

שם בית הספר: תיכון הרצוג

שם העבודה: IP Grabber

שם התלמיד: עידן רטיק

תעודת זהות: 216239939

שם המנחה: אופיר שביט

שם החלופה:

תאריך ההגשה: 24/05/25

Table of Contents

[כותרת 1 2](#_Toc198578522)

[כותרת 2 2](#_Toc198578523)

[**כותרת 3** 2](#_Toc198578524)

[מבוא 3](#_Toc198578525)

[ייזום 3](#_Toc198578526)

[**תיאור ראשוני של המערכת** 3](#_Toc198578527)

[**הגדרת יעדים ומטרות** 3](#_Toc198578528)

[**סקירת פתרונות קיימים** 3](#_Toc198578529)

[**סקירת טכנולוגיות בפרויקט** 4](#_Toc198578530)

[**סייגים והגבלות** 4](#_Toc198578531)

[**תיחום הפרוייקט** 4](#_Toc198578532)

[אפיון המערכת 4](#_Toc198578533)

[**יכולות ההורה** 5](#_Toc198578534)

[**פירוט בדיקות** 5](#_Toc198578535)

[**לו״ז** 7](#_Toc198578536)

[ידע 8](#_Toc198578537)

# **כותרת 1**

## **כותרת 2**

### **כותרת 3**

טקסט רגיל

# **מבוא**

## **ייזום**

### **תיאור ראשוני של המערכת**

אני מפתח כלי IP Grabber שמאפשר לאסוף ולאחסן כתובות IP של משתמשים שמבקרים בדף אינטרנט מסוים. הכלי יאפשר גם איסוף מידע נוסף כמו פרטי מערכת הפעלה, דפדפן, ומיקום גיאוגרפי משוער. המוצר המוגמר יהיה פלטפורמה שמאפשרת יצירה ושימוש ב-IP Grabber באמצעות ממשק פשוט ונוח למשתמש.

בחרתי בפרויקט זה כדי להעמיק את הידע שלי באבטחת תקשורת, בטכניקות מתקדמות ברשתות, ופיתוח. הפרויקט משלב היבטים של רשתות, מערכות הפעלה, פיתוח צד-שרת וצד-לקוח, וניתוח נתונים.

רוב הקשיים שאני צופה בהכנה של המערכת יהיו קשורים, לטכניקות מתקדמות ברשתות כגון ARP spoofing, וDNS spoofing, בתקשורת בין פרוססים

המערכת מיועדת להורים, שמעוניינים להשגיח על גישת ילדיהם לאתרים מסוימים. בעזרת שימוש במוצר, יוכלו ההורים לבדוק בצורה מדויקת מתי ומי נכנס לאתרים מסוימים.

### **הגדרת יעדים ומטרות**

* יצירת כלי פשוט וידידותי למשתמש לאיסוף כתובות IP, ללא פעולה אקטיבית של המתחברים לאתר.
* אפשרות לאחסון והצגת הכתובות שנאספו בממשק מאורגן
* יצירת גרפים וניתוחים סטטיסטיים של הנתונים שנאספו
* אפשרות לייצא את הנתונים בפורמטים שונים

חסכונות:

חיסכון בזמן באיתור והתמודדות עם גישות חשודות

ייעול תהליכי ניטור אבטחה

הפחתת הצורך בכלים יקרים יותר לניטור תעבורה

### **סקירת פתרונות קיימים**

Grabify - כלי פופולרי ליצירת קישורים למעקב. יתרונות: קל לשימוש. חסרונות: פונקציונליות מוגבלת.

IPLogger - מציע שירותי איסוף IP. יתרונות: מגוון אפשרויות. חסרונות: ממשק מיושן.

IP Tracker Online - כלי חינמי. יתרונות: זמינות גבוהה. חסרונות: אפשרויות אחסון מוגבלות.

.

### **סקירת טכנולוגיות בפרויקט**

צד לקוח: HTML, FLASK, CSS, SOCKET/TCP

צד שרת: HTTP,SCAPY, DNS, ARP, Multiprocessing ,Signal ,socket/TCP

נתונים: Json

### **סייגים והגבלות**

על המשתמש להכניס בצורה מפורשת את הURL שדרכו מתבצע המעקב.

חלק מהטכנולוגיות לא יעבדו כראוי אם יש רשת עם הרבה הגנות, או אם משתמשים בדפדפן זהיר.

