באג או פיצ'ר? ניצול לרעה של יכולות מובנות במערכת ההפעלה Windows

מאת אביאל צרפתי

הקדמה

זה לא באג, זה פיצ'ר! אנו שומעים את הביטוי הזה לא מעט, בעיקר כהלצה, אך אם ננסה להבין מה עומד מאחוריו, נוכל להסיק כי בעולם הפיתוח ישנם אילוצים שונים ומשונים הגורמים למוצרים לצאת לשוק כאשר הם מכילים בעיות שונות - באגים מעצבנים ולעתים אף פרצות אבטחה שונות המוסוות תחת פיצ'רים, אם בכוונה ואם בטעות. במאמר זה אעסוק בניצול לרעה של יכולות שונות במערכות ההפעלה מבית Microsoft (Windows 7/10) בכדי לבצע שלל פעולות התקפיות על מערכת ההפעלה ולבסוף ביטול של יכולות מנגנון דלף מידע מבית McAfee (Data Loss Prevention או DLP בקצרה, ניתן לקרוא בהרחבה כאן - https://whatis.techtarget.com/definition/data-loss-prevention-DLP).

במאמר זה אציג דרכי פעולה וקווים מנחים לתקיפת תחנת קצה, הכוללים גם ביטול של מערכת DLP מבית McAfee, המותקנת על עמדת קצה מסוג Windows 10. העבודה תיעשה בעזרת כלים מובנים של מערכת ההפעלה בלבד, וללא שימוש בכלים "חיצוניים" כלל. לכל אורך התהליך אתייחס גם לתחום ה-Defense Evasion והשארת מספר נמוך ככל האפשר של עקבות בתחנת הקצה.

פתרון ה-DLP מותקן על עמדת הקצה יחד עם Agent ייעודי של חברת McAfee הכולל מספר פתרונות אבטחה נוספים, שמהם נרצה להתחמק בכדי לייצר "רעש" מינימלי ככל האפשר על עמדת הקצה. בכדי להתחמק ממנגנוני האבטחה השונים המותקנים על עמדת הקצה, בשלב הראשוני נבחן מהם מנגנוני האבטחה הקיימים בכדי לדעת ממה "להיזהר":

- Anti-Malware סטנדרטי מבוסס חתימות.
- מערכת לזיהוי קבצים זדוניים המבוססת על "מוניטין" הקובץ. ניתן לקרוא בהרחבה במאמר הבא: https://www.techopedia.com/definition/4080/reputation-based-security
- מערכת לזיהוי ומניעת פעולות עוינות - Endpoint Detection & Response, או בקצרה - EDR.

בתרחיש שאדגים במאמר זה, אתחיל את הבדיקה כאשר בידינו משתמש סטנדרטי ללא הרשאות "גבוהות" על תחנת הקצה, ואנסה להגיע למצב של נטרול מערכת ה-DLP המותקנת על התחנה תוך שמירה על פרופיל נמוך ככל האפשר בכדי להימנע מגילוי על ידי אחת ממערכת ההגנה.

שלב 1 - Privilege Escalation | משתמש רגיל ← משתמש חזק

כפי שהבטחתי בתחילת המאמר - אעשה שימוש בכלים מובנים של מערכת ההפעלה בלבד על תחנת הקצה, בכדי להימנע מגילוי של מערכות הגנה מותקנות, או לפחות להקטין את הסיכוי לכך.

בשלב הראשוני, נמצא את עצמנו כמשתמש ללא הרשאות גבוהות על תחנת הקצה - זאת אומרת, ללא Local Administrator או כל הרשאה אחרת. נבחן מספר דרכים נפוצות להשגת הרשאות, ובפועל לבצע Privilege Escalation על עמדת הקצה - ממשתמש רגיל למשתמש חזק .

השיטה הראשונה אותה נבחן נקראת "Locate Stored Credentials" ומטרתה חיפוש אפקטיבי וממוקד של הרשאות (שמות משתמשים וסיסמאות) המאוחסנות על עמדת הקצה או על משאבי רשת שונים שאליהם יש לנו הרשאות.

יתרונות השיטה: ניצול של טעויות אנוש הקורות לעתים קרובות ולרוב של אנשי IT בעלי הרשאות גבוהות, שיטת חיפוש "שקטה" בעלת סיכוי נמוך להפעלת מנגנוני אבטחה והגנה שונים.

