

**פרויקט גמר הגנת הסייבר**

נושא העבודה: **אנטי וירוס**

שם תלמיד: **אוהד גיפס**

ת.ז תלמיד: **215426883**

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

שם המנחה:הילה גורן ברנע

מועד הגשה:27/05/2024



תוכן

[מבוא 3](#_Toc168180793)

[נושא העבודה 3](#_Toc168180794)

[מטרות מרכזיות 3](#_Toc168180795)

[בחירת הנושא 4](#_Toc168180796)

[קישור לחומר הנלמד 4](#_Toc168180797)

[תיאוריה 5](#_Toc168180798)

[התיאוריה של העבודה 5](#_Toc168180799)

[מוצרים קיימים 11](#_Toc168180800)

[תוצר סופי 13](#_Toc168180801)

[תיאור הפרויקט 13](#_Toc168180802)

[אלגוריתמיים עיקריים 15](#_Toc168180803)

[דרישות ומגבלות מערכת 17](#_Toc168180804)

[אבטחה 17](#_Toc168180805)

[ממשק משתמש 18](#_Toc168180806)

[תרחישים עיקרים 23](#_Toc168180807)

[נספחים 24](#_Toc168180808)

# **מבוא**

### נושא העבודה

אנטי וירוס – זוהי תוכנה שאחראית על זיהוי וירוסים ונוזקות במחשב ולהזהיר על כך למשתמש. בנוסף, תבצע פעולות לטיפול בווירוסים או פעולות למניעת וירוסים. התוכנה משתמש במגוון של שיטות שונות כמו: זיהוי חתימות, למידת התנהגות, ניתוח בארגז חול, וכדומה. כדי למצוא את כל הקבצים החשודים. כיום בגלל התפתחות הרבה של הווירוסים מוצאים עוד ועוד שיטות כדי לזהות אותם.

האנטי וירוס אותו עשיתי מורכב משני סוגים של זיהוי: זיהוי לפי חתימה וזיהוי לפי התנהגות של קבצים באופן סטטי. כרגע הספקתי לבצע זיהוי חתימות של HASHשנקרא 5MD. והתנהגות של הקבצים בודקת קבצים מסוג EXE במערכת הפעלה ווינדוס.

### מטרות מרכזיות

מטרות היישומיות של הפרויקט הן:

* Virus Signature Detection
* Windows Malware Detection (with machine learning)
* טיפול בקבצים חשודים
* ממשק משתמש GUI

מטרות אישיות הן:

* ללמוד תכנות ב- C++ - עד כה היה לי ניסיון רק בלתכנת רק בפייתון, בסי שארפ באסמבלי. ורציתי לעבודה הזו ללמוד גם שפה חדשה. בקיץ התחלתי ללמוד את הבסיס לה והמשכתי ללמוד אותה תוך כדי עבודה על הפרויקט.
* מימוש למידת מכונה – זהו תחום ששמעתי עליו הרבה בשנתיים האחרונות ורציתי לנסות לממש דבר כזה.
* יצירת תכונה הבנויה מכמה שפות – היום הרבה אפליקציות בשוק בנויות מכמה שפות תכנות וגם רציתי להתנסות בעבודה עם DLL לכן חשבתי שאני אלמד יותר ככה.

### בחירת הנושא

שחשבתי על נושא לעבודה רציתי משהו שונה ומאתגר מעבודות הקודמות שעשיתי. באסמבלי עשיתי משחק ובשנה שעברה עשינו טורנט ורציתי שהנושא היה קשור לסייבר וככה אני גם אלמד יותר על הנושא. לכן בחרתי לעשות אנטי וירוס שזה אפליקציה למחשב שמתעסקת בשמירה על ביטחון המחשב מפני דברים זדוניים. בנוסף, אנטי וירוס זוהי תוכנה לא מוגדרת לגמרי וככה אפשר לעשות משהו שלא רק לממש אלא גם לחשוב איך לעשות ומה לעשו ת באנטי וירוס שאני רוצה לעשות וכך גם לקבל החלטות לפי כמות הזמן שיש לעשות את הפרויקט. לכן בגלל שזה משהו שעוד לא עשיתי ושהו יחסית פתוח לאינטרפרטציות שונות בחרתי לעשות את זה.

### קישור לחומר הנלמד

אנטי וירוס מתקשר לכל הנושא של סייבר ואבטחת מידע שלמדנו במהלך השנתיים האחרונות. למדנו במהלך השנתיים האחרונות כל מיני סוגים של התקפות ופרצות שעלולים לקראות ואנטי וירוס אחראי להגן על המחשב מפני חלק מפרצות אלו.

בנוסף, למדנו תכנות ועבודה בפייתון וגם עבודה עם סייר הקבצים. בפרויקט שלי אני משתמש הרבה מהנלמד בתחום הזה. כמו: threads, sqlite.

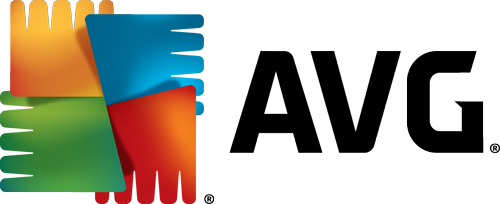
לכן, אני חושב שיש קשר להרבה מן הנלמד לחומר שלמדנו במגוון תחומי הסייבר וגם לימד עצמאית של חלק מן הדברים.

# תיאוריה

### התיאוריה של העבודה

#### **אנטי וירוס**

אנטי וירוס זוהי תוכנת אבטחה המיועדת למנוע, לזהות ולטפל בווירוסים ותוכנות זדוניות במחשב. היא משתמשת בשיטות שונות על מנת למצוא את כל הקבצים החשודים ולמגר את התקפות הווירוסים מכל הסוגים. אף אנטי וירוס לא יכול למנוע לגמרי את כל המתקפות. כמה דוגמאות לתוכנות פופולריות של אנטי וירוס:

תמונה שמכילה גרפיקה, גופן, עיצוב גרפי, לוגו

התיאור נוצר באופן אוטומטי

תמונה שמכילה גרפיקה, לוגו, גופן, עיצוב גרפי

התיאור נוצר באופן אוטומטי

היסטוריה

הווירוס הראשון הידוע הופיע לראשונה בשנת 1971 ונקרא Creeper virus. הווירוס היה תוכנת מחשב ניסיונית שנכתבה על ידי בוב תומאס מחברת ה- BBN (חברת מחקר ופיתוח בתחום הטכנולוגיה והפיתוח). הווירוס היה התועלת המחשב הראשונה בעולם והוא היה עובר ממקום למקום כאשר היו מפעילים את מערכת הפעלה TENEX (תוכנה שיצא לשוק בשנת 1969 על ידי BBN) באמצעות ARPANET (רשת אינטרנטית למחקר). לווירוס לא היו מטרות זדוניות רק מטרות מחקר.

לבסוף, האנטי וירוס הראשון שנוצר להתמודד עם וירוסים שונים היה AntiVir של Avira בשנת 1988 ובאותה שנה יצאו עוד מספר של אנטי וירוסים שונים כמו: VirusScan של McAfee וכו'. האנט וירוסים באותה תקופה לא התעדכנו בצורה אוטומטית ובשביל לעדכן אותם נאלצו להוריד אותם מדיסקים.



