**Malware documentation**

**Omri Zaiman**

חווית המשתמש ביחס לפוגען

הפוגען מוצג כתכנית anti-virus שהמשתמש מוריד ומתקין (כדי למנוע חשד) אך בפועל עושה בדיוק הפוך. המשתמש פוגש רק בקובץ ראשי אחד. ברגע שהוא מפעיל את הקובץ הראשי, הפעולות קורות "מאחורי הקלעים" בזמן שהמשתמש סבור שהוא מוגן (הפוגען רץ ברקע).

פעילות הפוגען "מאחורי הקלעים"

הפוגען מבוסס על קובץ EXE אחד שפועל בכמה דרכים שונות בהתאם למיקום שלו על המחשב. למעשה, יש קובץ ראשי (שיוצג בעת פעילות כהתקנה של אנטי וירוס) שיוצר את הפוגען בתוך תיקיה נסתרת עם השם "SystemSettings" שאמור להיות לא מחשיד, ועם שם שמידמה לקובץ מערכת- "svchost.exe" (הפוגען מנצל את העובדה שזהו שם של קובץ מערכת ובכך מקשה על מי שמתמודד איתו לזהות אותו במעבר על הprocesses, כיוון שהוא לא יהיה התהליך היחיד עם שם זה). הקובץ הראשי גם מקשר את הפוגען ל-run ב-registry וגורם לו לרוץ מחדש עם כל התחברות של המשתמש למחשב. המערכת בנויה על קובץ EXE אחד שמשכפל את עצמו ומשנה את ההתנהגות שלו.

* הערה- לחיצה על כפתור ההרצה בvisusal studio שווה להפעלת קובץ הexe שיוצר את הוירוס.
* למעשה קובץ הEXE האחד יתנהג כמו וירוס רק כאשר הוא ממוקם בתיקייה הרצויה, בכל מצב אחר הוא ישמש כהתקנה לוירוס.

תיאור כללי של איסוף המידע

בחרתי לבסס את הפוגען על מחלקה שיצרתי בשם Round. המחלקה מייצגת "סיבוב" אחד של איסוף המידע. במהלך פעולת האובייקט נאספים הנתונים הנדרשים. כעבור 60 שניות נקרא המפרק של האובייקט והוא זה שאחראי על שמירת המידע בקובץ הרלוונטי. ישנה לולאה כללית שבכל איטרציה יוצרת אובייקט Round, מחכה דקה אחת, ובסוף הscope מפרקת את האובייקט.

תיאור הפונקציות שאוספות את המידע

1. פונקציה לזיהוי keystrokes:

הפונקציה היא ת'רד שמורץ ביצירת האובייקט round ומסתיים בפירוק האובייקט. לאורך כל הזמן שהאובייקט קיים, הפונקציה בודקת בעזרת כלים של WinAPI מהם המקשים עליהם לחץ המשתמש. הפונקציה אוספת את כל הkeystrokes שהמשתמש מבצע בזמן שהתכנית מורצת ברקע ללא ידיעתו.

2. פונקציה לצילום screenshot:

הפונקציה משתמשת בכלים של WinAPI כדי ליצור צילום מסך של המשתמש. כלומר, מה שהוא רואה על המסך ברגע הצילום. הפונקציה מקשרת בין אובייקט הCimage בתכונות המחלקה Round, לבין הbitMap שנוצר בפונקציה.

3. זיהוי כתובות ה-IP אליהן פונה המחשב:

בחרתי להיעזר בsystem עם הפקודה "netstat -n". הפקודה מספקת מידע לגבי הקשרים הפעילים במחשב. את תוצאות הפקודה הכנסתי לקובץ זמני. לאחר מכן, עברתי על תוכן הקובץ לפי שורות וזיהיתי בכל שורה את כתובות הIP לפי חיתוך מחרוזות. לאחר ששמרתי את כל כתובות הIP הרלוונטיות בתוך וקטור של מחרוזות (כתכונה של המחלקה Round), מחקתי את הקובץ הזמני.

\* בחרתי לעבור על כל שורה בקובץ בנפרד ולא לשמור את כל תוכן הקובץ בתוך מחרוזת כיוון שגודל הקובץ תלוי במספר הקשרים הפעילים במחשב עליו מורץ הפוגען, במחשב המתקשר עם כמות גדולה של ישויות אחרות גודל הקובץ יכול להיות מסיבי ולכן לא נרצה לשמור אותו בתוך מחרוזת בודדת.

4. זיהוי שם המשתמש המחובר למחשב:

בחרתי להשתמש בפונקציה מספריית windows.h שמספקת את שם המשתמש המחובר למחשב. את השם שמרתי כמחרוזת שהיא תכונה במחלקה Round.