חסרונות השיטה: לעתים ניתקל ב"מלכודות דבש" (Honeypots, ניתן לקרוא בהרחבה כאן https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey), עלולים להיתקל בשלל הרשאות לא רלוונטיות או משתמשים שפג תוקפם.

ננסה להגדיר מספר מילות מפתח שאותן נחפש במיקומים המוגדרים כ"מועדים לפורענות":

- על הכונן המקומי שבו מותקנת מערכת ההפעלה ישנם מספר קבצים הנוצרים כברירת מחדל על ידי תוכנות הפצה או התקנות שונות, ובהם עלולים להישמר שמות משתמשים ואף סיסמאות. בין היתר, ניתן לחפש את הקבצים: sysprep.xml, sysprep.inf, unattend.xml.
- במידה ואנו נמצאים בסביבת Windows Domain, נרצה לחפש את הקובץ הידוע לשמצה Groups.xml, שבגרסאות שונות של Windows Server עלול להכיל סיסמאות מוצפנות של משתמשים חזקים בסביבת הדומיין - שאותן ניתן לפצח בקלות בעזרת מספר כלים פשוטים למדי. (ניתן למצוא הסבר יותר נרחב כאן https://pentestlab.blog/tag/cpassword/).
- אם נרצה להרחיב את החיפוש, נוכל לבצע חיפושים על מילות מפתח כגון "pass", "password", "credentials" ועוד. את החיפוש נתאים לסביבה שבה אנו נמצאים, ולכן אין כאן המלצה ספציפית - אלא הכוונה למציאת משאבי רשת שבהם משותפים קבצים רבים, מסמכים, וכו'.

השיטה השנייה אותה נבחן הינה ניצול הגדרות מתירניות של מערכת ההפעלה:

יתרונות השיטה: חיפוש ממוקד וקצר של הגדרות שונות במערכת ההפעלה.

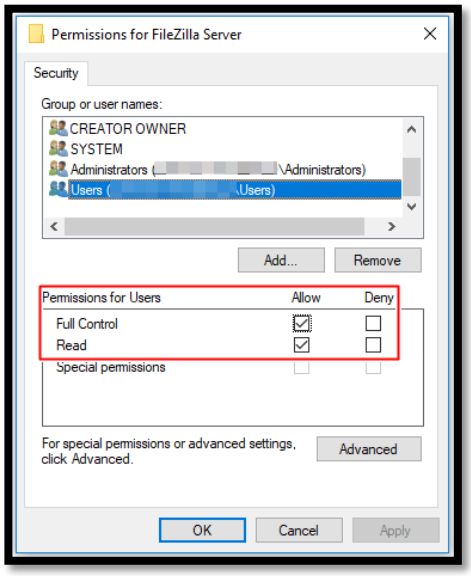
חסרונות השיטה: ההגדרות המתירניות אותן נחפש אינן נפוצות בסביבות ארגוניות גדולות.

נבחן מספר הגדרות של מערכת הפעלה המעידות כי ניתן לנצלן בכדי לבצע Privilege Escalation:

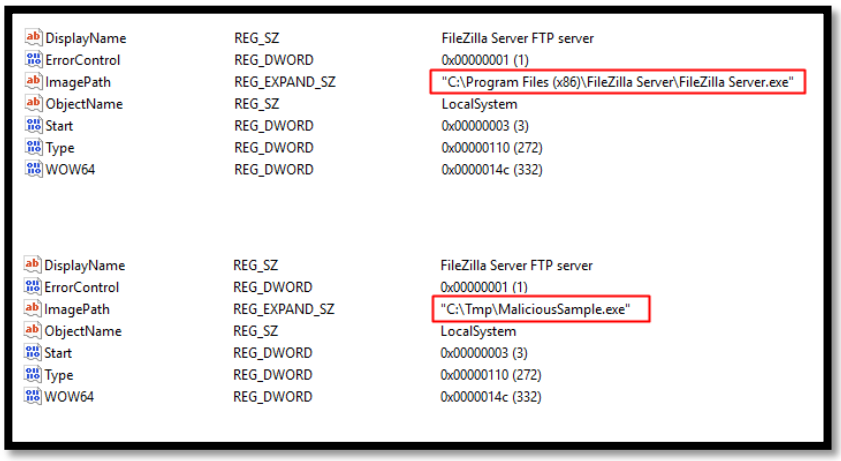
- שירותים (Services) בעלי הרשאות Registry מתירניות. במערכת ההפעלה Windows, כל המידע הקשור לשירותים שונים במערכת נשמר כערכי Registry במיקום הבא:

```
HKLM\SYSTEM\CurrentControlSet\Services\
```