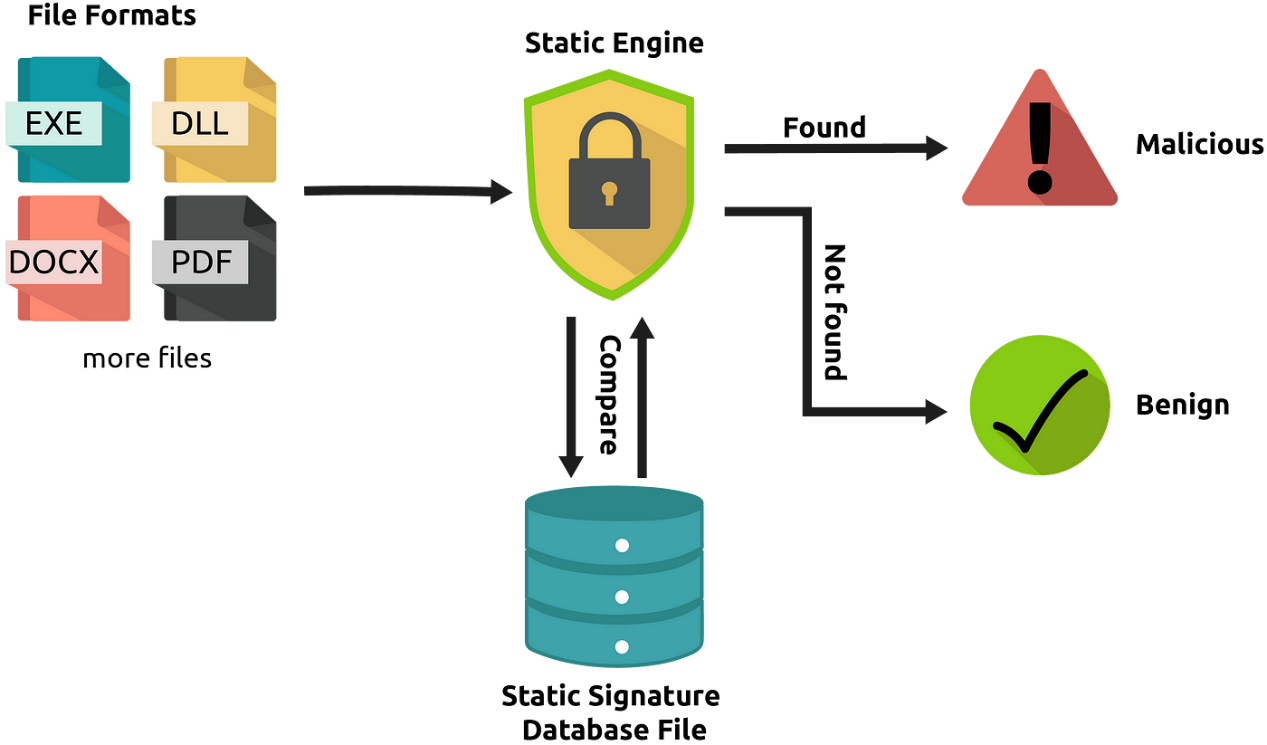
איך עובד אנטי וירוס?

רוב תוכנות האנטי וירוס פעולות ברקע של המחשב ועוקבות אחרי כל התנהגות חשודה ומבצעות סריקה על כל קובץ חדש שירד למחשב אך אפשר בגל האנטי וירוסים להפעיל סריקות בצורה ידנית גם כן. בנוסף, מונעים מהמשתמש לעשות פעולות העלולות לגרום לנזק במחשב או לחדירה של וירוסים כמו: פתיחת קבצים חשודים או הפעלת תוכנות לא מזוהות. לבסוף בקבצים החשודים מטפל בכך שמבודד אותם או מוחק אותם.

אנטי וירוס משתמש בשיטות שונות לזיהוי וירוסים שהתפתחו במהלך השנים:

#### זיהוי לפי חתימה – Virus Signature Detection

באמצעות זיהוי זה, התוכנית מחפשת וירוסים זדוניים על ידי זיהוי הדפוסים שלהם בקבצים וברשת. שיטה זו היא אחת השיטות הראשונות, הישירות והמבוססות ביותר לזיהוי תוכנות זדוניות. לכל וירוס יש מחרוזת נתונים ייחודית שידועה וניתן להשתמש בה כדי למצוא אותם.



יתרונות של זיהוי זה – קודם כל החתימות האלו יכולות להיות בכל מיני סוגים שונים של קבצים ולכן אפשר להשתמש בשיטה זו על כל סוגי הקבצים בלי דרישה למערכת הפעלה מסוימת או תנאים מסוימים. בנוסף באמצעות מסד נותנים מתאים ניתן לבצע סריקות מהירות של מספר קבצים רב.

חסרונות של זיהוי זה – כיום סוג זיהוי זה אינו מספיק לווירוסים של היום כי רבים מהם יודעים להתחמק מסריקות אלו. וגם ניתן לחזות מתקפות שהיו אבל וירוסים חדשים לא ניתן למצוא.

חתימות רבות של וירוסים כתובות בסוגי HASH שונים.

טבלת גיבוב (HASH)

זוהי פונקציה שלוקחת קלט באורך לא מוגבל ומשנה אותו לפלט באורך קבוע. טבלאות אלו משמשות גם למיון, הצפנה ואחסון . דוגמאות לטבלאות גיבוב: SHA-1, SHA-256 ו- MD5.

דוגמה לשימוש בטבלת HASH בשביל אחסון במסד נותנים


באנטי וירוס שלי אני משתמש בחתימות מסוג MD5:

MD5

פונקציית הגיבוב פותחה על ידי רונלד ריבסט ב-1991 והחליפה את MD4 מקור השם בא מצמד המילים תמצית מסרים. בשנת 1996 נמצא בו פגם באלגוריתם שלו ולכן הומלץ שלא להשתמש בו ולעבור לטבלת גיבוב אחרת (למרות שהפגם לא פגע ברמת ביטחונו) עד שבמהלך השנים נמצאו בו פרצות רבות.

