**הסבר על נוזקת windows offender**

בהסבר אציג את 3 הפונקציות החשובות ביותר של הנוזקה וכיצד פיתחתי והגעתי אליהן:

1.uac-bypass

הפונקציה מבצעת האצלת סמכויות במערכות ווינדוס 10-11. העיקרון פשוט, ישנו סרוויס האחראי על הגדרות כלליות של מערכת ההפעלה ושמו fodhelper.exe. לאחר מחקר מעמיק עליו, גיליתי כי הוא קורא מתוך הרגיסטרי מפתח מסויים תחת hkcu. כלומר, ניתן לגרום לפרוסס תחת הרשאות system לקרוא מפתח שיש לנו שליטה על תוכנו. נרצה לגרום לסרוויס לבצע פקודה מסויימת. על מנת להגדיר את תוכן המפתח כפקודה נשתמש בdelegate execute.

הפונקציה עובדת כך:

1. קבלת הpath של עצמה
2. כתיבה שלו לתוך מפתח ברגיסטרי(הנתיב בקוד)
3. כתיבת מפתח נוסף תחת אותו מקום מסוג delegate execute
4. קריאה לfodhelper
5. ריצה בהרשאות גבוהות

2. lsass-dumper

כלי האיסוף המרכזי של הנוזקה, המטרה שלו הייתה להוות תחליף שהוא הרבה יותר stealth מ mimikatzכך שתוכנות האנטיוירוס יפספסו אותו בסבירות גבוהה.

המטרה של מימיקאטצ היא למעשה לשלוף המו המון creds מתוך המערכת של ווינדוס, ועל ידי כך ניתן לפרוץ את ההאשים שלהן מקומית באמצעות jhon. לפני שניכנס לכך נבין איך מימיקאטצ עובד.

העיקרון ״פשוט״, לקחת את הסרוויס שאחראי על ניהול הcredidentials של windows ולשלוף את על הזיכרון שלו מתוך הram אל תוך קובץ טקסט. לשליפת הזיכרון לתוך הram יש טיפול פשוט, שימוש בwinapi minidumpcreatedumpfile. הפונקציה מקבלת pid וקובץ ופשוט כותבת את כל הזיכרון מתוך הפריים של הפרוסס שנתנו אל קובץ. אבל כפי שאתם מתארים, בשביל בכלל לגשת לכזה פרוסס כמו lsass נצטרך לעבוד מאוד קשה. שימוש בפונקציה הנל על lsass יגרום לווינדוס דיפנדר לסגור את הנוזקה שלנו!. לאחר מחקר מעמיק על הפרוסס, נצטרך ךהשיג הרשאות מאוד גבוהות ולהמשיך את ההאצלה מlocal admin. נצטרך שלפרוסס של הנוזקה תהיה הרשאה מיוחדת בשם SeDebugPrivilege. ההרשאה הזו מאפשרת לנו בתור מנהל מקומי, לגשת לזיכרון של פרוסס אחר, השילב של מנהל מקומי וההרשאה הזו ייתנו לנו לשהתמש בפונקציית הדאמפינג ולשלוף מהזיכרון את כל הסיסמאות!.

איך נוסיף את ההרשאה? נערוך את הsecurity token של הפרוסס שלנו בזמi הריצה. נשתמש בפונקציות הapi הבאות:

OpenProcessToken()

LookupPrivilegeValue()

AdjustPrivilegeToken()

נקבל הנדל לפרוסס, נבקש לקבל את הtoken שלו ונאחסן אותו בתוך struct מסוג luid. ניצור תוקן חדש ונכניס אליו את ההרשאה החדשה:

PrivilegeData.Privileges[0].Luid = Luid;

PrivilegeData.PrivilegeCount = 1;

PrivilegeData.Privileges[0].Attributes= SE\_PRIVILEGE\_ENABLED;

עכשיו נשתמש בפונקציה האחרונה כדי לעדכן את ההרשאות.

כעת יש לנו גישה מלאה לכמעט כל הזיכרון. רק נשארה סוגייה שנשמעת פשוטה אך היא לא ככ. לקבל pid של תהליך מתוך שם ולא הפוך. יש מספר גישות נפוצות לדבר, אך אני בחרתי להשתמש בפונקציה שכתבתי get-cmd-output תחת הפקודה :

tasklist |findstr lsass.exe

לאחר מכן השתמשתי בstrtok על מנת לפרסר את הoutput שלה ולהדיג את מספק הpid.

זה הכל,

עכשיו ניצור file חדש, וניתן לפונקציית הapi minidumpwritefile את הפידואת הקובץ. היא תכתוב את הכל לתוך הקובץ ובום, כל הקרדידנטיאלס במערכת נשלפו מתוך הram, מתוך הframe של lsass.exe והכל נמצא בקובץ שמוכן לשליחה לשרת הc&c שלנו, שמה נפרסר את האאוטפוט באמצעות מימיקאטצ ונפרוץ את ההאשים באמצעות dictionary based bruteforce ע״י jhonthereaper. מומלץ להשתמש בrockyou.txt.

**4.pe injection and hollowing in runtimebroker**

כפי שהסברתי, על הנוזקה להיות מאוד מאוד stealth. ועל כן, מה יותר טוה מאשר להחדיר את כל image שלה לתוך תהלעך שבכלל שייך לשמירת ההרשאות בווינדוס!. המטרה הייתה להזריק את כל תוכן הנוזקה לruntimebroker ככה שכל הפעילות הזדונית תצא מתהליך ששייך למערכת ההפעלה. אציין בשלבים איך עשיתי זאת ואוסיף רשימה של פונקציות הapi הרלוונטיות. הרעיון הכללי: לקחת את כל הimage שלנו, להקצות זיכרון בפרוסס המרוחק, לבצע את תפקיד הloader ולהתאים כתובות לפי הזיכרון המרוחק, להזריק את כל למרחב הכתובות של הפרוסס המרוחק, וליצור תראד על הפרוסס המרוחק שיהווה נקודת ההתחלה החדשה של התוכנה שלנו.

לפי שלבים:

1.נמצא את הpid של runtimrebroker לפי cmd.

2. נקבל אליו handle באמצעותOpenProcess.

3.נקצה בו זיכרון במרחב הזיכרון הורטואלי שלו באמצעות VirtualAllocEx ונקבל את התכתובת שלו.

4. נקצה זיכרון במרחב הזיכרון הוירטואלי של הפרוסס הנוכחי שלנו.

5. נקבל handle לimage של הפרוסס שלנו ונכתוב את כולו לתוך הזיכרון הוירטואלי שהקצנו במרחב שלנו.

6.נחשב את הdelta בין הכתובת שאליה הקצנו לבין הכתובת הנוכחית.

7. נגיע לטבלת הרילוקרציה באמצעות:

dos-header=>nt-header=>optional-header=>data-directory[4]

8. הטבלה מכילה מספר בלוקים כאשר כל בלוק מכיל מספר כניסות. אנו נירתה לעבור על כל כניסה ולהוסוף אליה את הדלתא שחשיבנו.

9.נכתוב לזיכרון המרוחק את הקוד ש״תיקנו״ ועשינו לו ״רירילוקשיישן״(:

10. נשתמש בCreateRemoteThread() על מנת ליצור טראד שירוץ על גבי התהליך המרוחק.

**הסברים נוספים בדרך... ☺**