- במידה ונמצא במיקום זה שירותים בעלי הרשאות מתירניות, למשל:



נוכל לנסות ולהחליף את ה-Executable שהשירות מריץ על ידי שינוי ערך ה-ImagePath:



ניתן כמובן גם לשנות את הערך ל-cmd.exe או powershell.exe ובכך להקטין את הסיכוי לגילוי קובץ זדוני במערכת.

שלב 2 - Defense Evasion - התחמקות מזיהוי והתרעה

בשלבים הבאים נשתמש בפעולות שהן מעט יותר "רועשות", ולכן נרצה לצמצם את הסיכויים שמנגנוני ההגנה המותקנים על המערכת יזהו אותנו.

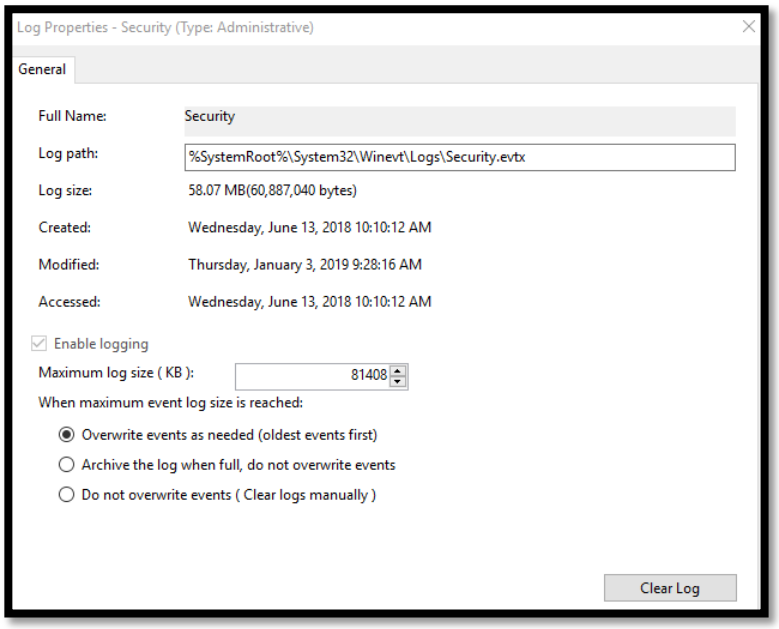
כמיטב המסורת, לא נעשה שימוש בכלים חיצוניים - אלא ביכולות מובנות של מערכת ההפעלה בלבד.

נדון כעת בשתי פעולות עיקריות:

- מניעת תקשורת בין עמדת הקצה לבין השרת - בכדי למנוע תקשורת מעמדת הקצה לשרת תוך שימוש ביכולות מובנות של מערכת ההפעלה בלבד, נשתמש ביכולות הניתוב המובנות של מערכת ההפעלה, וננתב כל תקשורת אל שרת הניהול של ה-Anti-Malware אל כתובת ה-Loopback. (ניתן לקרוא עוד כאן https://www.techopedia.com/definition/2440/loopback-address).
- את הפעולה נבצע בצורה הבאה, כאשר נגדיר את הכתובת 10.0.0.2 ככתובת שרת ה-Anti-Malware:

```
Route add 10.0.0.2 mask 255.255.255.255 127.0.0.1
```

- הסתרת פעולות על ידי "העלמת" אירועים של מערכת ההפעלה - ככל שמספר האירועים והעקבות שנשאיר אחרינו יהיה קטן יותר, כך יקטנו בהתאם הסיכויים לתפיסתנו (בזמן אמת, או במבט לאחור). בכדי להעלים עקבות, נרצה לפחות בשלב הראשוני למחוק אירועים מעמדת הקצה. במערכת ההפעלה Microsoft Windows יומן האירועים נקרא Event Viewer, והוא מכיל שלל אירועים לגבי מערכת ההפעלה, תוכנות צד ג', פעולות של משתמשים וקבוצות על התחנה, ועוד.
- מחיקה "סתמית" של יומן האירועים או של אירועים ספציפיים ביומן תגרום להיווצרותם של אירועים נוספים המתריעים על מחיקת אירועים, ובכך נייצר רעש מיותר.
- אז איך בכל זאת נוכל לגרום לטשטוש עקבותינו? נחטט בהגדרות היומן בכדי למצוא רמזים...