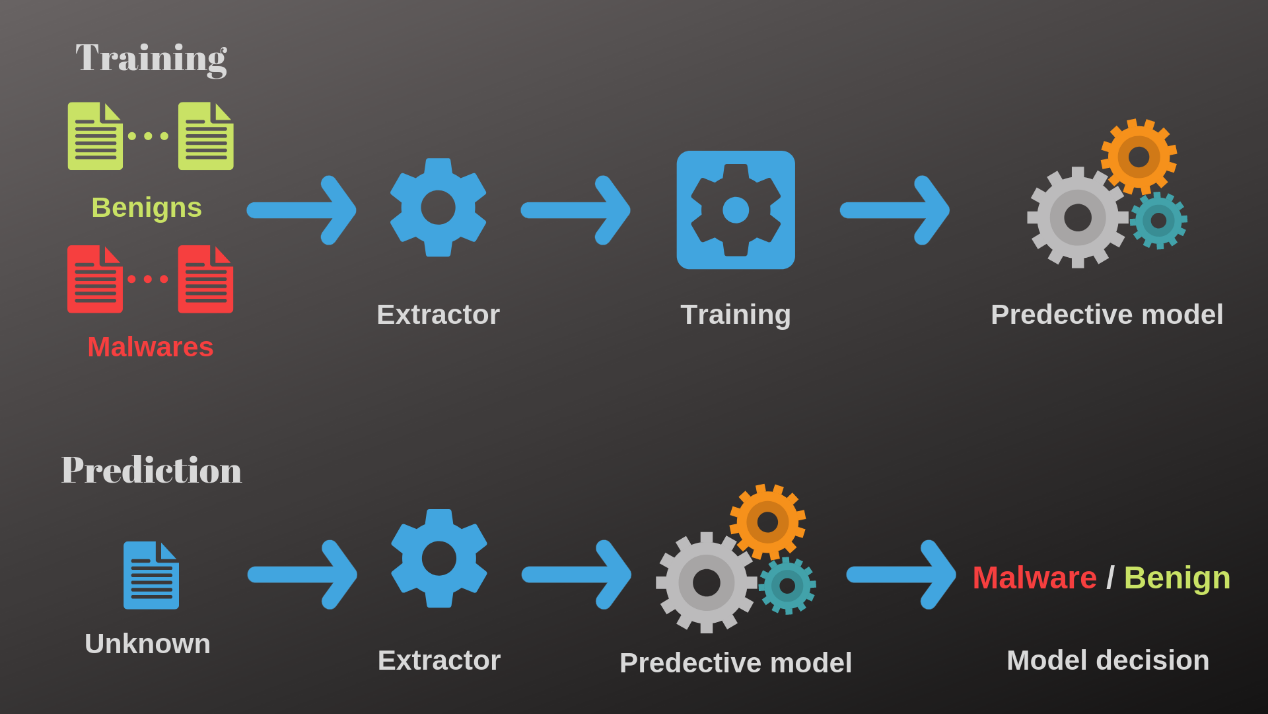
פונקציית הגיבוב לוקחת הודעה באורך לא מוגדר ו"מתמצת: אותו לאורך של 128 סיביות בבסיס הקסדצימלית (בסיס 16)

תמונה שמכילה טקסט, גופן, צילום מסך, לבן

התיאור נוצר באופן אוטומטיאיך הוא פועל? – הוא מקבל קלט ומחלק אותו תחליה לאורך של 512 סיביות אותם מעבד אחד אחרי השני. תחילה מתרחשת ריפוד כלומר גורמים לקלט להיות באורך של 448 מודולו 512 סיביות. לאחר מכן מתרחש הקידוד לפי סדר בתים קטן. וככה יכול להתחלק ל- 512 ללא שארית. לאחר מכן יש אתחול של הזיכרון, עיבוד הקלט עד של בסוף מחזיר פלט. לדוגמה:

#### Windows Malware Detection – זיהוי לפי התנהגות סטטית של קבצי הרצה של ווינדוס

בשיטה זו התוכנה אוספת מידע על קבצי הרצה מסוג EXE במערכת הפעלה ווינדוס כמו: לאיזה פעולות ממערכת ההפעלה הוא משתמש, באיזה קבצי DLL משתמש וגודל של חלקים בקובץ. לאחר מכן משתמש במערכי נתונים (Datasets) שבהם קיים מידע על איך קבצים רגילים מתנהגים ואיך קבצי וירוסים מתנהגים ואז משתמש באלגוריתם של למידת מכונה בשביל לזהות האם הקובץ מתנהג כמו וירוס או לא.



קבצי EXE – קבצי הרצה

קבצי הרצה הם קבצים אשר בתוכם נתונים המתורגמים לתוכנית על ידי המחשב והם כוללים ייצוג בינארי של פעולות מכונה בעבור מעבד מסוים דוגמה לקבצי הרצה:



למידת מכונה – Machine Learning

למידת מכונה הוא ענף בנושא הבינה מלאכותית ומדעי המחשב המתמקד בשימוש במידע ואלגוריתמים כדי לחקות איך בני אדם לומדים ולשפר את רמת דיוק של המכונה. הם משמשים כדי ליצור תחזיות, לסווג מידע, לאסוף נתונים ואף ליצור תוכן חדש.

כדי לאמן מכונה בצורה נכונה ומדויקת נדרשת בחירה של האלגוריתם הנכון למטרה רצויה וכמות מידע רבה כדי שהתוצאות היו מדויקות.

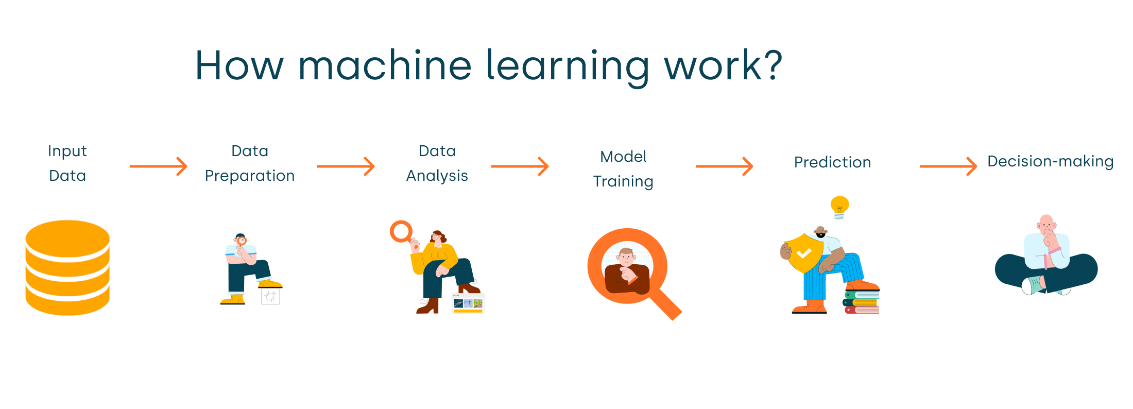
היסטוריה

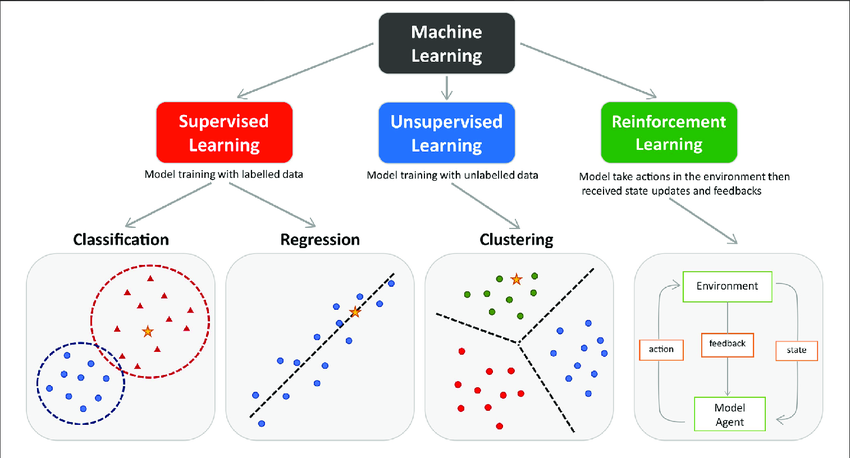
המושג למידת מכונה נולד לראשונה על ידי ארתור סמואל עובד של חברת IBMבשנת 1959. המודל המוקדם ביותר של למידת מכונה הוצג על ידו ואותו מודל היה אחרי לחשב סיכוי ניצחון בדמקה של כל צד.

איך זה עובד?