### **תיחום הפרוייקט**

הפרויקט עוסק במספר תחומים הקשורים לרשתות ומערכות הפעלה -

רשתות:

תקשורת בין לקוח לשרת בעזרת SOCKET

שימוש בפרוטוקול ARP כדי לנתב את בקשות הDNS אל השרת.

שימוש בפרוטוקול DNS כדי לגרום לURL שלא קיימים לעבוד.

שימוש בScapy כדי לשלוח פקטות.

שימוש בFLASK למימוש שרת http בשביל gui בסיסי

שימוש בפרוטוקול http כדי לבצע redirect.

מערכות הפעלה:

שימוש בmultiprocessing בצד השרת.

שימוש בPIPE כדי לדבר בין הפרוססים השונים (IPC)  
שימוש במערכת קבצים לשמירת מידע.

שימוש בsignal כדי לצאת מprocess בצורה נקייה.

הפרויקט יתמקד באיסוף מידע באופן פסיבי בלבד (כלומר, מידע שנשלח כחלק מהבקשות הרגילות) ולא יכלול איסוף אקטיבי או חדירה למערכות.

## **אפיון המערכת**

המערכת משמשת בתור כלי של ההורה, כדי לנטר ולבדוק מתי כל ילד התחבר לאתרים מסוימים.

השרת יחזיק אצלו בצורה לא מוצפנת מאגר של קישורים. המאגר יקשר כל קישור מומצא לקישור אמיתי.

ההורה יתחבר אל המערכת ושם יוצגו לו האופציות, אם להתחבר בתור משתמש רשום או להירשם.

לאחר שיתחבר בתור משתמש, יוכל לנהל בצורה דינמית את מאגר הקישורים שמוחזק בצד הלקוח.

כאשר מישהו ישתמש בקישור המומצא, המערכת תתעד את זמן הכניסה, כתובת ה-IP ממנה בוצעה הגישה, ופרטים נוספים. בנוסף, המערכת תנתב את הילד לקישור האמיתי מבלי שירגיש שהוא מנוטר.

כאשר יתחבר ההורה אל המערכת, תהיה לו אופציה לעבור לעמוד שבו יוכל לראות את כל הכניסות שבוצעו, למיין אותם לפי שעה, לפי מחשב, ועוד. ההורה יוכל לראות גרפים וסטטיסטיקות על תדירות השימוש, שעות הגלישה המועדפות, והתפלגות הגלישה בין המשתמשים השונים, ההורה יכול לייצא את הנתונים הללו לאקסל.

מערכת זו לא תדרוש שום דבר מהילד, חוץ מלהכניס את הקישור המזויף שניתן לו.

### **יכולות ההורה**

* ליצור חשבון.
* להתחבר לחשבון.
* לבקש את הקישור האמיתי, של קישור מסוים, מהמאגר.
* להכניס אל המאגר קישור חדש.
* להוציא מהמאגר קישור קיים.
* לבקש את כל הכניסות לאתר מסוים.
* לעבור בין התפריטים.