- אם נביט היטב ונעמיק בהגדרות הנ"ל, נוכל לראות כי היומן מוגדר כברירת מחדל למחוק אירועים ישנים כאשר הוא מתמלא. במידה ונקטין את יומן האירועים לגודל המינימלי האפשרי, נוכל לדרוס אירועים בקלות ולהעלים ראיות מתחנת הקצה. למזלנו - שינוי גודלו של יומן האירועים לא מייצרת אירוע כברירת מחדל ולכן נבחר באופציה הזו כאופציה המועדפת עלינו.

- ניתן לבצע את הפעולה באופן הבא:

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\EventLog\Security /v
MaxSize /t REG_DWORD /d 100000 /f
```

  - לאחר ביצוע הפעולה, נקבל התראה "The Operation Completed Successfully". חשוב לזכור! לאחר ביצוע פעולות "בעייתיות" יש לבצע מספר פעולות סטנדרטיות בכדי לדרוס אירועים ישנים.

כמובן שישנן עוד עשרות ואף מאות שיטות להימנע מגילוי על ידי תוכנות הגנה שונות, וניתן להרחיב בקריאה כאן:

https://attack.mitre.org/tactics/TA0005/

שלב 3 - Privilege Escalation | משתמש חזק ← SYSTEM

בשלב זה, נבחן דרכים אפשריות להשגת הרשאות SYSTEM על מערכת ההפעלה, ונמשיך לפעול על פי שיטת העבודה של שימוש בכלים סטנדרטיים של מערכת ההפעלה בלבד, ללא "עזרה חיצונית".

ישנן אינספור שיטות שבעזרתן נוכל להעמיק את שליטתנו על תחנת הקצה ולהשיג את הרשאת ה-SYSTEM הנכספת, אך נרצה לבצע זאת תחת המגבלות של שימוש בכלים מובנים של מערכת ההפעלה ויצירת "רעש" אפסי ככל האפשר, בכדי לשמור על חשאיות.

מי שבכל זאת מתעניין בדרכים נוספות להשגת הרשאות SYSTEM, או כל סוג אחר של Privilege Escalation, מוזמן להיכנס ל-Git הבא של "swisskyrepo" ולצפות בשלל דרכים שונות ומשונות, גם למערכות הפעלה נוספות:
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md

כאמור, נדון בשתי שיטות עיקריות להשגת הרשאות SYSTEM על מערכת ההפעלה ללא שימוש בכלים חיצוניים. שתי השיטות הינן הוכחות חיות ובועטות כי ארגונים כמו Microsoft לעתים רבות שמים את חווית המשתמש ונוחות העבודה במקום הראשון, הרבה לפני אבטחת מידע והגנה על המשתמשים ומערכת ההפעלה.

השיטה הראשונה שבה נדון היא ניצול יכולת ה-Debug המובנית של מערכת ההפעלה, מנגנון הנקראImage File Execution Options, או בקצרה - IFEO. המנגנון מאפשר הפעלת Debugger עבור קבצי הרצה שונים. יכולת זו הינה יכולת מבורכת, אך לצערנו יושמה בצורה בעייתית - ניתן להגדיר **כל קובץ הרצה** כ-Debugger לקובץ הרצה אחר, שיפעל באותן הרשאות.

במקור, היכולת פותחה לצורך הצמדת Debugger לתוכנות ושירותים שונים, בעיקר לצרכי פיתוח ו-QA, אך בפועל משתמשים בה רבות לצרכים זדוניים, כגון השגת Shell בעל הרשאות SYSTEM על תחנת קצה.

בכדי לנצל את מנגנון ה-IFEO בכדי לבצע Privilege Escalation, נצמיד לקובץ הרצה של מערכת ההפעלה Debugger מסוג Shell מובנה של מערכת ההפעלה כגון CMD.exe או PowerShell.exe. אך איך נבחר קובץ מתאים, שיוכל לתת לנו הרשאות SYSTEM?

נבחן אילו שירותים של מערכת ההפעלה רצים כ-SYSTEM (בקונוטציה מסוימת, כמובן). בנוסף לכך נרצה לנצל שירותים שמצד אחד יהיו נגישים ומצד שני לא יעלו "חשד" כאשר נשתמש בהם.

שתי האופציות הפופולריות ביותר וכמובן הרלוונטיות ביותר עבור המקרה שלנו הן חלק אינטגרלי משירותי הנגישות המובנים של מערכת ההפעלה, ושמן: Sticky Keys ו-Utility Manager.

הראשון משמש כעזר למשתמשים בעלי מוגבלויות פיזיות שונות, ואילו השני משמש כתפריט המנגיש שירותים שונים לבעלי מוגבלויות ראיה, מוגבלויות פיזיות ועוד.

ניתן לקרוא בהרחבה על הנושא כאן:

https://en.wikipedia.org/wiki/Sticky_keys

בכדי להצמיד Debugger מסוג CMD Shell אל שירות ה-Sticky Keys, נשתמש בפקודה הבאה:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options\sethc.exe" /v Debugger /t REG_SZ /d
"C:\Windows\System32\cmd.exe" /f
```