לכל למידת מכונה יש שלושה חלקים עיקריים:

1. **תהליך ההחלטה** – משתמש באלגוריתמים שונים של למידת מכונה כדי לבצע חיזוי או סיווג.
2. **פונקציית שגיאה** – פונקציית שגיאה מעריכה את החיזוי של המודל. אם יש דוגמאות ידועות הפונקציה משתמשת בהם לצורך דיוק המודל.
3. **תהליך האופטימיזציה** – כדי להתאים בין לנקודות הנתונים בערכת הנתונים, המודל מתאים את עצמו כדי להפחית את הפער בין הדוגמאות המוכרות להערכת המודל.





ישנם מספר סוגים של למידת מכונה:

1. **למידה מפוקחת** – המודל מתאים את המשקולות שלו עד שהוא מותאם כראוי. זה קורה כחלק מתהליכי אימות צולבים כדי להבטיח שהמודל ימנע התאמת יתר או תת התאמה. דוגמה לשיטות המשתמשות בלמידה מפוקחת.
2. **למידה לא מפוקחת** – מודל זה משתמש בשביל לנתח ולרכז מערכי נתונים לא מסומנים. סוג לאלגוריתם זה מגלה דפוסים נסתרים ללא התערבות אנושית. בגלל שהיא יכולה לגלות שונה ודומה בין מידע היא משמשת לניתוח מחקרי, לפיתוח אסטרטגיות שונות ועוד.
3. **למידת חיזוק** – הוא מודל שדומה ללמידה מפוקחת אך לא משתמש בנתונים בשביל להתאמן אלא בניסוי וטעיה.

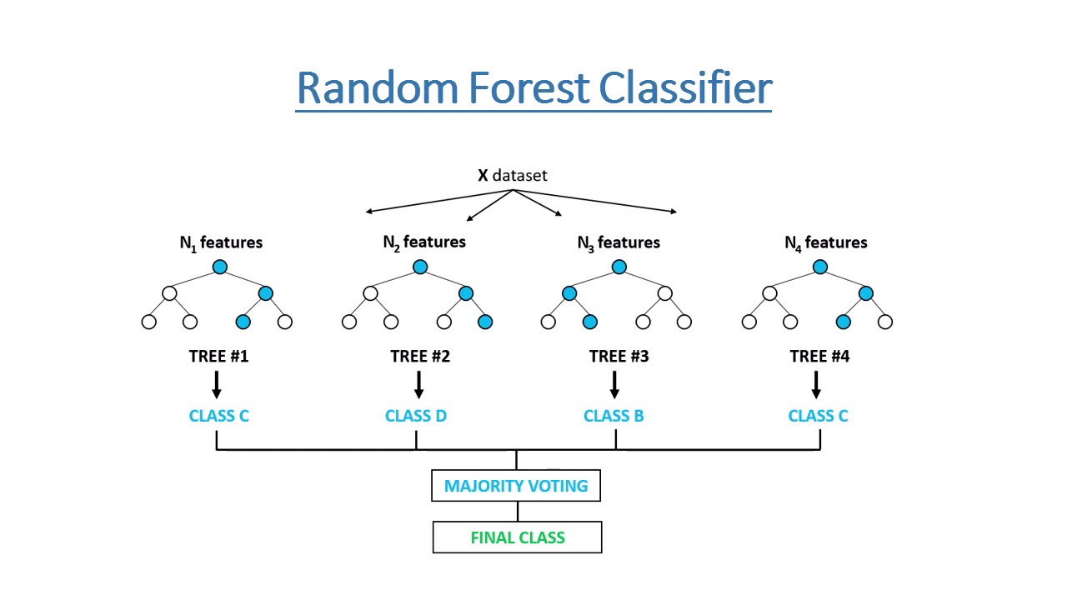
בעבודה אני משתמש בלמידת מכונה הנקראת: Random Forest Classifier

יער אקראי הוא אלגוריתם למידת מכונה נפוץ, הוא משלב פלט של עצי החלטה מרובים כדי להגיע לתוצאה אחת הוא מטפל גם בבעיות סיווג וגם רגרסיה.

עצי החלטה – כל עץ מתחיל משאלה בסיסית כמו: האם קיץ עכשיו? וככה גורם לפיצול של הנתונים כלומר יצירה של ענפים. העצים אחראים למצוא את הפיצול המתאים ביותר לנתונים שניתנו.

האלגוריתם של יער אקראי משתמש גם באקראיות בתכונות וגם ביצירת העצים ומזה יוצר תת קבוצה אקראית של תוכנות. וזהו הבדל בין עצי ההחלטה שהם בוחנים את כל הפיצולים האפשריים בעוד. יערות בוחרים רק תת קבוצות של התוכנות.

איך זה עובד? – לכל יער אקראי צריך להגדיר שלושה היפר פרמטרים עיקריים שצריך להגדיר אותם לפני האימון. גודל הצומת, מספר העצים ומספר התכונות שידגמו. משם ניתן להשתמש באלגוריתם כדי לפתור בעיות של סיווג.

העיר מורכב מאוסף של עצי החלטה. וכל עץ מורכב ממדגם של נתונים שנלקח ממערכת האימון ואוסף המידע הזה נקרא bootstrap. מתוך מדגם האימון שליש מופרש כנתוני מבחן הנקראים "מדגם מחוץ לתיק". לאחר מכאן שאר המידע נלקח באקראיות שם נקבעים תחזית בהתאם לסוג הבעיה על פי הצבעת רוב כלומר המשתנה השכיח ביותר. ולבסוף מאמתים את התחזית עם נתוני המבחן ששמרנו מקודם ומקבלים את התחזית הסופית.

### מוצרים קיימים

AVG - Anti-Virus Guard

זאת תוכנת אנטי וירוס ובטיחות אינטרנט התומכת בפלטפורמות המחשוב: ווינדוס, לינוקס וכו'. לראשונה יצאה לשוק בשנת 1991 ומציעה שירות חינמי להגנה מפני וירוסים. כיום שייכת לחברת Avast.

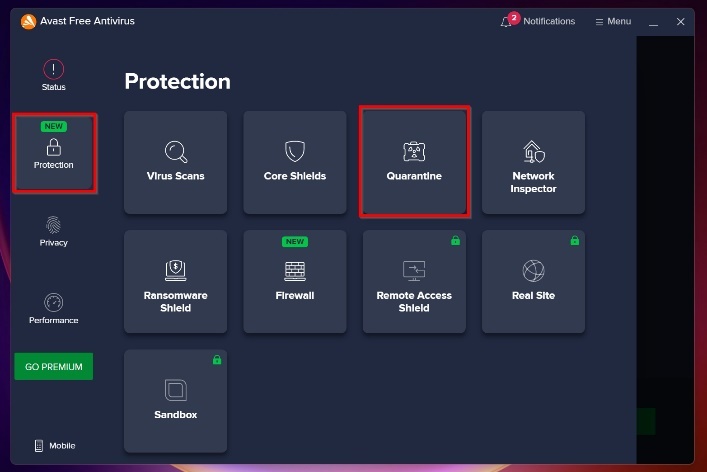
בשנת 2006 חבילת האבטחה של התוכנה גדלה וכוללת תוכנת אנטי ריגול. ובשנת 2009 שלבה גם תוכנה של מניעת גניבת זהות. וכיום מעל ל- 106 משתמשים בתוכנת האנטי וירוס AVG. התכונה משתמשת בשיטות זיהוי הבאות: זיהוי לפי חתימה, Heuristic Analysis, ניתוח התנהגות, זיהוי מבוסס ענן, למידת מכונה, ארגז חול, סריקה בזמן אמת.



