההפעלה לגרסת ה python (נובע מתוך האילוצים של ספריית ה scapy):
  - windows 8 ומעלה – python 2.7
  - windows 7 ומטה – python 2.6

8.2. הפעלה

תחילה להפעיל את השרת ולחכות להדפסת כתובת ה IP וה Port שלו. לאחר מכן, התחברות של ה Admin ובמידה ויש, גם של הלקוחות. לאחר מכן, המערכת מוכנה לריצה כרגיל.

9. ביבליוגרפיה

במהלך כתיבת הפרויקט הסתמכתי על מספר מקורות מידע:

- http://techgenix.com/operating-system-fingerprinting-packets-part1/
- http://resources.infosecinstitute.com/must-know-os-fingerprinting/
- https://www.digitalwhisper.co.il/files/Zines/0x35/DW53-1-NMap.pdf - Nmap digital whisper 2014 גליון 52 ,יולי
- https://yamakira.github.io/art-of-packet-crafting-with-scapy/network_recon/os_detection/index.html

10. נספחים

אין נספחים