**Antivirus Evasion via Code Injection**

**מבוא:**

לפני שנציג את סוגי הזרקות הקוד השונות אשר באמצעותן ניתן לחמוק ממנועי האנטי-וירוס, יש להגדיר מספר מונחי בסיסי אשר ישמשו אותנו במהלך הצגת הטכניקות השונות להתחמקות מאנטי-וירוסים:

* **אנטי-וירוס**: תוכנה המשמשת למניעה, גילוי והסרה של תוכנות זדוניות, כולל וירוסי מחשב, תולעים וסוס טרויאני. תוכנה זו עשויה גם למנוע ולהסיר תוכנות פרסום, תוכנות ריגול וצורות אחרות של תוכנות זדוניות.
* **AV מבוסס חתימות**: סוג זה של מנועי אנטי-וירוס מיוצרים על בסיס כמה חתימות (דפוסים שנוצרו כדי לזהות רצף מסוים של בתים) שנוצרו כדי לזהות איזה סוג קוד יכול היה להכיל ההפעלה הבאה. אם הקוד או התבנית של בתים נמצאים בתוכן שברשימה השחורה, אז התראת AV מופעלת.
* **AV מבוסס היוריסטיקה**: טכניקה מודרנית לאיתור מטען זדוני על ידי הגדרת כמה כללים ואלגוריתמים כך שאם הקוד מכיל קבוצה מסוימת של קוד / הוראות או קטעי הפעלה, שעושים פעולות מסוימות שעלולות להוכיח זדוניות, אז התראת AV מופעלת.
* **AV מבוסס "התנהגות" וסריקות דינאמיות**: טכניקה זו מנתחת את התנהגות הקובץ הבינארי ומסווגת אותו לזדוני ולא זדוני. מרבית האנטי-וירוסים כיום מבוססים על גישה דינאמית, כלומר, קובץ הקוד הפוטנציאלי כזדוני יורץ על גבי סביבה וירטואלית למשך זמן קצר וזאת תוך כדי השוואת התנהגותו אל מול רשימת החתימות + היוריסטיקה. במהלך הניתוח, יתאפשר גילוי מהותו של הקוד המורץ גם אם מדובר בקובץ מוצפן וזאת ללא פענוח וע"י חקר התנהגותו וחתימותיו בלבד.

האנטי-וירוס ממלא תפקיד מרכזי באבטחת המערכת. עבור האקרים / Pentesters, יש כמה בעיות גדולות בשלב שלאחר החדירה למערכת, בדרך כלל ירצה התוקף להעלות כמה כלים על מכונת הקורבן לשם שליטה טובה יותר, אך כאן האנטי-וירוס "משחק" עם הכלים שלנו ומזהה אותם כקובץ זדוני ומוחק אותם. זה עשוי להיות אפשרי לחמוק מזיהוי אם מתבצע שימוש בכלי שנוצר ספציפית לתקיפה זו ונכתב בלעדית, אך זהו מצב נדיר. במקרה בו מבוצע שימוש בכלים החשופים לציבור, יש סבירות גבוהה יותר להיתפס על ידי אנטי-וירוס.

פתרון לבעיה המתוארת לעיל ניתן למצוא באחת משתי דרכים עיקריות:

* הסתרת הקוד שעשוי להיות מזוהה כזדוני. זה נעשה בדרך כלל באמצעות הצפנה.
* קידוד בצורה כזו אשר תמנע מה-AV במערכת לזהות את הקוד המורץ כווירוס \ קובץ זדוני.

**תוכן:**

הזרקת קוד מורכבת מהפעלת קוד בזיכרון של תהליך אחר. באופן כללי באמצעות הזרקת DLL אך קיימות אפשרויות אחרות ואף ניתן להזריק EXE שלם. המורכבות של התהליך טמונה בעובדה שהקוד המוזרק חייב למצוא דרך לבצע את עצמו מבלי להיטען על ידי המערכת (במיוחד מכיוון שכתובת הבסיס אינה זהה). לצורך הזרקת DLL זה נעשה כאשר ה- DLL נטען. לצורך הזרקת קוד, הקוד צריך להיות מסוגל לשנות את מצביעי הזיכרון שלה בהתבסס על Reallocation.

**שימוש ב-Binders**:

Binders משמשים לקישור שניים או יותר קבצי EXE לקובץ EXE אחד.

כך ניתן להסתיר וירוסים או תולעים באמצעות Binders והם יכולים לפעול ללא זיהוי של AV . החתימה המקורית של הקבצים הזדוניים מועברת לקיזוז בבינארי החדש שנוצר והוא יכול להתחמק בקלות מכל מוצרים אנטי-וירוסיים סטטיים.

Binder בדרך כלל קושר לעצמו קבצי EXE אחרים ויוצר בינארי חדש. למשל הגודל המקורי של קובץ ה- Binder הוא 20KB וגודל ה- EXE שצורף הוא 35KB אז הגודל הסופי של ה- EXE החדש שנוצר יהיה 55KB.

**שימוש ב-**:**Packers**

עובד דומה מאוד לאופן שבו Binders פועלים אך ההבדל היחיד ביניהם במקרה של Packers, הבינארי הזדוני נדחס לפני שהוא מקבל מוטבע לבינארי של החבילה ליצירת ה- EXE הסופי. זה הופך כל מוצר אנטי-וירוס לחסר אונים באיתור הבינאריים הזדוניים מכיוון שבוצעו שינויים בחתימה שלו בגלל הדחיסה.

כמו כן, ישנן טכניקות בהן ניתן להשתמש כדי להמיר קובץ הפעלה או כל קובץ אחר (כמו .pif או .scr) לקובץ vbs ובביצוע קובץ vbs הקובץ הבינארי הנסתר יבוצע באופן אוטומטי.

**שימוש ב-Cryptors**:

מנגנון זה משנה באופן קריפטוגרפי את קוד התוכנית / ההפעלה ועושה פונקציית פענוח או תהליך משנה כמו stub. כאשר התוכנית נשלחת היא מוצפנת ותקע הפענוח מוסתר. לגרום לזה להיראות חסר תועלת לחלוטין עבור AV, וכך לעקוף אותו. פענוח קוד מוצפן עובד בזיכרון, וקוד הפעלה מפוענח נותר בדיסק. לפיכך AV לא יכול היה להסיר אותו לפני ביצועו.

**סיכום:**

הזרקת קוד היא ניצול של באג מחשב שנגרם מעיבוד נתונים לא חוקיים. הזרקה משמשת תוקף כדי להכניס (או "להזריק") קוד לתוכנת מחשב פגיעה ולשנות את דרך הביצוע. התוצאה של הזרקת קוד מוצלחת יכולה להיות הרת אסון, למשל על ידי מתן אפשרות לתולעי מחשב להתפשט. פגיעות בהזרקת קוד מתרחשות כאשר יישום שולח נתונים לא מהימנים למתורגמן. "פגמים" בהזרקה יכולים להימצא לרוב בשאילתות SQL, LDAP, XPath או NoSQL; פקודות מערכת הפעלה; מנתחי XML, כותרות SMTP, ארגומנטים בתוכנית וכו'. קל יותר לגלות פגמים בהזרקה בבחינת קוד המקור מאשר באמצעות בסריקה. הזרקה עלולה לגרום לאובדן נתונים או לשחיתות, חוסר אחריות או מניעת גישה. הזרקה יכולה לפעמים להוביל להשתלטות מוחלטת על המכונה.