Avast Antivirus

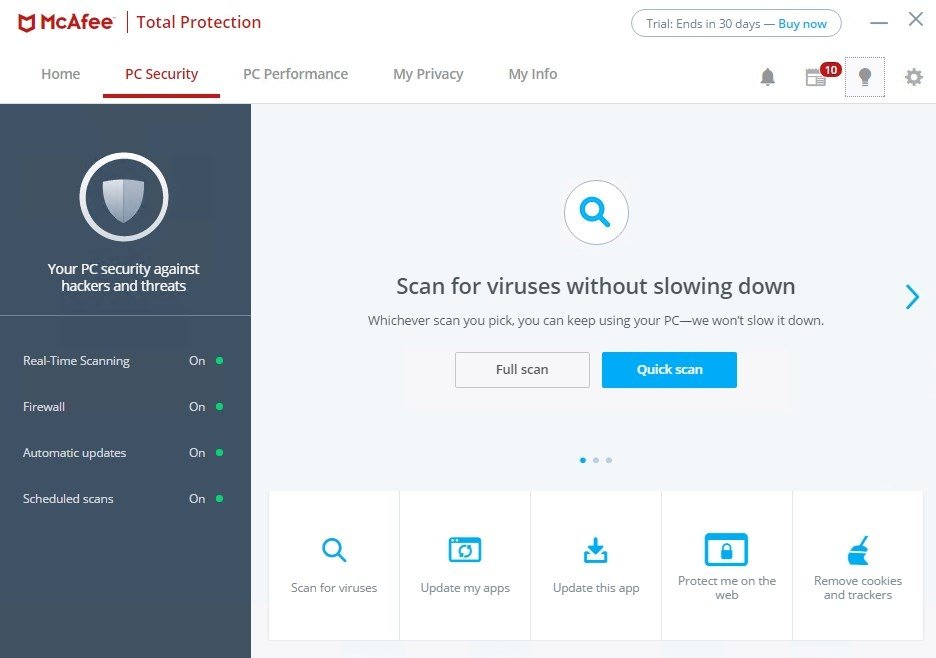
זאת תוכנת אנטי וירוס שפותחה על ידי החברה AVEST בממוקמת בצ'כיה בשנת 1988.

התוכנה מבוססת מנוע סריקה המתופעל על ידי מעבדות ICSA ומעבדות WEST COST, וכולל טכנולוגית להתגוננות מפני רוגלות ותוכנות Rootkit.

לתוכנה מהדורה חינמית וזמינה למערכות הפעלה ווינדוס ולינוקס. והיא זמינה גם בשפה העברית. כיום למעלה מ- 400 מיליון משתמשים בתכונה זו והיא זכתה במגוון פרסים בגלל אמינותה ורמת אבטחתה. משתמש בשיטות זיהוי הבאות: זיהוי לפי חתימה, Heuristic Analysis, ניתוח התנהגות, זיהוי מבוסס ענן, למידת מכונה, ארגז חול, סריקה בזמן אמת, הגנה על האינטרנט.

McAfee Antivirus / McAfee VirusScan

תוכנה זה יצאה לראשונה השנת 1988 וכיום מוצגת כמוצר אחד המורכב מכמה מוצרים של החברה הכוללים מגוון של שיטות הגנה מווירוסים. ומוצעת למגוון של מערכות הפעלה כמו: ווינדוס, לינוקס, אנדרואיד, MAC OS ועוד.

משתמש בשיטות זיהוי הבאות: זיהוי לפי חתימה, Heuristic Analysis, ניתוח התנהגות, זיהוי מבוסס ענן, למידת מכונה, ארגז חול, סריקה בזמן אמת, הגנה על האינטרנט, הגנה על אימייל, שיטות חציבת מידע.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| תמונה שמכילה גרפיקה, גופן, עיצוב גרפי, לוגו  התיאור נוצר באופן אוטומטי | תמונה שמכילה גרפיקה, אומנות קליפיפם, יצירתיות, סרט מצויר  התיאור נוצר באופן אוטומטי | תמונה שמכילה עיצוב גרפי, גרפיקה, צבעוני, אומנות  התיאור נוצר באופן אוטומטי | תמונה שמכילה לוגו, סמל, גרפיקה, גופן  התיאור נוצר באופן אוטומטי |  |
| תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | משתמשת במספר של שיטות זיהוי |
| תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם x עם מילוי מלא | תומך בכמה מערכות הפעלה |
| תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | משתמש בלמידה מכונה |
| 11+ | 10+ | 9+ | 2 | מספר שיטות הזיהוי בהם משתמש |
| תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | מבודד את הקבצים החשודים |
| תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | תיבת סימון עם v עם מילוי מלא | בעל ממשק משתמש ידידותי |

סיכום והשוואה של המוצרים הקיימים והפרויקט שלי:

# תוצר סופי

### תיאור הפרויקט

הפרויקט שלי נקרא AV (קיצור של אנטי וירוס). בפרויקט זה יצרתי אנטי וירוס הפועל באמצעות שני שיטות זיהוי: זיהוי לפי חתימה, וזיהוי לפי התנהגות סטטית עליהם הסברתי בחלק של התאוריה.

ביצוע הסריקות מתבצע בצורה ידנית (במטרה שיתבצעו גם בצורה אוטומטית) לאחר כל סריקה (בו תיקיות או קבצים בודדים נסרקים בשני השיטות) את הקבצים החשודים מבודד בתיקייה מיועדת לכך. ולכל קובץ חשוד ניתן או להמשיך לבודד אותו או למחוק אותו או לאפשר אותו (כלומר להחזיר אותו למקום בו היה).

#### רכיבי המערכת העיקריים

* **AV\_GUI.py / AV\_GUI\_UI.ui** – זוג הקבצים האלו מייצג את מערכת הפעלה של הפרויקט השני פורמטים. מערכת הפעלה מבוססת QT ומשמשת כמקשרת בין המשתמש למערכת מאחורי הקלעים.
* **GUI\_Setup.py** – קובץ הפיתון אחראי על לחבר 90% מהפעולות שמאחורי הקלעים ל-GUI בלעדיו אף כפתור לא היה עובד ולא היה אפשר להשתמש. תפקידו או להעביר את מיקומי הקבצים או התיקיות לפעולה שסורקת וגם להפעיל את ה- GUI. אין לקובץ MAIN אבל כשקוראים לו בקובץ המרכזי הוא מפעיל כל הדברים.
* **AV.py** – הקובץ הזה הוא ה- MAIN של הפרויקט הוא מחבר בין כל הקבצים המשניים למערכת המרכזית ודרכו מפעילים את הפרויקט.
* **PE\_ML.py** – קובץ זה הוא אחראי לבצע את הסריקה על פי התנהגות. בתוכה היא מצבעת את הסריקות לכל קובץ מסוג EXE באמצעות הנותנים שמקבלת ובאמצעות למידת מכונה מסוג סיווג יער אקראי.
* **Virus\_Signature\_Detection.dll** – ה-DLL הזה נכתב בשפת C++ והוא אחראי לבצע את הסריקה לפי חתימות. הוא מבצע את הסריקות בצורה מהירה יחסית בגלל שנכתב בשפה זו.
* **VirusHandle.dll** – ה- DLLהזה אחראי על טיפול בקבצים החשודים כולל בידודם, מחיקתם, חזרתם למקום ותיעוד שלהם במסד נתונים ומצב הנתון (מבודד, נמחק, מורשה).
* **תיקיית DATA** – תיקייה זו מאחסנת את כל הנתונים שהסריקות צריכות בשביל לעבוד וגם את כל הנתונים שהמערכת אוספת בלעדי התיקייה הזו המערכת לא תפעל. הסריקה לפי חתימות משתמש במסדי הנתונים: VS1,VS2 והסריקה לפי התנהגות משתמשת במערכי הנתונים מסוג CSV כדי לקבל מידע בשביל לאמן את מכונת הלמידה.