5. זיהוי הקבוצות אליהן המשתמש שייך:

בחרתי להיעזר בsystem עם הפקודה " WHOAMI /GROUPS /FO CSV /NH". הפקודה מספקת מידע לגבי הקבוצות אליהן המשתמש משתייך. את תוצאות הפקודה הכנסתי לקובץ זמני. לאחר מכן, עברתי על תוכן הקובץ לפי שורות וזיהיתי בכל שורה את שם הקבוצה לפי חיתוך מחרוזות. לאחר ששמרתי את שמות הקבוצות בתוך וקטור של מחרוזות (כתכונה של המחלקה Round), מחקתי את הקובץ הזמני.

\* בחרתי לעבור על כל שורה בקובץ בנפרד ולא לשמור את כל תוכן הקובץ בתוך מחרוזת כיוון שגודל הקובץ תלוי במספר הקבוצות אליהן שייך המשתמש במחשב עליו מורץ הפוגען, במחשב בו המשתמש שייך לכמות גדולה של קבוצות גודל הקובץ יכול להיות מסיבי ולכן לא נרצה לשמור אותו בתוך מחרוזת בודדת.

6. זיהוי גרסת מערכת ההפעלה:

בחרתי להשתמש בפונקציה RtlGetVersion שמספקת בין היתר מידע על גרסת מערכת ההפעלה.

7. זיהוי שמות התהליכים שמורצים על המחשב:

בחרתי להיעזר בsystem עם הפקודה "tasklist". הפקודה מספקת מידע לגבי הprocessים המורצים על המחשב. את תוצאות הפקודה הכנסתי לקובץ זמני. לאחר מכן, עברתי על תוכן הקובץ לפי שורות וזיהיתי בכל שורה את שם התהליך לפי חיתוך מחרוזות. לאחר ששמרתי את שמות התהליכים בתוך וקטור של מחרוזות (כתכונה של המחלקה Round), מחקתי את הקובץ הזמני.

\* בחרתי לעבור על כל שורה בקובץ בנפרד ולא לשמור את כל תוכן הקובץ בתוך מחרוזת כיוון שגודל הקובץ תלוי במספר התהליכים המורצים במחשב עליו מורץ הפוגען, במחשב בו מורצת כמות גדולה של תהליכים, גודל הקובץ יכול להיות מסיבי ולכן לא נרצה לשמור את כולו בתוך מחרוזת בודדת.

8. זיהוי כתובת ה-IP של המחשב:

בחרתי להיעזר בsystem עם הפקודה "ipconfig | findstr / C:\"IPv4 Address". הפקודה מספקת את הip הפנימי של המחשב ברשת, ה-subnet mask וה-default gateway. התוספת של findstr מסננת את התוצאות של הפקודה ומשאירה את מה שאמור להיות רלוונטי. את תוצאות הפקודה הכנסתי לקובץ זמני ומיד הכנתי את תוכן הקובץ למחרוזת ומחקתי את הקובץ. בהמשך זיהיתי את הIP לפי חיתוך מחרוזות ושמרתי אותו כתכונה של המחלקה Round.

\* בחרתי להכניס את כל תוכן הקובץ למחרוזת ולא לעבור על הקובץ לפי שורות כיוון שהתוספת findstr לפקודה מסננת רק את התוכן שחשוד להיות רלוונטי ולכן אין חשש שהקובץ יהיה גדול מדי. ניתן להכניס את כל התוכן למחרוזת.

הערות

\*תשובה לסעיף ד- אפשרות נוספת להרצה אוטומטית של הקובץ עם עליית המחשב היא השמה של הקובץ בתיקיית startup שמריצה תכניות עם עליית המחשב. בחרתי להשתמש בregistry (כמתואר בפסקה "פעילות הפוגען מאחורי הקלעים") כיוון שישנה סבירות גבוהה שמשתמש עם פחות רקע במחשבים יהיה יותר חשוף לstartup מאשר לערכים ב-registry. בנוסף, חומת האש של windows יכולה להתריע על דברים חשודים שהיא מזהה בתיקיית startup.

\*תשובה לסעיף ו'- הדרך הנוספת להקשות על זיהוי הפוגען היא לתת לקובץ שם שלא יעורר חשד, כפי שמתואר בפסקה "פעילות הפוגען "מאחורי הקלעים"".

נספחים

* התיקייה הנסתרת שמידמה לתיקיית מערכת:
* קובץ הוירוס שמידמה לקובץ של מערכת ההפעלה:



* ההופעה של הפוגען בRegistry:



* הקבצים שהוירוס יוצר:

