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 עניינים

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 HASH שנקרא MD5. 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:

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.

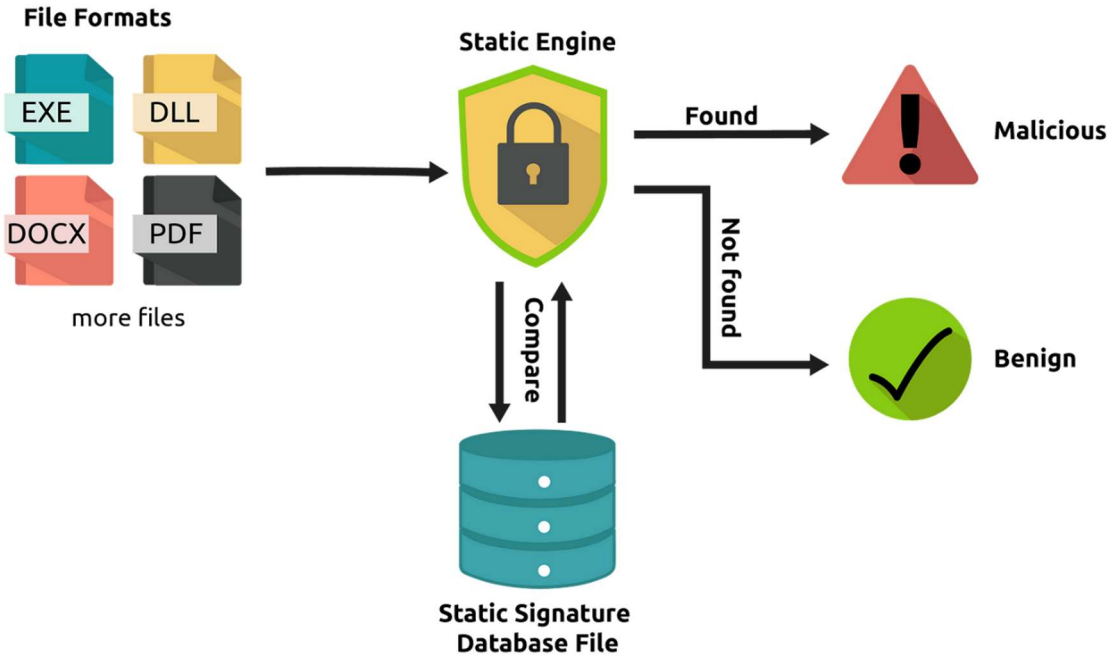
איך עובד אנטי וירוס?

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.



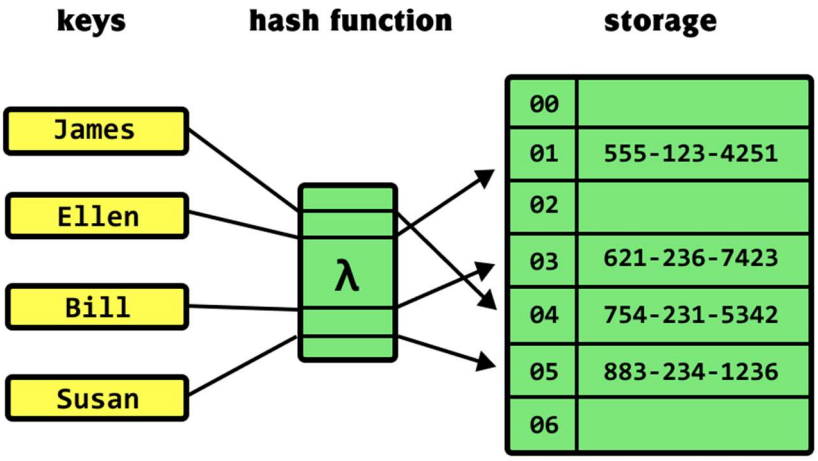
יתרונות של זיהוי זה – קודם כל החתימות האלו יכולות להיות בכל מיני סוגים שונים של קבצים ולכן אפשר להשתמש בשיטה זו על כל סוגי הקבצים בלי דרישה למערכת הפעלה מסוימת או תנאים מסוימים. בנוסף באמצעות מסד נותנים מתאים ניתן לבצע סריקות מהירות של מספר קבצים רב.

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.

<u>טבלת גיבוב (HASH)</u>

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.



באנטי וירוס שלי אני משתמש בחתימות מסוג MD5:

<u>MD5</u>

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)

איך 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:

קלט	פלט
MD5(“”)	d41d8cd9 8f00b204 e9800998 ecf8427e
MD5(“a”)	0cc175b9 c0f1b6a8 31c399e2 69772661
MD5(“abc”)	90015098 3cd24fb0 d6963f7d 28e17f72
MD5(“abcdefghijklmnopqrstuvwxyz”)	c3fcd3d7 6192e400 7dfb496c ca67e13b

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

תוצר סופי

תיאור הפרויקט

הפרויקט שלי נקרא AV (קיצור של אנטי וירוס). בפרויקט זה יצרתי אני 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-DLL הזה 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_exe כדי 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-MD5 ועושה 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.

נספחים

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