### אלגוריתמיים עיקריים

* **scan\_files (from AV.py)** – אלגוריתם זה הוא בעצם האלגוריתם הראשי שקורא לביצוע הסריקות במערכת. כאשר לוחצים על כפתור הסריקה ב- GUI המערכת לוקחת את הכתובת שנבחרה וסופרת כמה קבצים סך הכל עומדים להיסרק לאחר מכן מבצעים את שני סוגי הסריקות בתהליכונים במקביל שלבסוף היא מקבלת מכל סוג את מיקום הקבצים החשודים וסוגם ולאחר מכן מבודדת אותם ומודיע על כך למשתמש ב- GUI.
* **multi\_models\_predict\_exe (from PE\_ML.py)** – אלגוריתם זה הוא אלגוריתם הראשי של הזיהוי לפי התנהגות והוא אחראי לקבל מידע של הקבצים. ולהשתמש במכונה המאומנת בשביל לנחש איזה סוג קובץ הוא שישה סוגי וירוס או לא מסוכן. מה שקורה באלגוריתם הזה הוא שנוצרים 4 מכונות שכל אחד מהם לומד חלק אחר של התנהגות של קובץ: header, sections, what api functions it uses, dll it used. ואותו מידע נאסף גם מהקובץ שנבדק והמכונה יודעת לחזות אם בכל אחד מה רוב הסיכויים סוג הקובץ הזה (0 זה לא מסוכן, כל השאר סוגים של וירוסים) ובסוף לוקחים את ארבעת החיזויים ואם יש רוב למספר מסוים כלומר מופיע 3 פעמים לפחות אז החיזוי הסופי יהיה זה אחרת אם מופיע 0 ולא מופיע מספר אחר 3 אז יוחלט כאפס. אם לא מופיע 0 אז הולכים לפי הרוב לדוגמה: אם יש 2 מסוג 1 ואחד מסוג 6 ו4 אז הוא יוחלט כסוג אחד. לאחר בדיקות הוחלט שככה זה הכי מדויק מכוון שיש מקרים שבהם קובץ מתנהג כמו כמה סוגי וירוסים זה בגלל שהתנהגות בחלק מהדברים יכולה להיות דומה ביניהם. וגם להתנהגות של קבצים רגילים יכולה לפעמים להיות מוטעית כווירוס לכן צריך שהיה וודאות במקרה שמופיע 0 כי אז יכול להיות שקשה למודל להבדיל לגמרי.
* **PE\_Extraction.py** – כל הקובץ הזה הוא אלגוריתם אחד שאחרי להוציא את המידע מהקובץ החדש כלומר באיזה DLLs הוא משתמש ולאיזה פעולות הוא קורא, גודל של ה- HEADER שלו ומידע על המקטעים שלו כמו: .text, .data כל זה באמצעות הספרייה pefile. ולבסוף במידע הזה משתמשת הפעולה multi\_models\_predict\_exeכדי לבצע את החיזוי שלה לגבי הקובץ.
* **processFiles** **(from VirusSignature.cpp)** – פעולה זאת היא הפעולה הראשית של הזיהוי לפי חתימה וזוהי פעולה מסוג רקורסיבית. בה הפעולה מפרקת את התיקיות לקבצים וכל קובץ בודקת אם נמצא במסד הנתונים (כלומר וירוס) אחרי שהעביר את הקובץ ל- MD5. לבסוף מחזיר מערך שבכל תא יש שני מחרוזות אם מיקומי הקובץ וסוג הווירוס.
* **SpecifyVirus (from VirusSignature.cpp)** – זוהי הפעולה שפונה למסדי הנתונים ובעצם מבצעת את הזיהוי הסופי של החתימה של הקבצים החשודים לווירוסים. ולבסוף מחזירה את סוג הווירוס.
* **HashFileToMD5** **(from VirusSignature.cpp)** – אלגוריתם זה אחראי להפוך כל קובץ ל- MD5ועושה זאת באמצעות ספריית OPENSSL. בלי אלגוריתם זה אי אפשר לבצע את הסריקה כלל.
* **quarantinefile (from VirusHandle.cpp)** – זו הפונקציה האחראית לבודד את כל הקבצים החשודים ולשמור אותם במסד הנתונים. מה שהפונקציה עושה זה בעצם מעתיקה את הקובץ לתיקייה המבודדת ומוחקת את הקובץ מהמקום הקיים. בנוסף היא שומרת את המיקום הישן והחדש שלו במסד הנתונים ככה שכאשר צריך למחוק או להחזיר את הקובץ למקום נדע לאן.
* **class AV\_Application (from GUI\_Setup.py)** – בעצם זה האלגוריתם המרכזי של הממשק הפרויקט בשנייה שיוצרים משתנה ממנו מערכת הפעלה מתחילה. בתוכו מוגדרות הפעולות לכל הכפתורים מלבד כפתור הסריקה שפעולתו מוגדרת מחוץ למחלקה הזאת. בלי החלק הזה הממשק לא יעבוד והמשתמש לא יוכל להפעיל את הפרויקט (לסרוק ולטפל בקבצים.)
* **class Threat\_UI** **(from GUI\_Setup.py)** – עוד חלק חשוב מאוד לממשק המשתמש. כאשר קובץ נמצא כחשוד נוצר יישומון בשבילו שם כתוב שמו, סוג הווירוס, סטטוס שלו (מבודד, הוסר, מורשה) ושני כפתורים אחד מחיקה ואחד אישור ברגע שאחד הכפתורים נלחצו הם מבוטלים והיא אפשר לבצע את הפעולות האחרות כולל לבודד שוב פעם. כל עוד שום כפתור לא נלחץ הקובץ נשאר מבודד.

### דרישות ומגבלות מערכת

#### דרישות מערכת:

* בשביל להריץ את המערכת כראוי אין להזיז שוב קובץ ממקומו כי אחרת המערכת עלולה שלא לזהות חלק מן הקבצים ועלול להיגרם שגיאות ותקלות.
* ככול שהמעבד חזק יותר ככה הסריקות ייקחו פחות זמן. יש סיכוים שכאשר המעבד ישן או לא חזק מספיק שסריקות ארוכות של כל הכונן לדוגמה ייקח כמה ימים או יגרום לקריסה של המערכת. לכן, ממליץ לא לסרוק כוננים בכלל אלא רק תיקיות.

#### מגבלות מערכת

* תמיכה רק במערכת הפעלה מסוג ווינדוס: בגלל שסוג הזיהוי השני סורק קבצי exe שהם קבצים הרצה של ווינדוס אז המערכת עליה מפעילים את הפרויקט צריך להיות ווינדוס.
* גרסת פייתון 3.10 ומעלה. הפרויקט נכתב בגרסה זו של פייתון לכן יש סיכוי שגרסה קודמת לא תעבוד כמו שצריך. כאשר ניסיתי להריץ את זה בגרסת 3.9 הוא לא הצליח להריץ.
* המערכת כרגע לא תומכת בשפות שונות מאנגלית כלומר אם שם הקובץ בעברית או חלק מהכתובת שלו בעברית המערכת פשוט תדלג על הקובץ ולא תסרוק אותו או תטפל בו במידה וחשוד.
* ניתן לבצע רק סריקה אחת כל פעם וזה ככה כדי למנוע עומסים על המערכת גם אנטי וירוס בשוק מבצע סריקה בודדת כל פעם.
* קבצים חשודים שלא טופלו לא יופיעו שוב פעם אחרי אתחול המערכת (כרגע).