### **פירוט בדיקות**

|  |  |  |  |
| --- | --- | --- | --- |
| מס' | יכולת לבדיקה | תיאור הבדיקה | איך בודקים? |
| 1 | יצירת משתמש (הורה) | בדיקת תהליך הרשמה של משתמש חדש למערכת | מריצים את אפליקציית הממשק, נכנסים לדף ההרשמה וממלאים את הטופס עם פרטים תקינים. מוודאים שהמשתמש נוצר במסד הנתונים ושאפשר להתחבר עם הפרטים החדשים. יש לבדוק גם מקרי קצה: הכנסת פרטים לא תקינים, שם משתמש קיים, וסיסמה חלשה. |
| 2 | התחברות משתמש | בדיקת התחברות משתמש קיים | יוצרים משתמש מראש, מתנתקים מהמערכת ומנסים להתחבר מחדש עם שם המשתמש והסיסמה. בודקים שהמערכת מזהה נכון את פרטי ההתחברות ומציגה את ממשק הניהול. יש לבדוק גם התחברות עם פרטים שגויים. |
| 3 | יצירת קישור מזויף | בדיקת יצירת קישור מזויף וקישורו לאתר אמיתי | נכנסים לממשק ניהול הקישורים, יוצרים קישור חדש עם שם ייחודי והאתר האמיתי (למשל google.com). בודקים שהקישור נוצר במסד הנתונים ומוצג ברשימת הקישורים בממשק. |
| 4 | ARP Spoofing | בדיקת יכולת התחזות לנתב | מפעילים את מנגנון ה-ARP Spoofing כאשר Wireshark פועל על מחשב אחר ברשת. בודקים ב-Wireshark שחבילות ה-ARP המזויפות נשלחות לרשת, ושהמחשבים ברשת מעדכנים את טבלאות ה-ARP שלהם להצביע על ה-MAC של השרת שלנו במקום זה של הנתב. |
| 5 | יירוט בקשות DNS | בדיקת זיהוי ויירוט בקשות DNS לדומיינים מזויפים | מפעילים את מנגנון יירוט ה-DNS ואת Wireshark. מהמחשב של משתמש הקצה, מנסים לגשת לדומיין מזויף שיצרנו. בודקים ב-Wireshark שהשרת שלנו מספק תשובת DNS מזויפת המפנה ל-127.0.0.1 או לכתובת IP של השרת שלנו. |
| 6 | ניהול HTTP Server | בדיקת שרת ה-HTTP שמקבל את הפניות | מפעילים את שרת ה-HTTP ומוודאים שהוא עולה ללא שגיאות. מבצעים בקשות HTTP מפוזרות לשרת וצופים בלוגים כדי לוודא שהשרת מקבל אותן כראוי. בודקים את זמני התגובה תחת עומס על ידי יצירת מספר בקשות במקביל. |
| 7 | הפניה לאתר אמיתי | בדיקת הפניית משתמש מהקישור המזויף לאמיתי | לאחר הפעלת כל המנגנונים, ניגשים לקישור מזויף מדפדפן משתמש הקצה. בודקים שהדפדפן מופנה מיד לאתר האמיתי, וצופים ב-Developer Tools של הדפדפן כדי לוודא שמתקבלת הפניית 302 תקינה. |
| 8 | תיעוד כניסות | בדיקת תיעוד מידע על כניסות לקישורים מזויפים | מבצעים גישה לקישור מזויף ממספר מכשירים, דפדפנים ורשתות שונות. לאחר מכן, בודקים במסד הנתונים שכל הכניסות תועדו עם הפרטים המתאימים: כתובת IP, זמן הכניסה, סוג הדפדפן והמכשיר. |
| 9 | הצגת נתונים בממשק הורה | בדיקת הצגת סטטיסטיקות ונתוני כניסה | לאחר צבירת מספר כניסות במערכת, נכנסים לממשק ההורה ובודקים שהנתונים מוצגים נכון בגרפים ובטבלאות. בודקים את העדכון בזמן אמת על ידי פתיחת הממשק במקביל לביצוע כניסות חדשות. |
| 10 | תמיכה בריבוי משתמשים | בדיקת תמיכה במספר הורים ומספר ילדים | יוצרים מספר חשבונות משתמשים (הורים) ולכל אחד מהם מספר פרופילים של ילדים. בודקים שכל משתמש רואה רק את הקישורים והנתונים השייכים אליו, ולא את אלו של משתמשים אחרים. |
| 11 | ביצועי MITM | בדיקת יכולת Man-in-the-Middle | מפעילים את מנגנון ה-MITM ובמקביל מבצעים פעולות גלישה שונות ברשת. בודקים בעזרת Wireshark שהתעבורה עוברת דרך השרת שלנו, ושאין עיכובים משמעותיים או שגיאות תקשורת. |
| 12 | זיהוי DNS המזויף | בדיקת יכולת זיהוי דומיין מזויף | מפעילים את מנגנון יירוט ה-DNS ושולחים בקשות DNS הן לדומיינים מזויפים והן לדומיינים לגיטימיים. בודקים שהמערכת מזהה ומטפלת רק בדומיינים המזויפים ולא מפריעה לתעבורה הרגילה. |
| 13 | ייצוא נתונים | בדיקת ייצוא נתוני שימוש לפורמטים שונים | צוברים כמות מסוימת של נתוני כניסה במערכת, ואז מנסים לייצא אותם בפורמטים שונים (Excel, PDF). פותחים את הקבצים המיוצאים ובודקים שכל הנתונים הרלוונטיים נמצאים בהם ומוצגים נכון. |
| 14 | מולטי-פרוססינג | בדיקת פעולת המערכת במספר תהליכים במקביל | מפעילים את כל רכיבי המערכת במקביל (ARP Spoofing, יירוט DNS, שרת HTTP, ממשק ניהול). בודקים את צריכת המשאבים של המערכת ומוודאים שכל התהליכים מתקשרים ביניהם כראוי. מבצעים פעולות במקביל ובודקים שאין התנגשויות או חסימות. |
| 15 | עמידות בתקלות | בדיקת התנהגות המערכת במקרה של תקלות | מדמים מצבי תקלה שונים: ניתוק רשת פתאומי, כיבוי אחד התהליכים, ניסיון גישה לדומיין לא קיים. בודקים שהמערכת מתאוששת או מטפלת בתקלות בצורה הולמת ולא קורסת. |
| 16 | אבטחת מידע | בדיקת אבטחת המידע הנאסף | מנסים לגשת למידע השמור ללא הרשאות מתאימות. בודקים שתקשורת בין הלקוח לשרת מאובטחת. בודקים חסינות מפני התקפות נפוצות כמו SQL Injection או XSS. |
| 17 | פעולה ברשתות שונות | בדיקת פעולת המערכת ברשתות בעלות תצורות שונות | מפעילים את המערכת ברשתות בעלות תצורות שונות: רשת ביתית רגילה, רשת עם נתב מתקדם, רשת עם הגנות אבטחה. בודקים שהמערכת פועלת כראוי בכל הסביבות. |