בכדי להצמיד Debugger מסוג CMD Shell אל שירות ה-Utility Manager, נשתמש בפקודה הבאה:

```
reg add "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File
Execution Options\utilman.exe" /v Debugger /t REG_SZ /d
"C:\Windows\System32\cmd.exe" /f
```

יתרונות השיטה: קלה לשימוש וגמישה - ניתן להגדיר כל Debugger ולשנות אותו בכל עת.

חסרונות השיטה: קלה לזיהוי, מערכות הגנה רבות מכירות שיטה זו ומתריעות/חוסמות פעולות המשנות ערכי Registry ב-IFEO.

השיטה השנייה שבה נדון היא שיטה מעט "פרימיטיבית", אך לפעמים נהיה מוכרחים להשתמש בה בכדי להימנע מזיהוי על ידי תוכנות הגנה שונות, או לפחות להקטין את הסיכוי להיות מזוהה כ"פעולה עוינת".

סדר הפעולות בשיטה זו כולל: השתלטות מלאה על קובץ הרצה של מערכת ההפעלה והחלפתו בקובץ אחר, רצוי ב-Shell כלשהו של מערכת ההפעלה, שאיננו עוין (בינתיים...)

ננסה לייצור סוג של "אוטומציה" לפעולות הנ"ל, בעזרת קובץ Batch פשוט למדי:

```
Takeown /F %windir%\System32\sethc.exe
calcs %windir%\System32\sethc.exe /e /p @UsernameGoesHere@:f
ren %windir%\System32\sethc.exe sethc_backup.exe
copy %windir%\System32\cmd.exe %windir%\System32\sethc.exe
```

- ניתן לשים לב כי בשורה 3 אנו משנים את שמו של קובץ ה-sethc.exe ולא מוחקים אותו. נעשה זאת בכדי להחזיר את מערכת ההפעלה למצב הקודם, בכדי להשאיר כמה שפחות נזק ועקבות בסוף התהליך.
- יש לשים לב כי בשורה 2 צריך להכניס את שם המשתמש/הקבוצה הרלוונטיים, בהתאם לתחנת הקצה ולמשתמש הנוכחי.
- יש להריץ את הסקריפט תחת הרשאות Local Administrator.
- ניתן לבצע את אותן פעולות על קובץ ה-utilman.exe.

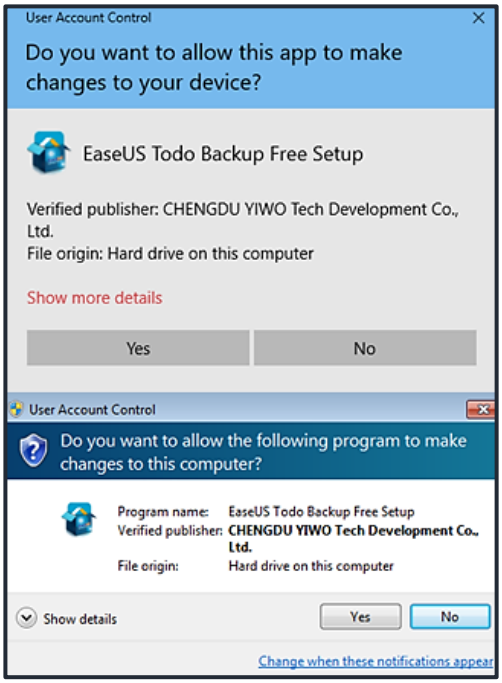
יתרונות השיטה: מספר מועט של מערכות הגנה מזהות את השיטה הזו.

חסרונות השיטה: מעט מסורבלת. שימוש לא נכון עלול לגרום למחיקת קובץ ההרצה המקורי ולהשאיר "עקבות" שונות על תחנת הקצה.

כעת, לאחר שהשתמשנו באחת השיטות לניצול "כלי הנגישות" השונים של מערכת ההפעלה לצורך השגת Shell, נרצה להריץ את ה-Shell ולעבוד כ-SYSTEM. לצורך כך, נבחן באילו מקרים כלי הנגישות שניצלנו פועלים בהרשאות SYSTEM:

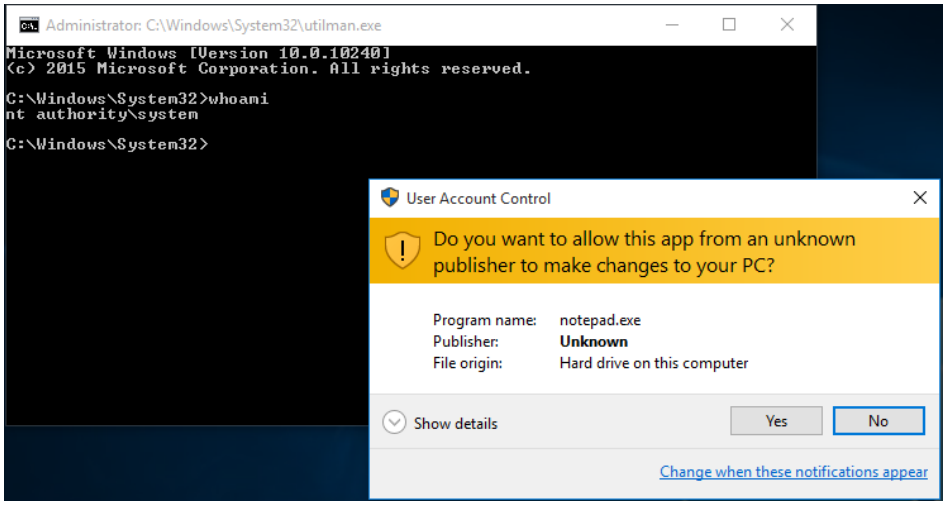
- מסך ההתחברות הראשוני של מערכת ההפעלה. מכיוון שבזמן עליית מערכת ההפעלה רצות פעולות רבות כגון Group Policies, אך בפועל עדיין לא בוצעה התחברות עם משתמש, Microsoft החליטו להפעיל את שלל האפליקציות והתכונות תחת הרשאות SYSTEM.

  אופציה זו הינה האופציה הנפוצה ביותר, אך יש לה מספר חסרונות שכדאי להכיר:
  - דורשת התנתקות/החלפת משתמש. במידה ואנו נמצאים בתחנה שאינה שלנו, ואין לנו שם משתמש וסיסמה עבור התחנה - לא נוכל להתחבר חזרה. כמובן שנוכל לייצר לעצמנו משתמש מקומי ולהתחבר בעזרתו, אך נייצר רעש רב ובמקרה זה נראה להימנע ככל האפשר מאופציה זו.
  - בגרסאות שונות של מערכת ההפעלה Microsoft Windows ישנם באגים שונים המפריעים לנו להפעיל מספר אפליקציות ותכונות של מערכת ההפעלה, ובכך לגרום לנו לעבודה ארוכה ומפרכת. (למשל, נתקלתי בגרסה של Windows 7 שאינה מאפשרת לי לפתוח תפריטים מבוססי MMC (https://en.wikipedia.org/wiki/Microsoft_Management_Console).
  - מערכת ה-UAC (User Account Control) של Windows. יידרשו הסברים רבים בכדי להעמיק ולהרחיב על מערכת ה-UAC, אך בקצרה - המערכת מונעת ממשתמשים או תוכנות לבצע פעולות המוגדרות כ-"שינויים" במערכת ההפעלה, אלא בעזרת אישור של משתמש בעל הרשאות גבוהות במערכת ההפעלה.



ניתן לקרוא בהרחבה על ה-UAC באתר הבא: https://www.digitalcitizen.life/uac-why-you-should-never-turn-it-off, ובנוסף אתן קרדיט לאתר על התמונה המוצגת לעיל.

כאשר ננסה להריץ קובץ מסוים במערכת ההפעלה כ-Administrator (לחיצה ימנית Run as Administrator), תצוץ התראה של מערכת ה-UAC שתוודא מולנו כי הפעלת הקובץ בוצעה במודע ואנו מאשרים שינויים שהקובץ עלול לבצע במערכת ההפעלה. מערכת ה-UAC רצה כ-SYSTEM על מערכת ההפעלה, ולכן בזמן שהיא "שואלת" אותנו לגבי התוכנה שאנו עתידים להריץ, ננסה להפעיל את ה-Sticky Keys או ה-Utility manager וזאת על ידי לחיצת Shift 5 פעמים ברצף, או לחיצה על Winkey+U:



על ידי הפקודה "whoami" נוכל לוודא כי ה-Shell שהופעל אכן רץ תחת הרשאות SYSTEM. כעת, לאחר שהשגנו את משתמש ה-SYSTEM הנכסף, נבחן כיצד נוכל לנטרל את מערכת ה-DLP המותקנת על עמדת הקצה.

נבחן את כל השירותים שרצים על מערכת ההפעלה, בעזרת הפקודה:

```
Sc query
```

נקבל מספר רב של תוצאות, ולכן נרצה לסנן את התוצאות בזמן אמת לכדי שירותים הקשורים ל-DLP בלבד:

```
Sc query | find DLP
```

נראה כי קיים שירות בשם "McAfee DLP Endpoint Service" - ננסה לעצור אותו אך נראה שזה בלתי-אפשרי, ככל הנראה כי הוא כבר פועל. ננסה להגדיר את צורת האתחול שלו על ידי:

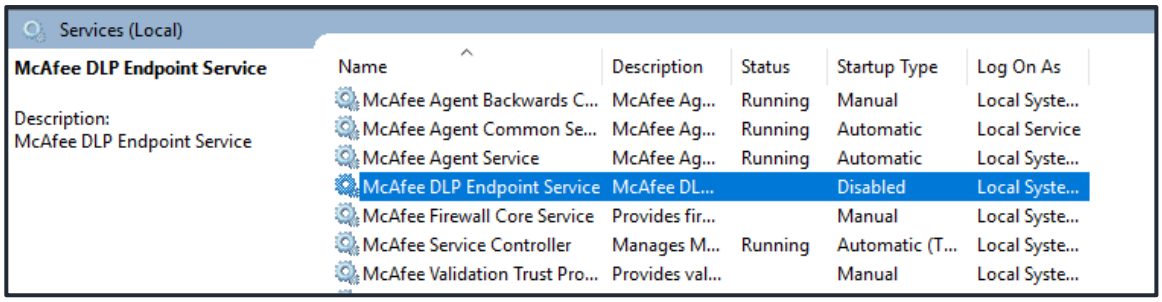
```
Sc config McAfeeDLPAgentService start= disabled
```

ננסה עכשיו לעצור את השירות:

```
Sc stop McAfeeDLPAgentService
```

נראה כי השירות אכן עצר, וננסה לחבר התקן USB בזמננו הפנוי ☺.

ה-Watchdog שמטרתו להגן על שירותי McAfee בעמדת הקצה לא מוגדר להגן על שירות ה-DLP ולכן ניתן היה בקלות יחסית לבטלו:



בגסאות מסוימות של McAfee DLP Agent נצטרך לבצע הפעלה מחדש של מערכת ההפעלה בכדי ששירות ה-DLP יבוטל לאלתר.

סיכום

כפי שראינו לאורך המאמר, מערכת ההפעלה טומנת בחובה מספר רב של כלים הניתנים לניצול על ידי תוקף - ולכן אם אין באפשרותו להעביר כלים חיצוניים לתחנת הקצה, בכל זאת יהיה ניתן לבצע מספר רב של פעולות זדוניות. פשטות היא שם המשחק, ואלתור תוך שימוש במה שנמצא בהישג יד הוא יכולת חשובה, גם לצד התוקף וגם לצד המגן.

המאמר נכתב ע"י אביאל צרפתי, יועץ וחוקר אבטחת מידע בחברת הייעוץ Accenture.