### אבטחה

אנטי וירוס מטרתו הוא להגן על המחשב מפני קבצים חשודים וזדוניים שהמשתמש עלול להוריד בטעות או שקיימים על המחשב והוא לא ידע שהם עלולים לפגוע במחשב. לכן הפרויקט עצמו נותן אבטחה למשתמש.

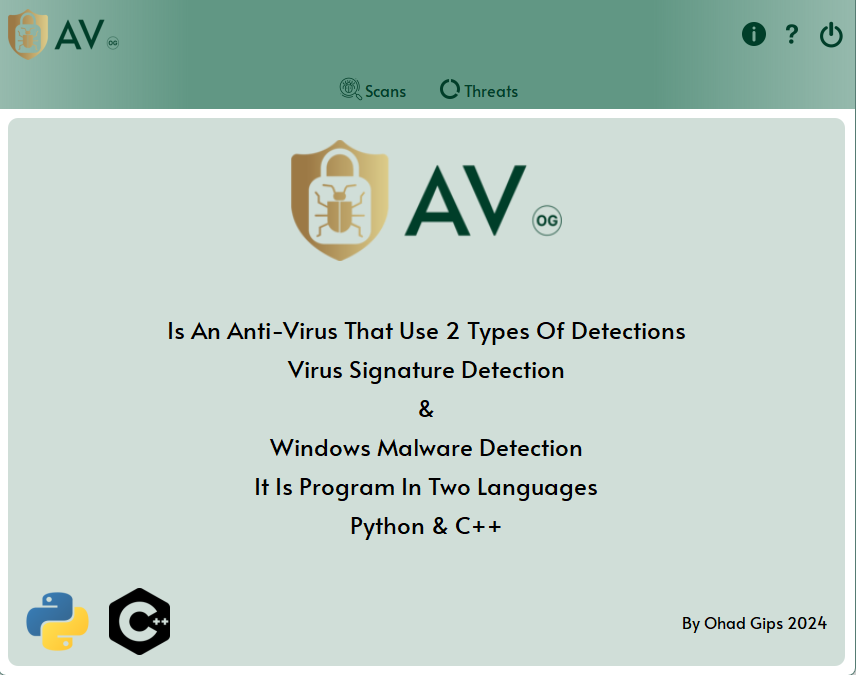
בנוסף הפרויקט לא מבצע תקשורת עם מכשירים אחרים כי אמור לשמור על המידע ולתת לווירוסים להתפשט לכן בכך שלא מבצע תקשורת שומר על המחשב מפני תוכנות זדוניות ממחשבים אחרים.

לסיכום, בתוך הפרויקט קיים מניעת שגיאות כדי שמערכת לא תקרוס. אבל מעבר לכך המערכת עצמה מטרתה לתת את לאבטח את המחשב. לכן, אין בה איזה אבטחה נוספת וגם היא לא צריכה.

### ממשק משתמש

ממשק המשתמש שלי נכתב ונבנה על ידי בעזרת התוכנה QT והתיקייה שלה בפייתון PyQt5

#### חלון ההתחלה / חלון המידע

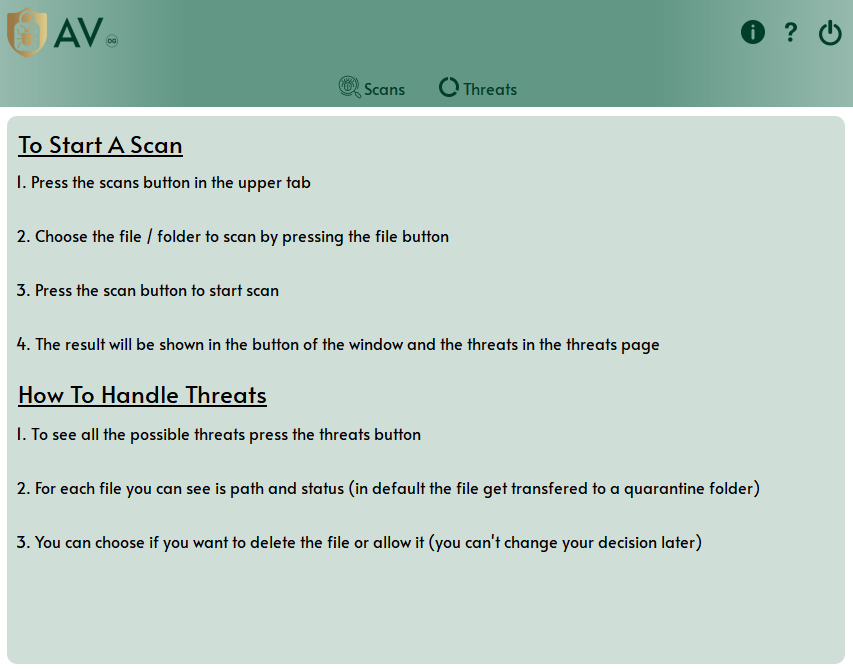


כאשר מריצים את הפרויקט זה החלון הראשון אותו רואים והוא נקרא דף המידע הוא מציג מידע קצר על הפרויקט כפי שניתן לראות בתמונה. כדי להגיע אליו שוב פעם צריך ללחוץ על כפתור ה- i למעלה בצד ימין.

במידה ורוצים לסגור את התוכנית בצורה נכונה יש ללחוץ על כפתור הכיבוי בפינה הימינית.

כדי לקבל עזרה על איך להתחיל סריקה ועוד יש ללחוץ על הכפתור בצורת סימן שאלה בפינה הימינית למעלה.

#### חלון העזרה



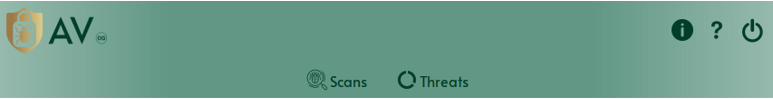
בחלון הזה כפי שניתן לראות יש הסבר מפורט באנגלית איך להתחיל סריקה בתוכנה ואיך לטפל בקבצים החשודים ולראות אותם אחרי הסריקה.

לחלון ביצוע הסריקות

הכיבוי

לחלון המידע

לחלון העזרה



לחלון טיפול בקבצים

לאחר שלחוצים על הכפתור שנקרא Scans מגיעים לחלון הבא:

#### חלון הסריקות

תמונה שמכילה טקסט, צילום מסך, גופן, תוכנה

התיאור נוצר באופן אוטומטי

כדי לבצע סריקה צריך קודם כל לבחור תיקיה או קובץ בשביל לסרוק.

כדי לבחור תיקייה נלחץ על הפתור שנקרא כמו הלוגו של הסייר קבצים בצורת תיקיה. ועל כפתור בצורת דף בשביל לבחור קובץ בודד. רק כאשר נבחר כותבת שקיימת (כלומר כתוב כתובת שקיימת בריבוע השמנת באמצע) יופיע כפתור הסריקה פשוט צריך להקליק על כפתור.

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

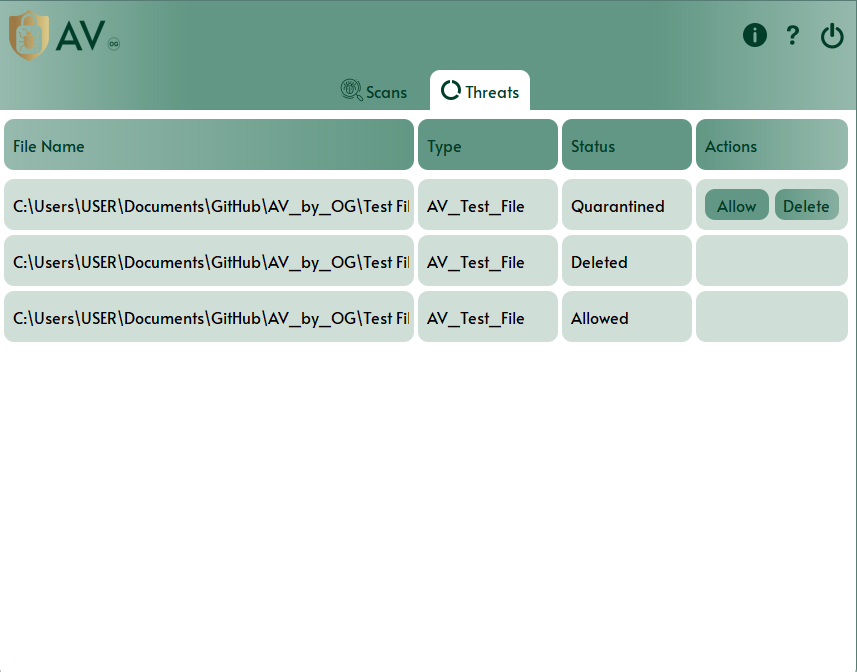
התיאור נוצר באופן אוטומטי

דוגמה לחלון לאחר סריקה ובמהלך סריקה אחרת. במהלך סריקה הכפתור של הסריקה נעלם ומוחלף עם אנימציה של טעינה. ובסוף כל סריקה בחלק למטה שנקרא Scan Result

יתעדכן בסוף כל סריקה: מתי התרחשה הסריקה וכמה זמן לקחה (בשניות), כמה קבצים נסרקו וכמה קבצים חשודים נמצאו.

כדי לראות את כל הקבצים שנמצאו חשודים ולטפל בהם צריך ללחוץ על כפתור ה- Threats פירוט על קבצים החושדים לא מופיע בעמוד הסריקה אלא בעמוד נפרד.

#### חלון הקבצים החשודים והטיפול בהם

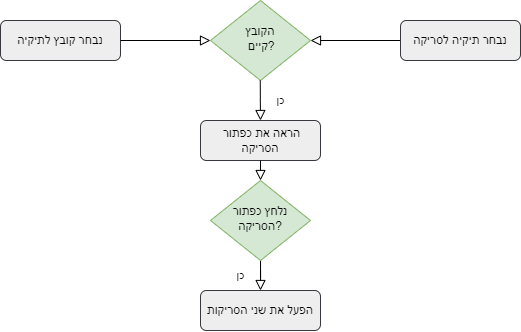


כאשר המערכת מוצאת קבצים חשודים היא רושמת אותם כמו שניתן לראות את בקובץ הראשון (הקובץ הראשון הוא תמיד הקובץ שהתגלה בסריקה האחרונה כלומר כל קובץ חשוד חדש נרשם עליון ברשימה).

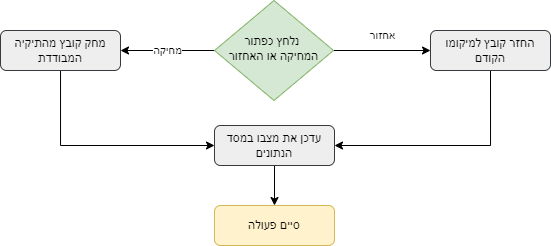
בצורה אוטומטית הקבצים מבודדים לתיקיה נפרדת המיועדת לכך. וניתנת האפשרות או למחוק אותם או להחזיר אותם חזרה למקומם (במקרה של טעות או אם סומכים על הקובץ) במידה ונבחרה אחת מהפעולות לא היה ניתן לבצע שום פעולה אחרת על הקובץ וגם לא להחזיר אותו לבידוד (אם חזר למקומו ומבצעים סריקה אז אפשר לבודד אותו שוב).

### תרחישים עיקריים

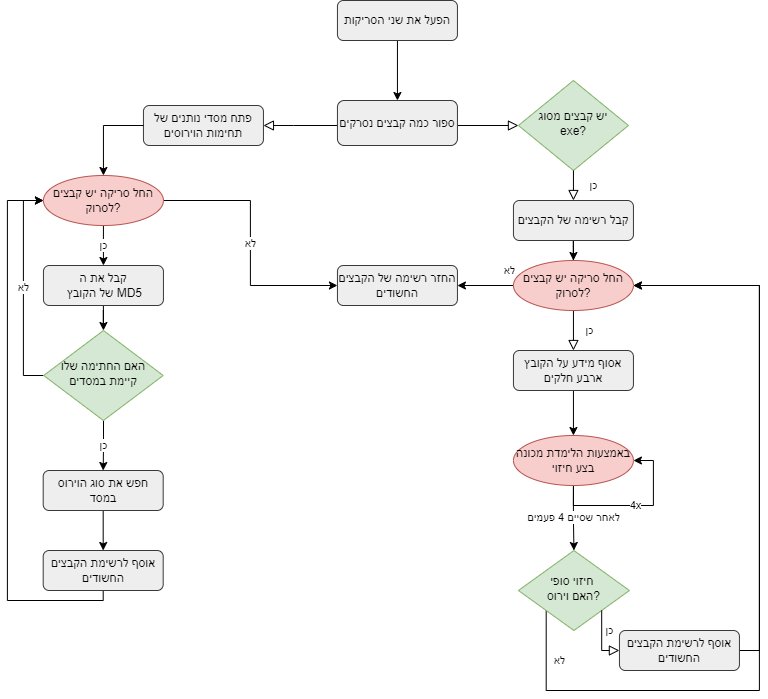
#### ביצוע של הסריקה בממשק המשתמש



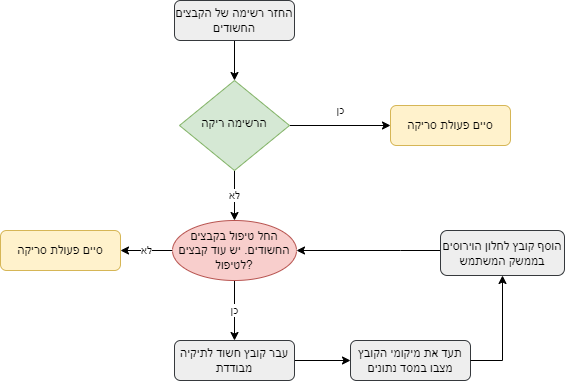
#### מחיקה או אחזור של קבצים חשודים



#### הפעלת הסריקות וקבלת קבצים חשודים



#### טיפול ברשימה של קבצים חשודים שהתקבלו מהסריקה



# נספחים

* קישור לקוד - <https://github.com/Ohadgips/AV-OG>