### **לו״ז**

|  |  |
| --- | --- |
| **תאריך** | **משימה** |
| 31.11 | אפיון בסיסי של הפרוייקט |
| 1.1 | הוכחות יכולות בסיסיות (ARP spoofing, dns poisoning) |
| 3.1 | מימוש תקשורת שרת לקוח עם ממשק טקטסואלי |
| 31.3 | חיבור התקשורת עם כל היכולות |
| 10.4 | הכנת ממשק גרפי |
| 31.4 | סיום הכנת קוד כולל בדיקת באגים |

# **ידע**

## **יכולות בצד הלקוח**

|  |  |  |  |
| --- | --- | --- | --- |
| **שם** | **מהות** | **יכולות הדרושות למימוש היכולת** | **אובייקטים נחוצים** |
| יצירת משתמש | שליחת לשרת הודעה מוצפנת של יצירת משתמש עם הפרטים הנדרשים. | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר. |
| התחברות למשתמש | שליחת הודעה מוצפנת לשרת של התחברות למשתמש. | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר. |
| בקשת מידע מהשרת | שליחת לשרת הודעה מוצפנת של לפי הפרוטוקול | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר, מאגר קישורים. |
| הוספה של קישור למאגר | שליחת לשרת הודעה מוצפנת של לפי הפרוטוקול | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר, מאגר קישורים. |
| מחיקת קישור מהמאגר | שליחת לשרת הודעה מוצפנת של לפי הפרוטוקול | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר, מאגר קישורים. |
| בקשה של פירוט התחברויות לאתר מסוים | שליחת לשרת הודעה מוצפנת של לפי הפרוטוקול | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר, מאגר קישורים. |
| בקשת הקישור האמיתי התואם לקישור מזויף מסוים. | שליחת לשרת הודעה מוצפנת של לפי הפרוטוקול | שליחת הודעה לשרת, בניית הודעה לפי פרוטוקול, קבלת קלט משתמש | תקשורת מוצפנת עם השרת, סביבת משתמש גרפית, מאגר משתמשים, טיפול בשגיאות שהשרת מחזיר, מאגר קישורים. |
| החלפת מפתחות | החלפת מפתחות AES בעזרת RSA | תקשורת בין שרת ללקוח | עוטף לסוקט, מפתח RSA |
| הצפנה ופענוח מידע | הצפנה ופענוח מידע שנשלח או מגיע מהשרת בעזרת AES | תקשורת בין שרת ללקוח, החלפת מפתחות | עוטף לסוקט, מפתח AES |
| שליחת הודעה מוצפנת לשרת | שילוב בין יכולת הצפנת המידע לבין יכולת שליחת הודעה לשרת. | שליחת הודעה לשרת, הצפנת מידע | עוטף לסוקט, מפתח AES |
| בניית הודעה לפי פרוטוקול | בניית הודעה לפי הפרוטקול, בהתאם לקלט המשתמש. |  | עוטף לסוקט. |
| שליחת הודעה לשרת |  | יצירת תקשורת עם השרת | עוטף לסוקט. |
| יצירת תקשורת עם השרת. | Connect |  | עוטף לסוקט. |
| קבלת קלט משתמש | שימוש בממשק גרפי כדי לקבל ולנתח קלט משתמש. | הצגה ושימוש בממשק גרפי | ממשק גרפי. |
| הצגה ושימוש בממשק גרפי. | הצגת כל התפריטים, מעבר בינהם, והזנת נתונים. |  | Flask |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |